

ATM & MPLS Theory & Application

Foundations of Multi-Service Networking

- Get details on current standards within ATM, frame relay, and IP (MPLS) technology
- Solve network and application problems with speed and precision
- Understand how MPLS supports cost-effective IP networking
- Preview the future role of multi-service ATM and MPLS networks


DAVID E. MCDYSAN AND DAVE PAW



ATM & MPLS Theory & Application: Foundations of Multi-Service Networking

DAVID **MCDYSAN**
DAVE **PAW**

McGraw-Hill/Osborne
New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto



McGraw-Hill/Osborne



A Division of The McGraw-Hill Companies

Copyright © 2002 by The McGraw-Hill Companies, Inc. All rights reserved. Manufactured in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

0-07-222837-7

The material in this eBook also appears in the print version of this title: 0-07-222256-5

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill eBooks are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please contact George Hoare, Special Sales, at george_hoare@mcgraw-hill.com or (212) 904-4069.

TERMS OF USE

This is a copyrighted work and The McGraw-Hill Companies, Inc. (“McGraw-Hill”) and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill’s prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED “AS IS”. MCGRAW-HILL AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

DOI: 10.1036/0072228377



Professional

Want to learn more?

We hope you enjoy this McGraw-Hill eBook! If you'd like more information about this book, its author, or related books and websites, please [click here](#).

[For more information about this title, click here.](#)

CONTENTS



Introduction	xix
------------------------	-----

Part I

Overview, Introduction, Background, Motivation, and Standards

▼ 1 Introduction to ATM and MPLS and Overview of the Book	3
Overview of This Book	6
Review	12
▼ 2 Background and Motivation for ATM and MPLS Networking	13
A Brief History of Communications	14
Recurring Trends in Encoding and Relaying	14
Data Networking: Enabling Computers to Communicate.	15
Changing Organizations of People and Networks	16
Defining the Demand for Communications	17
Residential and Commercial Users	17
Applications and Networks Change Faster Than Behavior	18
Geographical Aspects of Networking	18
The End Result: Tremendous Internet and Data Traffic Growth	19
Technology Trends	19
Processor and Memory Cost Trends: Moore's Law	19
Distributed Computer Communications Protocols	20
Modernization of Transmission Infrastructures	20
Faster and Farther, but Never Free	21

	The Accelerating Bandwidth Principle	21
	Worldwide Cooperation for Standards	22
	Review	24
▼ 3	ATM- and MPLS-Related Standards Bodies	25
	ATM- and MPLS-Related Standards Bodies	26
	International Telecommunications Union (ITU)	27
	ATM Forum	27
	Internet Engineering Task Force (IETF)	28
	Frame Relay Forum	29
	MPLS Forum	29
	DSL Forum	30
	Other B-ISDN/ATM Standards Bodies	30
	Creating Standards: The Players	30
	Vendors	30
	Users	31
	Network Service Providers	31
	Creating Standards: The Process	32
	Charter and Work Plan	33
	Meetings and Contributions	33
	Drafting and Review	33
	Approval and Consensus	34
	User Acceptance and Interoperability	34
	Other Aspects of Standards	35
	Business and Politics	35
	Measures of Success and Proven Approaches	35
	Predicting the Future of Standardization	36
	Review	36

Part II

Networking and Protocol Fundamentals

▼ 4	Networks, Circuits, Multiplexing, and Switching	39
	General Network Topologies	40
	Point-to-Point	41
	Multipoint and Broadcast	42
	Star	44
	Ring	45
	Mesh	46
	Data Communications and Private Lines	47
	Simplex, Half-Duplex, and Full-Duplex Transmission	47
	DTE-to-DCE Connections	48
	Private Lines	50
	Data Transmission Methods	50
	Asynchronous and Synchronous Data Transmission	51
	Asynchronous Versus Synchronous Transfer Modes	52
	Principles of Multiplexing and Switching	53
	Multiplexing Methods Summarized	54
	Space Division Multiplexing (SDM)	54
	Frequency Division Multiplexing (FDM)	54
	Time Division Multiplexing (TDM)	55
	Address or Label Multiplexing	55

Code Division Multiple Access (CDMA)	55
Point-to-Point Switching Functions	56
Point-to-Multipoint Switching Functions	56
Examples of Multiplexing	57
Examples of Switching	62
Review	67
▼ 5 Basic Protocol Concepts	69
A Brief History of Packet Switching	70
Early Reasons for Packet Switching	71
Principles of Packet Switching	71
Darwin's Theory and Packet-Switching Evolution	73
Basic Protocol Layering Concepts	75
Open Systems Interconnection Reference Model	77
Layers of the OSI Reference Model	81
Physical Layer	81
Data Link Layer	82
Network Layer	83
Transport Layer	83
Session Layer	84
Presentation Layer	84
Application Layer	84
Mapping of Generic Devices to OSI Layers	84
Layered Data Communication Architectures	85
Internet Protocol (IP) Architecture	85
IBM's Systems Network Architecture (SNA)	86
IEEE 802.X Series (LAN/MAN/WAN)	87
Integrated Services Digital Network Protocol Architecture	89
Network Service Paradigms	91
Connection-Oriented Network Service (CONS)	91
Connectionless Network Services (CLNS)	92
Connection-Oriented Versus Connectionless Services Analogy	94
Review	94
▼ 6 Time Division Multiplexing and the Narrowband Integrated Services Digital Network	95
Circuit Switching	96
History of Circuit Switching	96
Digitized Voice Transmission and Switching	97
Digital Data Circuit Switching	98
Private-Line Networks	100
Private (Leased)-Line Characteristics	100
Private-Line Networking	100
Permanent Versus Switched Circuits	103
Digital Time Division Multiplexing (TDM)	104
Plesiochronous Digital Hierarchy (PDH)	104
SONET and the Synchronous Digital Hierarchy (SDH)	106
Basic SONET Frame Format	110
Basics and History of Narrowband ISDN (N-ISDN)	113
Narrowband ISDN Basics	113
BRI and PRI Service and Protocol Structures	115
ISDN D-Channel Signaling	117
Review	119

▼ 7	Connection-Oriented Protocols—X.25 and Frame Relay	121
	Packet Switching	122
	Origins of X.25	122
	Protocol Structure	123
	Networking Context	124
	SDLC, HDLC, and X.25's Link Layer Protocol	125
	Packet Layer Format and Protocol	131
	Control Functions	133
	Example of X.25 Operation	133
	Traffic and Congestion Control Aspects of X.25	135
	Service Aspects of X.25	137
	Frame Relay—Overview and User Plane	137
	Origins of Frame Relay	137
	Frame Relay Protocol Structure	138
	Frame Relay Networking Context	139
	Frame Format	140
	Frame Relay Functions	142
	Example of Frame Relay Operation	143
	Traffic and Congestion Control Aspects of Frame Relay	144
	Service Aspects of Frame Relay	147
	Frame Relay—Control Plane	149
	Frame Relay Control Protocol Networking Context	149
	Frame Relay Standards and Specifications	150
	Frame Relay PVC Status Signaling	152
	Frame Relay PVC Status Signaling Example	155
	Multilink Frame Relay	157
	Frame Relay Service Level Agreements (SLAs)	159
	Frame Relay Operations, Administration, and Maintenance	161
	Frame Relay Fragmentation and Compression	164
	Frame Relay Privacy	166
	Frame Relay Switched Virtual Connections (SVCs)	168
	Example of Frame Relay SVC Operation	168
	Frame Relay Signaling Message Information Elements	169
	Review	174
▼ 8	Connectionless Protocols—IP and SMDS	175
	The Internet Protocol SUITE, TCP/IP	176
	Origins of TCP/IP	176
	TCP/IP Protocol Structure	177
	TCP/IP Networking Context	178
	Generic Link Layer Protocols for IP	180
	IP Version 4 (IPv4) Packet Format	182
	Internet Protocol (IP) Addressing	183
	Next Generation IP—IPv6	184
	Quality of Service in IP Networks	186
	Transmission Control Protocol (TCP)	190
	User Datagram Protocol (UDP)	196
	Real-Time Transport Protocol (RTP)	196
	Service Aspects of TCP/IP	198
	Switched Multimegabit Data Service (SMDS)	198
	Origins of SMDS	198
	SMDS/IEEE 802.6 Protocol Structure	199
	SMDS/802.6 Protocol Data Unit (PDU) Formats	199

	DQDB and SMDS Operation	202
	Example of SMDS over DQDB Operation	204
	Traffic and Congestion Control Aspects of DQDB and SMDS	204
	Service Aspects of SMDS	205
	Review	206
▼ 9	LANS, Bridging, and Routing	207
	Bridging, Routing, and Internetworking	208
	Basic Terminology	208
	Address Assignment and Resolution	210
	Routing, Restoration, and Reconfiguration	211
	IEEE Local Area Networking (LAN) Standards	212
	Layered LAN Protocol Model	213
	Typical LLC and MAC Sublayer Implementations	213
	The Logical Link Control (LLC) Sublayer	214
	The Media Access Control (MAC) Sublayer	215
	Ethernet and the CSMA/CD 802.3 MAC Sublayer	217
	Ethernet User Priority and VLANs	219
	Token Ring	220
	100 Mbps Fast Ethernet	222
	100VG-AnyLAN	223
	Gigabit and 10 Gbps Ethernet	224
	Fiber Distributed Data Interface (FDDI)	224
	Basic Fiber Distributed Data Interface (FDDI)	225
	Hybrid Ring Control (FDDI-II)	228
	Bridging Concepts, Systems, and Protocols	229
	Bridging Context	230
	A Taxonomy of Bridges	231
	Spanning Tree Protocol	232
	Source Routing Protocol	233
	Bridge Network Design	234
	Routing Concepts, Systems, and Protocols	235
	Packet-Forwarding and Routing Protocol Functions	235
	Link-State Routing Protocols Defined	238
	Routing and Logical IP Subnetworks (LISs)	242
	Address Resolution Protocol (ARP)	245
	Bridging and Routing Systems Design	247
	Review	249

Part III

Foundations of ATM and MPLS: Protocol and Structure

▼ 10	Introduction to ATM and MPLS	253
	Introduction to ATM and B-ISDN	254
	B-ISDN Protocol Reference Model	254
	B-ISDN Architecture	255
	Overview of the Application of ATM	256
	ATM as a Technology	257
	ATM as a Protocol	257
	ATM as an Interface	258
	ATM as Integrated Access	259

ATM as an End-to-End Service	261
ATM as a Scalable Infrastructure	261
Origins of MPLS: Reinventing IP over ATM	263
Ipsilon's IP Switching	265
Toshiba's Cell Switching Router (CSR)	267
Cisco's Tag Switching	267
IBM's Aggregate Route-Based IP Switching (ARIS)	270
Early IETF Multiprotocol Label Switching (MPLS)	272
Introduction to MPLS	274
Traffic Engineering of IP Networks	275
Network-Based IP VPN using MPLS Tunneling	276
Multi-Service MPLS Tunneling	276
Considerations in the Choice of Cells Versus Frames	277
Effect of Link Speed on Packet Performance	277
Rationale for the Choice of ATM Cell Size	278
Hardware Price-Performance Trade-offs	279
Review	280
▼ 11 ATM and MPLS: Physical Layer and Label Switching Functions	281
Overview of Physical, ATM, and AAL Layer Functions	282
B-ISDN Protocol Layer Structure	283
Hardware and Software Implementations of B-ISDN Layers	284
ATM Physical Layer	285
Physical Medium-Dependent Sublayer	285
Transmission Convergence (TC) Sublayer	287
TC Header Error Check (HEC) Functions	288
TC Cell Rate Decoupling	290
Inverse Multiplexing over ATM	290
xDSL Physical Layer for ATM	292
ATM Layer	296
ATM UNI and NNI Defined	296
ATM Virtual Paths and Channels (VPs and VCs)	297
The ATM Cell	302
ATM-Layer QoS and Service Categories	306
Multiprotocol Label Switching (MPLS)	308
IP over MPLS Architecture and Terminology	308
MPLS Forwarding Operations	309
Example of MPLS Forwarding of IP Packets	312
MPLS Encapsulation Standards	312
MPLS Shim Header	312
MPLS over ATM	315
MPLS over Frame Relay	317
Review	318
▼ 12 ATM Adaptation and MPLS Tunneling Protocols	319
ATM Adaptation Layer (AAL)	320
ATM Adaptation Layer (AAL)—Protocol Model	320
AAL Protocol Structure Defined	321
Key AAL Attributes	322
ATM Adaptation Layer 1 (AAL1)	323
AAL1 Segmentation and Reassembly (SAR) Sublayer	324
AAL1 Convergence Sublayer Functions	325

Structured Data Transfer (SDT) Convergence Sublayer	327
Unstructured Mode Convergence Sublayer	329
AAL1 Clock Recovery Methods	330
ATM Adaptation Layer 2 (AAL2)	332
AAL2 Protocol Structure and PDU Formats	333
Example of AAL2 Operation	335
ATM Adaptation Layer 3/4 (AAL3/4)	337
AAL3/4 SAR Sublayer	337
AAL3/4 CPCS Sublayer	338
Example of AAL3/4 Operation	339
AAL3/4 Multiplexing Example	340
ATM Adaptation Layer 5 (AAL5)	340
AAL5 Segmentation and Reassembly (SAR) Sublayer	342
AAL5 Common Part Convergence (CPCS) Sublayer	342
Example of AAL5 Operation	343
AAL5 Multiplexing Example	344
Multi-Service Tunneling over MPLS (and Other Protocols)	346
General Concept of Protocol Tunneling	346
ATM Forum's ATM over MPLS Network Interworking	348
IETF Pseudo Wire Emulation Edge to Edge (PWE3)	350
"Martini" Multi-Service Encapsulation	351
Review	352
▼ 13 Higher-Level User and Control Plane Protocols	353
Overview of Higher-Layer ATM and MPLS Protocols	354
Circuit Emulation Voice, Video, and WAN Data Protocols	354
Local Area Networking and IP-Based Applications	356
ATM Service Category and AAL Support for Applications	358
Overview of ATM and MPLS Control Plane Protocols	358
Generic Control Plane Functions	359
Switched and Permanent ATM Virtual Connections	359
ATM Control Plane Protocols	360
MPLS Control Plane Protocols	361
ATM Control Plane Structure and AAL	361
ITU-T B-ISDN Signaling Protocols	362
Types of Signaling Channel Association	363
Layered Signaling AAL Model	365
Service Specific Coordination Function (SSCF)	365
Service Specific Connection-Oriented Protocol (SSCOP)	366
ATM User-Network Interface (UNI) Signaling	368
Base Signaling Functions: Q.2931 and UNI 3.1	368
ATM Forum UNI Signaling 4.0 and ITU-T Standards	368
ATM Forum UNI Signaling 4.1 and ITU-T Standards	370
UNI 4.1 Signaling Message Types	371
Signaling Message Information Elements	372
Examples of ATM Signaling Procedures	373
ATM Control Plane Addressing	378
Control Plane Addressing Levels	378
ATM Level Addressing	379
ATM Addressing Formats	379
ATM Forum ATM End System Address (AESAs) Formats	381
Group Addresses and Anycast	382

ILMI Address Registration	383
Bi-Level Addressing	384
ATM Name Service (ANS)	384
Review	385
▼ 14 MPLS Signaling and Routing Protocols	387
MPLS Control Plane Architecture	388
MPLS Control and Forwarding Plane Model	388
Motivation for Constraint-Based Routing	389
MPLS Label Distribution Control Protocol Attributes	391
MPLS Label Distribution Signaling Protocols	397
Label Distribution Protocol (LDP)	397
RSVP Traffic Engineering (RSVP-TE)	400
Constraint-Based Routing LDP (CR-LDP) Extensions	404
Use of BGP for Label Distribution	405
IGP Traffic Engineering Extensions: OSPF and IS-IS	407
General Modifications for Traffic Engineering	407
Specific Modifications for IS-IS TE	408
Specific Modifications for OSPF-TE	408
Open Issues and Challenges Ahead	409
Example Applications of MPLS in IP Networks	409
Traffic Engineering in an IP Backbone	409
Label Distribution in Support of Other Services	411
MPLS Connectivity Across Multiple Providers	412
Review	413
▼ 15 ATM NNI Signaling and Routing Protocols	415
Interim Interswitch Signaling Protocol (IISP)	416
Private Network-Network Interface (PNNI)	416
Architecture and Requirements	417
Network Addressing Philosophy	418
A Tale of Two Protocols	419
PNNI Routing Hierarchy and Topology Aggregation	420
Beyond Connectivity to Quality and Bandwidth	427
Soft Permanent Virtual Connections (SPVCs)	431
Minimum Interoperable PNNI 1.1 Subset	432
Broadband InterCarrier Interface (B-ICI)	433
B-ISDN User Services Part (BISUP)	433
B-ICI's Replacement: ATM Inter-Network Interface (AINI)	434
Extended PNNI and AINI Routing and Signaling Capabilities	436
Review	439

Part IV

ATM and MPLS Support for Networking Applications

▼ 16 Enabling Voice, TDM, and Video Over ATM and MPLS	443
Packet Voice Networking	444
General Network Architecture	445
Media Gateway Functions	446
Packet Voice Encoding Standards	446
Quality Considerations	448

Voice Trunking Using ATM and MPLS	449
Voice over ATM (VoATM) Trunking	450
Voice over MPLS (VoMPLS) Trunking	457
Broadband Local Loop Emulation Using AAL2	459
Circuit Emulation Using ATM and MPLS	463
AAL1-Based Circuit Emulation Service (CES)	463
Circuit Emulation over MPLS	466
Video over ATM and Packet Networks	467
Commonly Used Video Coding Standards	467
MPEG-2 Video Over ATM and Packet Networks	468
QoS Considerations Related to Video	471
Review	472
▼ 17 Connection-Oriented Protocol Support	473
Interworking, Access, and Trunking	474
Overview of Frame Relay/ATM Interworking	477
Frame Relay/ATM Network Interworking	478
FR Service-Specific Convergence Sublayer (FR-SSCS)	479
Status Signaling Conversion	480
Congestion Control and Traffic Parameter Mapping	480
Frame Relay/ATM Service Interworking	481
Status Signaling Interworking	482
Address Resolution Protocol Interworking	483
FR/ATM SVC Service Interworking	484
FR/ATM Interworking Applied	486
ATM Access to SMDs	488
Frame-Based Interfaces Supporting ATM	489
ATM Data Exchange Interface (DXI)	489
Frame-Based User-Network Interface (FUNI)	493
Frame-Based ATM over SONET/SDH Transport (FAST)	496
Frame-Based ATM Transport over Ethernet (FATE)	497
MPLS-Based Support for Link Layer Protocols	498
Pseudo-Wire and Service Emulation Considerations	499
Martini Encapsulation and Transport of FR, AAL5, ATM, and HDLC	500
FR over MPLS Network Interworking	502
Review	503
▼ 18 ATM and MPLS Support for LAN Protocols	505
Multiprotocol Encapsulation over AAL5	506
Protocol Encapsulation	506
VC-Based Multiplexing	508
Considerations in the Selection of Multiplexing Method	510
ATM Forum LAN Emulation (LANE)	511
Hardware and Software in an Emulated LAN	511
LANE Components and Connection Types	514
Summary of LANE Operation	514
LANE and Spanning Tree	518
LANE Implementation Considerations	519
Ethernet over MPLS	520
Martini Encapsulation of Ethernet over MPLS	520
Virtual Private LAN Service (VPLS)	521

VPLS and Access to the Internet	525
Interworking Network Layer Protocols over MPLS	526
Metropolitan and Wide Area Ethernet over MPLS Networking	528
Review	530
▼ 19 ATM and MPLS Support of Enterprise-Level IP Networks	531
IP over ATM Virtual Private Networks	533
Classical IP over ATM	533
Multiprotocol over ATM (MPOA)	537
IP Multicast over ATM	542
IP Virtual Private Networks (VPN) over MPLS or IP Tunnels	545
General Virtual Private Network (VPN) Terminology and Concepts	545
Network-Based IP VPN Concepts	548
Aggregated Routing Network-Based VPNs Using Tunnels	550
Virtual Router Network-Based VPNs using Tunnels	554
Considerations and Trade-offs with Network-Based IP VPNs	556
Considerations Regarding Choice of Tunnel Type	557
VPN Representations and Configuration Complexity	558
IP Path Maximum Transfer Unit (MTU) Discovery	560
MTU Path Discovery over AAL5	560
MTU Path Discovery over MPLS	561
Review	562

Part V

Quality of Service, Traffic Management, and Congestion Control

▼ 20 The Traffic Contract and Quality of Service (QoS)	565
The Traffic Contract	566
Reference Models	567
Generic Allocation of Impairments Model	567
ATM Equivalent Terminal Model	568
Diffserv Per-Hop and Per-Domain Behavior Models	569
Quality Of Service	571
Application QoS Requirements	571
ATM QoS Parameters	573
IP Performance Metrics (IPPM)	578
Traffic Parameters and Conformance Definitions	579
ATM Traffic Descriptor	579
IP Traffic Descriptor	582
ATM Conformance Definitions	583
IP Traffic Conformance Definitions	585
Classes of Service	586
ATM Forum QoS Classes and Service Categories	586
ITU-T ATM QoS Classes	588
Mapping Between ATM Forum and ITU-T QoS Definitions	591
Diffserv Per-Hop Behaviors (PHBs)	594
MPLS Support for Diffserv	595

Comparison of ATM and IP QoS and Traffic Parameters	596
ATM Service Categories Optimized for Packet Switching	597
Guaranteed Frame Rate (GFR)	597
Switch Modifications to Support GFR	601
UBR with BSC and MDCR	603
Use of Differentiated UBR to Support Diffserv	604
Use of Differentiated UBR to Support IEEE 802.x	605
UBR Service Category with Optional MDCR Parameter	605
Review	607
▼ 21 Traffic Control, QoS Mechanisms, and Resource Management	609
Achieving Conformance	610
Checking Conformance: Policing	612
ATM Policing	613
Examples of Leaky Bucket Policing	613
Generic Cell Rate Algorithm (GCRA) and Virtual Scheduling	619
IP and MPLS Policing	620
Ensuring Conformance: Shaping	624
Overview of Possible Shaping Methods	625
Leaky Bucket Buffering	626
Token Bucket Shaping	627
Delivering QoS: Prioritization, Queuing, and Scheduling	630
Prioritized Queuing and Scheduling	630
Priority Discard Thresholds	631
Performance Implications of Priority Control	632
Overview of Weighted Scheduling Algorithms	633
Meeting the Traffic Contract: Resource Management	634
Admission Control	634
ATM VPs and Label Stacked MPLS LSPs	638
Review	640
▼ 22 Congestion Control	641
Congestion: A Familiar Phenomenon	642
The Nature of Congestion	642
Busy Seasons, Days, and Hours	643
Impact of Congestion	644
Examples of Congestion in a Network	644
Congestion Control: A Range of Solutions	645
Open- and Closed-Loop Congestion Control	645
Impact of Congestion on Performance	646
Categorization of Congestion Control Approaches	649
Congestion Management	652
Resource Allocation	652
Network Engineering	652
Congestion Avoidance	653
Congestion Indication	653
Policing and Tagging	654
Connection Blocking	654
Closed-Loop Flow Control	654
Generic Closed-Loop Flow Control Methods	655
ATM Generic Flow Control (GFC)	656

Available Bit Rate	657
The Great Rate Versus Credit Debate	659
Congestion Recovery	667
Selective Discard	667
Early/Partial Packet Discard (EPD/PPD)	668
Dynamic Usage Parameter Control (UPC)	670
Disconnection and/or Rerouting	670
Operational Procedures	671
Review	671

Part VI

Communications Engineering, Traffic Engineering, and Design Considerations

▼ 23 Basic Communications Engineering	675
Philosophy	676
Communications Channel Model	676
Deterministic Versus Random Modeling	677
Probability Theory	677
Randomness in Communications Networks	677
Random Trials and Bernoulli Processes	678
The Normal/Gaussian Distribution	678
Common Digital Signals and Their Spectra	679
The Telegraph Pulse: Binary On/Off Keying	680
A Better Way: Pulse Shaping	681
Pushing the Envelope: Quadrature Amplitude Modulation	681
Error Models and Channel Capacity	685
Typical Communications Channel Error Models	685
Shannon's Channel Capacity	686
Error Performance of Common Modulation Methods	688
Error-Detecting and -Correcting Codes	689
Simple Parity Check Schemes	689
Cyclical Redundancy Check (CRC) Codes	690
Performance of ATM's HEC	691
Undetected Error Performance of HDLC and AAL5	694
Data Compression	694
Review	696
▼ 24 Traffic Engineering	697
Philosophy	698
Source Model Traffic Parameter Characteristics	698
Modeling Accuracy	699
Overview of Queuing Theory	699
General Source Model Parameters	699
Poisson Arrivals and Markov Processes	702
Queuing System Models	705
Call Attempt Rates, Blocking, and Queuing	708
Statistical Model for Call Attempts	708
Erlang's Blocked Calls Cleared Formula	709
Erlang's Blocked Calls Held Formula	711
Performance of Buffering Methods	713
Input Versus Output Queuing Performance	713

Output Buffer Overflow Probability	714
Shared Buffer Performance	716
Deterministic Constant Rate Performance	718
Equivalent Capacity	720
Fluid Flow Approximation	721
Statistical Multiplex Gain Model	722
Equivalent Capacity Approximation	726
Priority Queuing Performance	728
Review	730
▼ 25 Design Considerations	731
Impacts of Delay, Loss, and Delay Variation	732
Impact of Delay	732
Impact of Loss	735
Impact of Delay Variation	738
TCP Performance Considerations	742
TCP Window Size Impact on Throughput	742
TCP over ATM: UBR and ABR	742
TCP/IP Performance in a Congested Scenario	743
Voice and Data Integration	745
Voice Traffic Model	745
Statistically Multiplexing Voice Conversations	746
Voice/Data Integration Savings	747
Overview of the Network Planning and Design Process	748
Network Design Approaches and Modeling Philosophy	749
Measuring Traffic and Performance Data	750
Analyzing and Simulating Candidate Networks and Technology	751
Practice Makes Perfect	752
Network Design and Modeling Tools	753
Design Tool Graphical User Interface (GUI)	753
Specifying Design Scenarios	754
Modeling Network-Specific Capabilities	755
Displaying and Comparing Results	755
Review	756

Part VII

Operations and Network Management for ATM and MPLS

▼ 26 Operational Philosophy and Network Management Architectures	759
OAM&P Philosophy	760
Administration	760
Provisioning	761
Operations	762
Maintenance	762
Unique Challenges Created by ATM	763
Unique Challenges Created by MPLS	763
Network Management Architectures	764
Centralized Versus Distributed Network Management	764
OSI Network Management Functional Model	765

ITU Telecommunications Management Network (TMN)	766
ITU-T Generic Transport Network Architecture	769
ATM Forum Network Management Architecture	772
Review	773
▼ 27 Network Management Protocols and Management Information Bases (MIBs)	775
Network Management Protocols	776
IETF Simple Network Management Protocol (SNMP)	776
ITU-T Common Management Interface Protocol (CMIP)	780
Proprietary Network Management Protocols	781
Considerations on Choice of Network Management Protocol	782
ATM Management Information Bases (MIBs)	782
ATM Forum Integrated Local Management Interface (ILMI)	783
IETF AToM MIBs	786
Other ATM MIBs	787
MPLS Management Information Bases (MIBs)	787
Label Switch Router (LSR) and Related MIBs	788
Traffic Engineering (TE) MIBs	789
Multiservice PPVPN and PWE3 MIBs	789
IP-Based Management Tools for MPLS	790
ICMP PING and Traceroute	790
Vendor-Proprietary ICMP Extensions for MPLS	791
IETF Direction for IP-Based MPLS Management	792
Review	793
▼ 28 ATM and MPLS Management and Performance Measurement	795
ATM OAM Flow Reference Architecture	796
ATM OAM Cell Formats	798
ATM OAM Fault Management	800
AIS and RDI Theory and Operation	800
Loopback Operation and Diagnostic Usage	802
Continuity Check (CC)	806
ATM Protection Switching	806
ATM Performance Specification and Measurement	810
Network Performance and Quality of Service	810
ATM Performance Measurement (PM)	810
NP/QoS Parameter Estimation	814
MPLS OAM Status and Direction	819
Overview of ITU Direction for MPLS OAM	819
MPLS Protection Switching and Fast Rerouting	820
Review	820

Part VIII

Design Considerations and Future Directions Involving ATM and MPLS

▼ 29 Design Considerations for ATM and MPLS Networks	825
Efficiency Analysis	826
Circuit Emulation Efficiency	826
Packetized Voice Efficiency	828
Efficiency of Cells Versus Frames for Packet Switching	829

IP/ATM, IP/MPLS, and IP/SONET Efficiency	831
Packet Video Efficiency	834
Multiservice Efficiency Comparison	835
Scalability Analysis	837
Addressing and Hierarchy	837
Supported User and Routing Table Growth	838
Packet Forwarding and Moore's Law	839
Connection-Oriented Versus Connectionless Paradigms	840
Support for a Wide Range of Interfaces and Speeds	841
Capacity Bottlenecks	842
Complexity Analysis	842
To Switch or Not to Switch? An Answer to This Question	842
Keep It Simple to Succeed	843
Hardware Is Hard, but Software Is Harder	843
Are QoS and Bandwidth Reservation Really Necessary?	844
Reliability, Availability, and Stability	846
Supportability and Operability	847
Security	847
Review	848
▼ 30 Future Directions and Applications Involving MPLS and ATM	851
Future Directions and Applications of ATM	852
Multiservice Backbone Network Infrastructure	852
Convergence and Integrated Access	853
Lessons Learned from ATM for Multiservice Networking	853
Don't Operate at the Per-Flow Level	853
Use Basic QoS and Traffic Management on Aggregates	854
Use Bandwidth Reservation for Constraint-Based Routing	854
Assume a Heterogeneous Underlying Network	854
Future Applications and Directions of MPLS (and IP)	855
Next Generation Multiservice Network Infrastructure	855
Optical Networking for Scalability	855
Generalized MPLS (GMPLS)	857
Separation of Forwarding and Control	860
Possible Future of Multiservice Networking	861
What Will Continue the Internet's Explosive Growth?	861
Will MPLS Become the Ubiquitous Multiservice Network?	862
Will GMPLS Effectively Control Next-Generation Backbones?	863
What Will Happen to ATM?	863
Review	863
▼ A Acronyms and Abbreviations	865
▼ B References	881
▼ Index	913

ABOUT THE AUTHORS

Dr. David E. McDysan is a fellow at WorldCom in the Internet architecture & technology department. He specializes in network cost optimization, next generation edge and core router technology, IP QoS, network-based VPNs, Voice over IP, and Internet standards. Prior to this assignment, he led an architectural planning group for next-generation switched networks at MCI and MCI WorldCom. During the mid-1990s, he designed and managed the commercial ATM network for MCI. David is currently active in the Internet Engineering Task Force (IETF). He has held leadership roles in the Multiservice Switching Forum and the ATM Forum. He has authored books on VPNs, QoS and traffic management, and coauthored three books on ATM.

Dave Paw is currently a consultant for telecommunications network design. Prior to this, he was a senior engineer, developing architectural options for next-generation networks. His expertise encompasses multi-service ATM and Frame Relay solutions, metro access networks, and optical control networks. Dave provided PNNI and MPLS expertise in the Network Architecture and Advanced Technology organization at WorldCom, and was active in the ATM Forum. Before his involvement with data network architectures, he produced detailed specifications for the WorldCom SONET/SDH and Digital Cross Connects, and was part of the group that planned and developed WorldCom's first DWDM infrastructure. When he's not working on network designs, Dave involves himself with inter-cultural and inter-religious projects.

ABOUT THE TECHNICAL EDITORS

Richard Carrara is currently a senior network architect at Data Return, Inc., based in Irving, Texas. He is a CCIE and CISSP with more than eight years of information technology experience. He specializes in the design and architecture of secure, large-scale IP networks.

Lei Yao is a Ph.D. candidate in the department of Electrical and Computer Engineering at George Washington University. His research interests include IP-QoS, queuing theory, traffic control, and IP routing. He has published more than 10 papers on related topics. He got his M.S. in Computer Engineering from the Institute of Automation, Chinese Academy of Sciences, in 1996. From 1999 to 2002, he was a senior network engineer at WorldCom, where he was a lead engineer on various IP-QoS, MPLS and IP-VPN projects, and coauthored four Internet drafts and seven U.S. patent applications.

INTRODUCTION



THE PURPOSE OF THIS BOOK

Why did we decide to update this book on ATM once again? Mainly, because the publisher asked us to! Seriously, though, in the fast moving telecommunications industry, a lot has happened since the publication of the last edition in 1998. In case you have been asleep, Internet-based communication is clearly *the* killer application for networking. Much effort is being expended for it to support an ever-broader range of communications applications in a more cost-effective manner. During the early part of the Internet growth spurt in the mid-1990s, ATM was an essential technology employed by Internet service providers to provide higher-speed switching than the routers of that time could support. However, since ATM was not designed specifically to support IP, and was actually somewhat inefficient in doing so, there arose a strong motivation to take the best parts of ATM and put them into a protocol specifically designed to provide a high performance, cost effective infrastructure for IP. The result of that effort has become known as Multiprotocol Label Switching (MPLS). This is the reason that this acronym now shares the title of this edition with ATM.

Therefore, we chose to add to this edition an extensive amount of new material on MPLS, which was in a formative stage back in 1998. Because of its heritage of providing better support for IP networks, MPLS shares some important characteristics with ATM, but also has some important differences. Similarities include support of traffic engineering, Quality of Service, and the use of signaling protocols to

establish efficient switching using locally significant labels. However, ATM was envisioned at the outset with a multi-service mindset that would support any previously-conceived communication service, and hence has support for things like voice, circuit emulation, and support of Frame Relay designed into it from the ground up. On the other hand, MPLS was designed specifically to support IP, and hence has a unique set of functions here that ATM does not; for example, a time-to-live counter that helps avoid routing loops. Interestingly, the designers of MPLS have recently focused on a goal similar to the multi-service vision of ATM. These functions are now being added to the MPLS infrastructure, but also consider support of multiple services over IP and not just MPLS.

This book covers aspects of ATM and MPLS in parallel so that the reader can see these similarities and differences, and appreciate the impact they have on the practical application of these approaches in a network context. We now give a brief summary of how the contents of this book have changed from the previous edition, with Chapter 1 providing a more in depth overview of the book.

Part 1 of this edition removes much of the ATM marketing hype of the previous edition. Instead, it provides a more detailed outline of the book, along with more up-to-date motivation and a summary of the standards organizations that produce much of the technical content described in this book.

Part 2 of this edition retains the extensive background information on general communications technology and the historical development of voice and data protocols can be used as a introductory course to communications or as a practical reference guide for the practicing professional. It adds significant updates in the areas of Frame Relay, Ethernet, and IP, and removes some details for other protocols like X.25, FDDI, and SMDS that are in the sunset of application. Most commercial Frame Relay networks run over ATM, and therefore this is an area of focus of this edition. We chose to continue to dedicate many pages to these descriptions of services that were an integral part of the multi-service vision of ATM, which is now being adopted by MPLS and IP.

Part 3 covers the basics of ATM and MPLS, starting at the physical layer and moving up through the protocol stack to functions necessary to support a multi-service networking environment. This includes not only those functions necessary to forward ATM cells or MPLS packets, but also those necessary to determine the route and signal the association of labels to the path that these cells or packets follow. Support for higher-layer applications over ATM has seen somewhat limited application, and the coverage of these areas in Part 4 is reduced to make room for new material on how MPLS and IP networks could potentially achieve the multi-service vision originally envisioned by the designers of ATM.

Also expanded on in this edition in Part 5 are updates on the hallmark of ATM—traffic management and Quality of Service (QoS). This edition adds material on the initial application of these concepts in IP and MPLS networks. Part 6 contains an introduction to basic communications engineering as well as some updates to traffic engineering extended to apply to MPLS and IP as well as ATM networks. As is often the case in many communication technologies, network management is often the last subject addressed, and MPLS networks are no exception. Because ATM is relatively mature, the standards and approaches for managing ATM-based networks have also matured and Part 7 updates

this information. However, the same cannot be said for MPLS and therefore we outline the potential directions under investigation at publication time.

When the ancient Chinese said, “May you live in interesting times,” they meant it as a curse. Their bureaucratic society abhorred change, preferring the comfort of stability over the uncertainty of progress. Part 8 explores the current era of interesting times in communication and computer networking. This final part contains mostly new material. Starting in the wide area network where efficient use of expensive WAN bandwidth is key, the text objectively studies the efficiency of voice, video, and data over ATM and MPLS or IP packet networks. We also consider the more difficult-to-quantify subjects of complexity, scalability, and reliability, moving into the local area network, where equipment price and simplicity are key considerations, because bandwidth is much less expensive in the local area when compared with the wide area. An interesting divide is the Metropolitan area network, where new applications of ATM and MPLS are being designed and deployed.

INTENDED AUDIENCE

This book can be used as an up-to-date comprehensive textbook on communications networking, since it covers much more than just ATM and MPLS. We have taken this approach, since both ATM and MPLS have adopted the charter of supporting multiple services. In order to understand how this is done, a complete treatment must describe each of the multiple services that are supported. It focuses on protocols, operation, standards, technology, and services for use by the communications manager, network design engineer, practicing professional, or student of data networking. This book also captures important historical aspects of the development of these technologies. In general, we provide a summary augmented by an extensive list of technical references for the reader who wishes to further delve into a particular subject.

The reader should have some prior knowledge of telecommunications principles, although most of the basic concepts of communication networking are covered in Part 2. Not only will the technical professional benefit from this book, but sales and marketing, end users of the technology, and executives will gain a more in-depth view of how ATM and MPLS technology and services can impact their businesses. This book should also help practicing engineers become well-versed in the principles and empower them to communicate these principles effectively to their management. While we strove to keep the text accurate up to the time of publication, the reader is urged to use the references provided to confirm information and obtain the latest published standards.

HOW TO USE THIS BOOK FOR COURSES

This book can be used to teach a single-semester course focused on MPLS and/or ATM, or as a two-semester course on data communications with a focus in the second semester on

the details of MPLS and/or ATM. It can be used as an intermediate-level text for data communications, or can be used as a companion volume when used with an introductory book.

If the subject matter is to be taught over two semesters, we recommend that the text be broken into two parts. Material for use in a first-semester course on an introduction to data communications and basic architectures, protocols, technologies, and services could include Parts 1, 2, 3, and 4. Chapters of focus for a second-semester course on advanced MPLS and ATM protocols and technologies could cover Parts 5, 6, 7 and a recap of Part 4, with either selected outside reading or a research assignment.

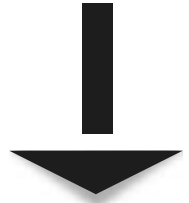
A single-semester course dedicated to data communications services (circuit switching, Frame Relay, Ethernet, IP, ATM and MPLS) focusing on MPLS and/or ATM should consider selections from Parts 1, 2, 3, 4 and 5. The student should have a minimum working knowledge of the material contained in Part 2 if this book is used in a single-semester course.

Labs should contain design problems based on the cumulative knowledge gained from the class readings and outside reading assignments (e.g., recent technology updates or application notes from vendor Web sites). Assigned exercises should involve multiple end-system and intermediate-system design problems. Because of the fluid nature of emerging standards, students should be encouraged to use the text as a working document, noting any changes as the standards from the sources listed in the appendices are revised and updated. This is your book—write in it!

AUTHORS' DISCLAIMER

Accurate and timely information as of the date of publication was provided. Some of the standards we've used were merely drafts at the time of writing, and we assumed that they would become approved standards by the time of publication. At times, we present material that is practical for a large-scale design, but must be scaled down for a smaller enterprise environment. Many data communications networks will operate and continue to run quite well on a dedicated private line network, but eventually the economics of switched technologies and services, even on the smallest scale, are worth investigating. Please excuse the assumption that the user is ready for these advanced technologies—in some cases it may take some time before these technologies can be implemented.

PART



Overview, Introduction, Background, Motivation, and Standards

The first chapter provides a brief introduction to ATM and MPLS, summarizing the various aspects of the technology, including protocols, multi-service support, and network design and operation. We then provide an overview in the form of a summary outline of the remainder of the book so that the reader can use this as a guide from which to continue reading, as well as make use of it as a reference for finding material on a particular subject. Chapter 2 then provides additional background and motivation for ATM and MPLS networking, and the

multiple services for which they provide infrastructure. Finally, Chapter 3 summarizes the standards bodies active in the specification of ATM and MPLS protocols, along with other protocols that state how they support other services and applications. Knowing how to get standards and what the respective roles are of the various organizations is essential background for further study.

CHAPTER 1



Introduction to ATM and MPLS and Overview of the Book

What the heck are ATM and MPLS, and why should I care? In short, the answer is that ATM and MPLS are usually infrastructure and not a service for end users or applications. This means that a residential user will probably never have direct access to these protocols as an end-to-end service. Furthermore, only larger enterprises will make use of them as either purchased from a service provider or as infrastructure within a privately owned and managed network. Electronic equipment (e.g., a switch or a router) implements ATM and MPLS functions on interfaces that connect to transmission systems. A network is a set of ATM or MPLS nodes containing such equipment connected by transmission links, like that shown in Figure 1-1. These nodes exchange digitally encoded messages over these lines, either for the purpose of forwarding packets of information to a specific destination, or for internal control and management purposes. These nodes may also have external interfaces that provide other services to end users or customers. As shown in the upper left-hand corner of the figure, each ATM cell or MPLS packet of information has a header that has an identifier, or label, that determines what packet forwarding action the next node should take. In the example of the figure, originating node 1 prepends the label *A* to some information and sends it to node 2. This node has an entry in its forwarding table that indicates that packets received with label *A* are to be sent to node 4, and that the label should be changed to *B* before doing so. In ATM and MPLS, control protocols form an association of a sequence of label forwarding actions along a sequence of nodes between a source and a destination, forming what is called an ATM Virtual Connection (VC) or an MPLS label switched path (LSP), as shown at the top of the figure.

Okay, if we just told you what ATM and MPLS are in a paragraph and one drawing, then why is there an entire huge book on the subject? The answer is that actually implementing the relatively simple networking concept just described turns out to have a rather intricate and complex solution. Why is the solution so complicated, you ask. The answer has several dimensions, all related to the addition of complexity in order to

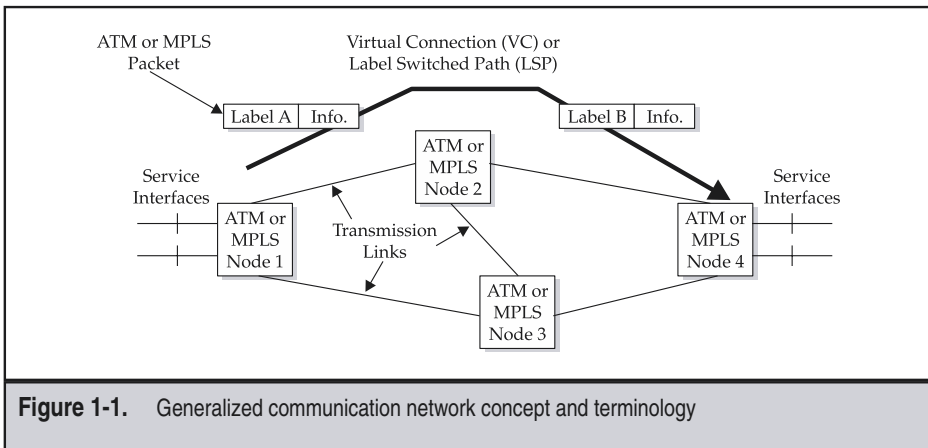


Figure 1-1. Generalized communication network concept and terminology

reduce overall cost. One important aspect of complexity lies in the fact that implementing label switching at high speeds, like those encountered in the backbone of the Internet, is a challenging electro-optical design problem. In order to keep costs in line, engineers must define incredibly detailed standards for interfaces between nodes (and even interfaces between chips within the same node) to allow manufacturers to specialize in developing certain components and systems and achieve certain economies of scale.

But this isn't even the really hard part. Unless one is careful in designing a data communications network, the costs of paying for people to operate the network and/or developing software just to maintain it can be greater than the cost of the ATM or MPLS nodes and the transmission links that connect them. Designers have addressed this problem by developing control protocols that automate many of the tasks involved in operating a network. For ATM, many of these protocols are derived from the telephone network, while for MPLS, many of the protocols are based upon those originally developed to run the Internet. For example, questions of how to automate discovering what the other nodes and links are in a large network, determining the best path for labeled packets to traverse, and signaling the configuration of this path are all complicated algorithmic problems. Add to this the fact that these algorithms are implemented as computer communication protocols in software, which must have voluminous specifications such that nodes manufactured by different companies and/or networks operated by different organizations can understand each other. And as history has taught us, a large computer program is an incredibly complex thing to develop, maintain, and operate. Progress is continually made in the area of protocol development, since a large number of vendors and service providers have a compelling interest to make it all work automatically, because automation achieves a tremendous cost savings as compared with manual configuration. And in fact, automation is essential for networks beyond even a modest size in order to implement packet-switched communication at all.

Unfortunately, the drivers for adding complexity don't stop there. Since ATM and MPLS are usually an infrastructure, and not a native service meaningful to an end-user device, engineers must precisely agree on how to adapt a native service protocol (e.g., IP) to a specific configuration of ATM or MPLS. Often this adaptation is itself also another form of infrastructure within a service provider or enterprise network. Inside a network, there are often many ATM or MPLS paths that can compete for resources at a node or transmission line. Therefore, a whole science of applied algorithms has been defined to route traffic to achieve a desired performance level, or Quality of Service (QoS). This can be quite important, because some applications need a certain level of QoS in order to function properly. There is also a driver to have complex routing algorithms to minimize cost of expensive interfaces on ATM or MPLS nodes and the transmission links that connect them. And once we've defined the solution to all of these problems, there is always a need to do some amount of configuration, and based upon similar motivations there is a strong desire to automate such activities to a certain extent to reduce ongoing operational costs. Finally, within such a complex networking environment, it is inevitable that at some point something will go wrong. Therefore, a whole suite of management approaches and protocols have been defined so that organizations can operate, manage, and diagnose problems in ATM and MPLS networks.

Therefore, the trick is to define just the right amount of complexity that gets the job done efficiently and reliably. The description of the various aspects of complexity in ATM and MPLS just given is essentially the outline of the book beginning at Part 3. The balancing act between complexity and cost effectiveness is a challenging engineering trade-off and one that changes over time. Change has a number of sources. New science or designs can drive fundamental technological advances. Other changes occur as a result of market forces when the industry adopts a de facto standard set by some vendor or service provider. Also, regulation can play a role in changing the telecommunications geo-economic landscape. Therefore, there is also a natural evolution of packet-switching technology that is driven by these changes as well as the relentless human drive for ongoing innovation and improvement. Communications engineers have been working on refining the solution to the same basic problem of labeled packet switching for almost three decades, and just summarizing this history takes several hundred pages in Part 2 of this edition. In fact, many of the services supported in the multi-service vision of ATM and now MPLS are these legacy services. Continuing to support legacy services on next-generation infrastructure was an important tenet of ATM that is now being carried forward by MPLS.

So, we have a lot of material to cover to completely address the complexity defined to achieve a reasonable level of cost effectiveness just described. Since this is a large set of subjects, each with quite a bit of detail, the remainder of this chapter provides an overview of the contents of the rest of this book as a guide for the reader. This is useful to read over to get an understanding of the way we have organized the material to help you in deciding in what order to read these chapters. Furthermore, since many chapters cross-reference the material in other chapters, having an understanding of the outline should help you find information more easily.

OVERVIEW OF THIS BOOK

This book not only reviews highlights of standards, but it also applies the concepts through illustrations, examples, and real-world applications to make the transition from theory to application. It strives to teach the reader not only what the ATM- and MPLS-based technologies involve, but also why each subject is important. Since the Internet has become the de facto networking protocol standard, we focus a great deal of material on how ATM and MPLS apply to the Internet, as well as to intranets and extranets.

This text covers the three critical aspects of ATM and MPLS: drivers, technology, and practical, hands-on application. It interleaves descriptive material replete with many drawings with application notes along with the results of actual networking experience. The book gives examples for network planners, development engineers, network designers, end users, and network managers. The text cites numerous references to complementary texts and sources for the reader interested in more detail.

ATM and MPLS: Theory and Application: Foundations of Multi-Service Networking is arranged in eight parts, each with several chapters.

Part 1 provides an overview and introduction, along with motivation and background. It also summarizes important ATM and MPLS standards bodies.

Chapter 1 provides a high-level introduction to ATM and MPLS, differentiating their role from other protocols as a foundational network infrastructure and not an end-user service. It also contains an overview, or roadmap, for this rather long book. Refer back here or to the table of contents if you are looking for specific information.

Chapter 2 summarizes some of the motivations for ATM and MPLS. It summarizes the changing environment of computing and communication networking and how this affects the evolving corporate communications environment. The scope ranges from the desktop to the local area network and client/server networking domains, to the rapidly growing world of the Internet, intranets, and extranets. An important trend is that changing operational and competitive paradigms demand higher performance communications at reduced unit costs. Further benefits, such as integration savings, flexibility, and economies of scale of multi-service networking, are also important. There is also a set of technology trends that shape the development of solutions in response to these needs, including processor enhancements, modernized transmission networks, and decreasing switch and router costs.

Chapter 3 describes the ATM and MPLS standards bodies and the manner in which they define the standards and specifications, complete with references to the major standards used in this book and how to acquire them. It summarizes the standards process and how standards affect real-world implementations and networks.

Part 2 presents a comprehensive background on communications networking and protocols, and can be used for a course on these subjects. This includes the basic concepts of multiplexing and switching, an introduction to layered protocol models, and tutorials on the major communication networking techniques in use today in the networking environments.

Chapter 4 covers basics of network topologies, circuit types and services, and asynchronous and synchronous transmission methods. The definitions of the terms asynchronous and synchronous are covered in detail. This chapter concludes with a comprehensive review of the principles of multiplexing and switching, with key points illustrated through examples.

Chapter 5 begins with a brief history of packet switching. It then introduces the basic protocol layering concepts used throughout the book. The text then discusses several layered protocol architectures as applied examples of layered protocols, for example, open systems interconnection (OSI) and the Internet. It also presents a discussion of connectionless and connection-oriented data services.

Chapter 6 then introduces the connection-oriented digital Time Division Multiplexing (TDM) communication technique widely used in contemporary voice and private line networks. The text then moves on to an in-depth review of one of ATM's key ancestors: the Narrowband Integrated Services Digital Network (N-ISDN) protocol stack. Here, the reader is introduced to the concept of multiple planes of layered protocols serving the user, control, and management functions.

Chapter 7 covers the key connection-oriented packet-switching protocols in use today: X.25 and Frame Relay, with a focus on the latter. The text gives the origins of each protocol, their structure, and protocol details. Examples illustrate key points of operation for each protocol. The text separates the description of the user and control plane protocol stacks for Frame Relay as an introduction to a similar separation of function employed in ATM and MPLS.

Chapter 8 describes the key connectionless packet switching protocols defined for use in communication networks. The focus is on the Internet Protocol (IP), with some historical information also provided for Switched Multimegabit Data Service (SMDS). This chapter traces the background of each protocol, details the packet formats, and illustrates key operational aspects through examples. The text not only covers IP, but summarizes the entire Internet protocol suite composed of transport and other major application protocols that support e-mail, the Web, and streaming media applications like voice and video.

Chapter 9 presents a tutorial on bridging and routing as background for applications described in Part 4 that support these important services. The text first introduces basic terminology and concepts, some of which have been adopted in ATM and MPLS control protocols. It then describes commonly used LAN protocols like Ethernet, Token Ring, and FDDI. The text then introduces the concepts of routing and addressing.

Part 3 covers the basics of the ATM and MPLS protocol landscape, providing a structured introduction and reference to all terminology, protocols, and standards.

Chapter 10 introduces the overall broadband ISDN (B-ISDN) protocol reference model in terms of the user, control, and management planes. The layers common to all of these planes are physical, ATM, and ATM Adaptation Layer (AAL). It then provides a high-level introduction to ATM. We then summarize the origins of MPLS as a method of improving on the traffic engineering of early IP over ATM networks. It then provides an introduction to MPLS. This chapter concludes with a discussion of consideration of the choice between fixed-length ATM cells and variable-length packets.

Chapter 11 details the physical layer and ATM layer and corresponding MPLS protocols. The text describes how a single ATM layer operates over a number of physical media. It also introduces the concepts of the ATM traffic contract, ATM service categories, and Quality of Service (QoS), leaving further details to Part 5. This chapter covers the manner in which MPLS labels are encoded over various physical and logical networks. In particular, it describes how MPLS can run over SONET, ATM, Frame Relay, or Ethernet.

Chapter 12 describes the ATM Adaptation Layer (AAL), which provides support for all higher-layer services, such as signaling, circuit emulation, Frame Relay, and IP. Since MPLS standards are not yet complete in this area, this chapter summarizes the current state and direction of MPLS infrastructure being defined with the aim of supporting multiple services. It describes a particular proposed encapsulation method as an example of the type of protocol that will likely be standardized at some point. We also summarize specific approaches proposed for supporting ATM over MPLS in this chapter.

Chapter 13 introduces the higher layers in the user plane in the WAN, the LAN, and the internetwork; control plane signaling and routing for ATM and MPLS. This chapter then introduces the ATM control plane and its AAL and underlying structure. This

chapter covers the ATM user-network interface (UNI) protocols, leaving the coverage of network-network interface (NNI) protocols to Chapter 15. It also summarizes ATM addressing and name services.

Chapter 14 describes the concepts and control protocols involved in MPLS networks. These include protocols that exchange routing information and those that signal the establishment of LSPs. The chapter concludes with several real-world examples of the use of MPLS as infrastructure for traffic engineering, VPNs, and inter-service provider LSPs.

Chapter 15 covers the network-network interface (NNI) protocols defined for ATM. The focus is on the Private/Public Network-to-Network Interface (PNNI) routing protocol defined by the ATM Forum. This chapter also describes the control plane protocols employed between ATM service provider networks.

Part 4 covers ATM and MPLS support for higher-layer protocols in the user plane in the WAN and LAN.

Chapter 16 covers the support by ATM and MPLS of voice, TDM, and video. The text summarizes the important aspects of voice over packet technologies, and summarizes the work on Voice and Telephony over ATM (VTOA) performed by the ATM Forum and the related ITU work. It also summarizes the support for voice over MPLS trunking, which is effectively a simplified version of voice over ATM. It then covers the emulation of Time-Division Multiplexed (TDM) circuits and interworking with Narrowband ISDN (N-ISDN) over ATM, with a summary of the work ongoing for support over MPLS as well. Finally, the chapter concludes with protocol support for video over ATM and IP.

Chapter 17 defines the concepts of true protocol interworking, use as an access method, or use as a common trunking vehicle. The text then details the two approaches defined for interworking ATM with Frame Relay; one as a direct service translation and the other as a means to trunk FR services over an ATM backbone. Both of these approaches are in wide commercial use. It also summarizes the support for interworking ATM and SMDS. It continues with a summary of frame-based ATM protocols designed to make ATM available to legacy devices or use transmission facilities more efficiently. The chapter concludes with a discussion of proposed MPLS support defined for trunking of Frame Relay.

Chapter 18 first describes standards support for carrying multiple protocols over ATM. It then summarizes the ATM Forum's LAN Emulation (LANE) protocol and the problems that it addressed, and how changes to Ethernet and market forces eclipsed its brief period of enthusiastic adoption. This chapter concludes with a description of emerging standards and deployed services using Ethernet over MPLS in metropolitan and wide area networks.

Chapter 19 first describes the most widely implemented classical IP over ATM protocol for a single logical IP subnetwork. It summarizes the Multiprotocol over ATM (MPOA) and IP multicast over ATM efforts, which were never widely deployed, and extracts lessons that MPLS-based solutions should consider. This chapter focuses primarily on the hot topic of IP over MPLS virtual private networks (VPNs). It concludes with a discussion of the important practical topic of negotiating maximum packet sizes over ATM and MPLS networks.

Part 5 provides the reader with an application-oriented view of the ATM and MPLS traffic parameters, congestion control, traffic engineering, and design considerations. Complex principles are presented in a manner intended to be more readable and understandable to a wider audience than in other current publications.

Chapter 20 begins a description of ATM and IP traffic and Quality of Service (QoS) reference models and terminology. Most MPLS traffic and QoS support is inherited directly from IP. It continues with a description of the simplifying concept of ATM layer service categories, summarized by acronyms; for example, CBR, VBR, and UBR, which correspond to constant, variable, and unspecified bit rates, respectively. In parallel, we also define the IP differentiated service (Diffserv) definition of QoS. We then compare the ATM and IP traffic and QoS parameters to illustrate their similarity. This chapter concludes with a description of new ATM service categories designed to better support switching of variable-length packets.

Chapter 21 details the important concept of traffic defined by a set of parameters that are implemented by a leaky bucket for ATM or a token bucket for IP and MPLS. It then describes how policing can enforce and how shaping can ensure compliance with a traffic contract. The coverage continues with a discussion of queuing and scheduling techniques that are often employed to deliver different levels of quality and provide fair allocation of resources to various classes of traffic. This chapter concludes with a description of how admission control ensures fair access to resources, and how hierarchical label switching can result in a more scalable network design.

Chapter 22 covers the important topic of congestion control. It begins by defining congestion and its impact. The chapter then presents a number of solutions and their range of applications, including the levels of congestion that can occur and the types of responses that exist; the standard method of selective discard; and long-term control by use of resource allocation, network engineering, and management. The chapter concludes with a review of the closed loop flow control technique, and an in-depth description of the ATM-based technique called Available Bit Rate (ABR).

Part 6 is the technical center of the book, providing practical guidelines and design formulas that afford considerable insight into the benefits and applicability of ATM or MPLS. As such, it is rather detailed and makes extensive use of mathematics. As much as possible, we try to give a “cookbook”-style solution that allows the reader to do some basic calculations regarding their application need. The text gives references to other books and periodicals containing further detail.

Chapter 23 covers the basic philosophy of communications engineering. Central to this discipline is the notion of random processes that model real-world events like errors, queuing, and loss. The chapter describes how electrical and optical communication signals occupy a specific frequency spectrum that determines overall transmission capacity. In the presence of noise, Shannon’s channel capacity theorem defines the highest achievable transmission rate on a specific channel. The text then describes how a communications system may employ error correction or detection coding to achieve improved performance.

Chapter 24 provides an in-depth overview of traffic engineering philosophy, basic queuing models, and approximate performance analysis for delay and loss in ATM and

MPLS devices and networks. It includes models for estimating blocking, queuing delay, and queuing loss performance. It also presents an in-depth discussion regarding statistical multiplexing gain and equivalent bandwidth that is the mathematical foundation for traffic engineering. It also summarizes priority queuing performance and summarizes the impact of self-similar traffic on performance. The emphasis is not on mathematical rigor, but rather on usable approximations and illustration of key trade-offs.

Chapter 25 discusses additional practical network design considerations. We begin with a discussion of the impact of loss, delay, and delay variation on various applications. The coverage then moves to summarize guidelines on the best way to run TCP/IP over ATM. The text then provides a model for quantifying the potential savings of voice and data convergence. The chapter concludes with a review of the network planning and design process and the tools available to support these important activities.

Part 7 provides the reader a view of operations, network management architectures, standards, network management protocols, and performance measurement.

Chapter 26 begins with the topic of Operations, Administration, Maintenance, and Provisioning (OAM&P) philosophy. It concludes with a description and comparison of the various network management architectural approaches for managing ATM and MPLS networks.

Chapter 27 continues the theme of network management, but at the next level closer to the actual network elements. The text describes the IETF's Simple Network Management Protocol (SNMP) and the ITU-T's Common Management Interface Protocol (CMIP). It continues with a summary of important Management Information Bases (MIBs) defined to allow configuration and operation of ATM and MPLS network elements. This includes a detailed description of the ATM Forum's Integrated Local Management Interface (ILMI) and the IETF ATM MIB (AToMMIB). It concludes with a description of how IP-based network management tools have been applied to deployed MPLS networks.

Chapter 28 introduces ATM layer and fault management architecture, which is largely based upon the philosophy involved in operating TDM networks. The text presents ATM Operations and Maintenance (OAM) cell structures for use in fault detection and identification, as well as the measurement of network performance. The chapter describes methods to ensure that a network meets specific QoS objectives through the use of performance measurement OAM cells. At the time of writing, some of these same approaches were under consideration for extending the scope of MPLS management functions, and this chapter summarizes these proposals.

Part 8 provides a comparison of technologies that complement and compete with each other in the wide area and local area network environments. The book concludes with a look at the future of MPLS and ATM.

Chapter 29 presents an objective comparison of ATM-, MPLS-, and IP-based approaches to multi-service networking primarily along the dimensions of efficiency, scalability, and complexity. A key requirement for the WAN is efficient use of expensive bandwidth, and toward this end, this chapter presents an objective efficiency analysis of ATM and MPLS versus their competitors for support of voice, video, and data services. We continue by discussing some objective measures of scalability. Complexity is more

difficult to quantify, but we also discuss this important aspect, since it is often such a critical factor. This chapter also discusses other design considerations, like reliability, availability, stability, supportability, and security.

Chapter 30 concludes the book with a review of the prospects for the future use of MPLS and ATM networking. The future for ATM is more certain, while the prospects for MPLS are much broader. We summarize important lessons learned in the development of MPLS from ATM and other protocols. The chapter includes an overview of hot topics like optical networking, generalized MPLS (GMPLS), and the separation of control and forwarding. The chapter concludes with a presentation of several possible future networking scenarios as food for thought.

ATM and MPLS have a unique set of terminology, replete with many acronyms. To aid as a reference in deciphering what amounts to a language of its own, this book also contains two appendixes and an index. *Appendix A* lists the major acronyms and abbreviations used in the book, while *Appendix B* provides a list of references cited in the body of the book.

REVIEW

We introduced ATM and MPLS as an infrastructure primarily used by service providers, which in some cases is used as a service by larger enterprise customers. They are both simplified packet-switching technologies that can be operated at high speeds as detailed in standards. Although simple in concept, both ATM and MPLS require a significant amount of complexity in order to make them cost effective in a large-scale network. For example, the automatic configuration of label-switched paths is a sophisticated, complex set of protocols unto itself. Also, there must be some form of adaptation between the ATM or MPLS infrastructure protocol and one that is meaningful to an end device, for example, IP, in order to achieve the multi-service vision to which both ATM and MPLS strive. Additionally, there is a need to ensure that just the right level of traffic is sent through a particular node or over a transmission line to ensure that the performance necessary for an application to work correctly is delivered. Finally, even with the best-laid plans there is a need to do some configuration, and even more importantly, figure out what went wrong and fix it. Therefore, other protocols must be defined to manage the nodes in an ATM or MPLS network. The outline of this book is essentially the description of how ATM and MPLS meet each of the challenges just described.

CHAPTER 2



Background and Motivation for ATM and MPLS Networking

This chapter provides some background and motivation for the use of ATM and MPLS protocols to meet various service provider, enterprise, and consumer needs. We begin by extracting some relevant drivers in a brief review of the history of communications. Next, the text moves on to define various aspects of demand. Finally, the chapter concludes with a summary of important technology trends that enable ATM and MPLS networking.

A BRIEF HISTORY OF COMMUNICATIONS

Often, our coverage of a particular topic is in historical order. This is because we believe that many lessons from the past are carried forward into the next generation of technology. The newer invention retains the good qualities of the old and discards the bad. In other words, if it isn't broken, then don't fix it. And, if it is broken, then try to fix it without introducing too many new bugs. Furthermore, the newcomer technology or protocol also must bring some important innovation, enhancement, or simplification; otherwise, it will never be successful. The historical account of the evolution of communications protocols shows this pattern time and again. Studying the reasons why some approaches work better for some situations than others can yield deep insights that can help us make better decisions going forward. Additionally, understanding the mistakes of others allows one to move ahead more productively by avoiding those same mistakes. In a complementary manner, knowing that a particular approach is a tried and true solution to a particular type of problem is reassuring—that is, at least until someone comes up with a better idea.

The history of human communication is a long one, and there are several ways of looking at it to try to discern a pattern and derive guidance and knowledge from our predecessors. The following sections present several perspectives.

Recurring Trends in Encoding and Relaying

One view of the history of communication is along the dimensions of analog versus digital encoding and synchronous versus asynchronous relaying, or scheduling. Speech and the visual arts are forms of analog-encoded information, while writing and computer signals are digitally encoded information. Communication is the process of transferring information. Analog communication was the earliest and is the most natural for many people. The beginnings of spoken analog human communication are possibly 100,000 years old. Graphic images over 30,000 years old have been found in caves. We have been talking and showing things to each other ever since.

Written records from ancient Syrian civilizations scribed over 5000 years old in a discrete cuneiform alphabet on clay tablets mark the beginnings of digital communication. Soon thereafter, messengers conveyed this written digital information, and eventually this was institutionalized as a postal service. Although usually viewed as a modern phenomenon, digital long-distance optical communications began over two millennia ago when the ancient Greeks used line-of-sight optical communications to relay information using placement of torches on towers at relay stations. The practice continued through

the seventeenth and eighteenth centuries when optical telegraphy was extensively used in Europe. That is, until someone came along with a better idea when Samuel F.B. Morse invented electrical telegraphy in 1846, marking the beginning of modern digital electromagnetic communications. Marconi invented radio telegraphy shortly afterward, enabling digital communication at sea and providing access to remote areas. Broadcast analog radio communications of audio signals followed in the late nineteenth and early twentieth centuries. Telephone companies applied this technology to analog voice communication in the same time frame. Television signal broadcasting began in 1928 and became commercially viable in the late 1940s, with color broadcasts occurring in the 1960s. In the 1950s, the conversion of analog voice to digital signals began in large metropolitan areas to make better use of installed cabling.

This was followed by the invention of packet switching in the 1960s as an offshoot from research into secure and highly resilient military communication networks. Packet switching includes many technologies covered in detail in the next part, including X.25, Frame Relay, and the Internet Protocol (IP). ATM and MPLS are the latest ancestors of packet switching in the 1990s. Fiber optic transmission and the new concept of synchronous digital transmission introduced in the early 1980s moved the unceasing progress of technology ahead another notch. From this we learned about the complexity of building a distributed system with stringent timing synchronization requirements. The next major leap in technology in the 1990s was wavelength division multiplexing (WDM), which is analog and asynchronous. Switching an entire optical line or a single wavelength is a hot topic early in the twenty-first century that promises to service the projected geometrically increasing demand for capacity fueled by Internet-based applications.

Data Networking: Enabling Computers to Communicate

Once an organization had more than one computer and more than a few users, an urgent need arose to interconnect these computers, which was the birth of data communications. Beginning in the 1960s, mainframe-based networks accessed by local and remote terminals evolved through the use of private networks and packet-switched services. For example, IBM's Systems Network Architecture (SNA) provided the means for many dumb terminals to communicate with an intelligent host or mainframe in a hierarchical fashion. This hierarchy developed because collecting expensive intelligence at the host and making the terminals dumb was the most cost-effective solution at the time.

The rise of first minicomputers and then the personal computer (PC) ushered in the era of modern data communications. It also ushered in the client/server computing paradigm, where a not-so-dumb client terminal could do some stand-alone operations, relying on a departmental server for data or other services, such as shared printers or mass storage. In order to interconnect these devices, local area networks (LANs) were the next major development in the computer communications networking with the invention of Ethernet by Xerox in 1974. Just as minicomputers invaded mainframe turf when the cost fell to departmental budget approval levels, so too did PCs, LANs, bridges, and routers invade the minicomputer turf. Initially, these LANs sprang up as disparate islands run by entrepreneurial departments, but then the need arose to share the information. As

described in Chapter 9, first bridges and then routers enabled interdepartmental connectivity of diverse computing resources in a cost-effective manner.

When the need arose to interconnect these LANs, the mainframe manager and entrepreneurial LAN managers had to work together in order to provide access from the LANs to the mainframes. In addition, WAN interconnectivity often went beyond the scope of a single LAN manager because costs had to be shared across multiple LANs, and here too cooperation was necessary. The router also found its place here as the gateway from the LAN to the WAN to fill this need, as well as the device that interconnected disparate LANs running a common network layer protocol. Of course, the latest news is how the Internet burst from relative obscurity as a research network to become the de facto standard in networking.

Changing Organizations of People and Networks

Computer communications networks evolved over the last 30 years from centralized mainframe computing, through the departmental minicomputer era, into the current era of the distributed client/server processing. Traditionally, these data networks and the organizations that ran them were completely separate from voice communications networks and organizations. However, with technology offering the potential for convergence of voice and data networks, this tradition is now changing as well.

A great parallel is occurring in the evolution of networks and organizations. Networks and entire corporate organizations are making the transition from hierarchical to distributed structures, requiring both greater interconnection and more productivity from each individual or group. We call this move from hierarchical to distributed structure *flattening*. In networking terminology, flattening means the creation of fewer network elements with greater logical interconnection. In organizations, flattening often refers to the reduction of middle management, which requires greater horizontal communication and interaction within the organization. This trend continues as executives reengineer the corporation, which in turn requires reengineering of the computing and communications network support infrastructure.

With an ever-increasing number of two-income households, more enterprises strive to accommodate work at home and telecommuting. The corporate world now realizes that the quality of life and productivity improvements achieved by allowing people to work from home, instead of fighting through hours of rush hour traffic, pay off in the long run. Here, too, technology has been an enabler by offering higher-performance access technologies like cable modem and digital subscriber line (DSL).

Evolving corporate infrastructures require communications networking flexibility to respond to ever-changing business needs. Reorganizations, mergers, and layoffs frequently place information workers in different physical locations, accessing different network resources. These factors drive network designers to distribute the storage and processing of data to servers supporting groups of clients. The result is that an increasing amount of computing occurs via distributed processing using the client/server paradigm, which is precisely the model of the Internet.

Several technologies enable and drive this change. *Distributed processing* involves the location of network intelligence (i.e., processing) to many network sites, where each site communicates on a peer-to-peer level, rather than through a centralized hierarchy. *Client/server computing* distributes the actual storage and processing of information to many server sites as opposed to storing and processing all information at a single, centralized location. Servers provide the means for multiple clients to share applications and data within a logical, or virtual, workgroup. Servers also allow users to share expensive resources, such as printers, CD-ROM juke boxes, mass storage, high-speed Internet connections, Web servers, and centralized databases.

DEFINING THE DEMAND FOR COMMUNICATIONS

What creates demand for communication services? What are the killer applications? What human activities or automated system applications have an insatiable thirst for the capabilities that only MPLS or ATM can provide? Part of the answer peeks out at us from the experience of the World Wide Web (WWW). Web browsers evolved from interesting demos in the early 1990s to slick professional programs on which the twenty-first-century world now relies. Beginning as simple text pages with simple graphics, contemporary Web applications sport animated graphics, audio, video, and an increasing set of interactive applications downloaded in real time. Furthermore, the tremendous popularity of cable television and wireless devices also shows a possible source of the future traffic. If people spent only a fraction of the time spent watching television on multimedia Internet communications, then the demand would be a thousand times the current level of Internet traffic. Also, there are more wireless access devices than there are traditional Internet users. The experience of wireless communications teaches us that convenience and mobility are important features of communication services.

Residential and Commercial Users

Traditionally, communication users have been placed into one of two broad categories in terms of the types of services and capacity requirements they demand: residential or commercial. A residential user ranges from the household with dial-up modem access to the Internet up to people with broadband Internet and/or ISDN access connections. The set of commercial users overlaps residential use (e.g., the single-occupant home office) up through access connections that support multiple gigabits per second. There is also a difference in the set of services that a residential user wants (or can afford), as compared with a commercial customer. In general, residential users often want just the most basic services, usually from only a single location or device. On the other hand, commercial users are the ones to demand the most sophisticated services, sometimes requiring simultaneous support for hundreds or even thousands of sites and devices. Commercial users are candidates for use of native ATM or MPLS services, or services directly based upon these technologies, while on the other hand, MPLS and/or ATM can provide the infrastructure for residential as well as commercial services.

Applications and Networks Change Faster Than Behavior

It is often easier to get people to perform existing tasks using a better tool than it is to get them to change their overall behavior. This has been a hallmark of the historical innovation of communications. For example, telegraphy was a major disruptive technology in that it decoupled communication from transportation [Odlyzko 01]. No longer did communication travel at the rate at which a messenger or postal system could transport packages—a short message could be sent across long distances in seconds to minutes. People still communicated messages, but telegraphy made this happen much more rapidly. E-mail and the Web are similar innovations in that an incredibly detailed message can be sent or retrieved almost instantaneously. This is a huge step beyond physical delivery of a document in the ability to convey information. However, in the end, the basic human behavior of generating a document, distributing it, and reading it have not changed—only the applications and networking involved in this process have. In a similar vein, people will still always need to talk to each other, and e-mail and the Web will never replace that. Therefore, there is a tremendous drive to support human voice communication over IP, MPLS, and ATM networks.

Geographical Aspects of Networking

There is also a strong geographic context to communications. The cost of connecting computing components is much less on your desktop than across the desks in your immediate local area network (LAN). Similarly, it is less expensive to connect sites within the same metropolitan area than it is to get connections across a country or a continent. And finally, when communication must occur across transoceanic cables, the costs can be quite high. In a very real sense, communication begins at the desktop, and users have come to expect that capabilities and performance for mission-critical applications available only at their desks a few years ago will be available almost independent of geography. However, this fact is often mitigated by a geographically defined community of interest based upon the scope of the enterprise, language, national boundaries, or culture.

Typically, two scenarios drive enterprise needs for greater connectivity across the metropolitan or wide area network. The first is an increased need for interconnection between distributed computing devices on remote LANs. The second is the logical extension of LAN-speed communications across wider geographic areas; for example, access to remote intranet servers. Geographically dispersed LANs now have a range of connectivity choices, ranging from dedicated circuits to switched wide area and metropolitan area networks to broadband data services. The choice of WAN technology and services is based upon several factors, which include cost but also features and security. In fact, connecting LANs across the WAN often leads to the hybrid use of private data networks in conjunction with public WAN services in many large enterprises.

The End Result: Tremendous Internet and Data Traffic Growth

Internet traffic growth has doubled annually on average for many years, and enterprise data traffic has consistently grown at a factor of tens of percentage points per year, far outpacing the average growth of voice, which has been less than ten percent per year. Unfortunately for service providers, the revenue for Internet and data services isn't growing as fast as traffic. According to various accounts, the volume of Internet and data communications traffic exceeded that of voice and private line traffic some time in the late 1990s. What is causing Internet and data traffic to grow so much more rapidly than that of voice and traditional private lines? There are many answers: Web browsing, electronic mail, file transfer, local area network interconnection, interactive applications, and emerging multimedia applications are all creating demand for new data services. Think about the amount of time you and your co-workers spend on the phone versus surfing the Web, reading and sending e-mail messages, and sending files across the network. Think back a few years and reflect on how this has changed. Most people would report that their use of data networking increased markedly.

TECHNOLOGY TRENDS

A number of trends in the area of technology drive traffic growth and enable new applications. These include enhancements to the basic components that make up computers and network elements (e.g., switches and routers), namely, processors, protocols, and transmission systems.

Processor and Memory Cost Trends: Moore's Law

One of the biggest trends driving the need for a high-speed network infrastructure like ATM or MPLS is the continued growth of computing power and its use in distributed client/server computing. Processing power (MIPS), memory size (megabytes), and display size (megapixels) capacity are all doubling every 18 months according to Moore's law, but they continue to be available at the same price. This continuing enhancement of the price-performance ratio of computers creates a tremendous demand for Internet and data communication. Furthermore, as discussed earlier, organizational reengineering further increases demand for bandwidth to interconnect these workstations and servers. Mass storage of information has shrunk to a fraction of its original size and cost, so that even a modest machine can be a server (if it is well connected), a fact that further distributes demand.

These technologies not only propel the cost effectiveness of computing performance, but they are also directly applicable to the electronic hardware components that implement

communication equipment, such as ATM- and MPLS-based switches and routers. And, although a doubling of communication speed every 18 months may seem impressive, it turns out not to be enough to keep up with the historical Internet growth rate of annual doubling. We come back to this subject in Chapter 30, where we discuss how MPLS control of optical networking could be the architectural change necessary to solve this scaling problem.

Distributed Computer Communications Protocols

Desktop machines not only have the processing power the centralized host once had, but their operating systems contain the TCP/IP suite of protocols that enable many of the functions known as the Internet. A large set of standards details the formats, procedures, and uses of these protocols that have resulted in a number of interoperable implementations, enabling competition in the IP, MPLS, and ATM marketplace.

Increased memory in the end stations allows protocols to implement larger retransmission windows, and hence achieve increased throughput in high-performance network environments. RFC 1323 increases TCP window size from 64 kilobytes to over 1 gigabyte for this very reason. Increased processing power enables the implementation of more sophisticated flow control and windowing mechanisms. One example of a good complexity trade-off is placement of sophisticated TCP flow control algorithms in the end station, such that the processing can be distributed to the edges, leaving the simpler IP forwarding function to be done by switches or routers in the core network. This is an example of an important tenet of the Internet architecture at work; namely, moving complexity (i.e., processing and storage) as close to the end system as possible.

Older network protocols like X.25 packet switching implemented complex procedures just to ensure that a packet could be reliably sent from node to node, sometimes requiring multiple retransmissions over noisy, error-prone analog links. The simplification of network core switching protocols is primarily a result of essentially error-free physical layer communications over digital facilities replacing the older error-prone analog facilities. The infrequent occurrence of errors and associated retransmission is then achieved much more cost-effectively in end systems. Simpler network protocols, such as Frame Relay, ATM, and MPLS, rely on the performance of digital fiber optic transmission, which provides very low error rates, typically less than 10^{-12} . This means that less processing power is required in intermediate routers, since the complexity of error detection and retransmission can be moved to the end system.

Modernization of Transmission Infrastructures

Fiber optics replaced digital microwave transmission in industrialized nations even more rapidly than digital transmission systems replaced analog systems. Satellite communications have been relegated to serving as a high-quality digital transmission medium for connectivity to remote areas or as a backup to terrestrial facilities. The nationwide and metropolitan area networks of most service providers are almost exclusively fiber based, and many routes employ automatic protection in the event of failure. The potential

bandwidth between cities that could be carried over the hundreds of pairs of optical fibers, each operating at terabit-per-second (10^{12} bps) speeds, have created a capacity glut in certain geographic regions in the early twenty-first century.

Modern digital and fiber optic transmission communications establish a new baseline for the performance of digital data communications, just as digital transmission made long-distance-calling sound quality comparable to local calls in the 1980s. The low error rates and high reliability of digital transmission over fiber optics was an important factor in networking protocol simplification. Furthermore, the cost-effective availability of high-speed digital transmission circuits at rates ranging from millions to billions of bits per second is an important enabler for MPLS and ATM infrastructures.

Faster and Farther, but Never Free

Two complementary and competing phenomena are occurring simultaneously. LAN and WAN speeds are converging such that provision of LAN-speed connectivity across the WAN is becoming technically feasible, although cost is of course dependent on distance. Bandwidth in the WAN and the LAN, when viewed in terms of cost per megabit-per-second, has steadily decreased in cost over time (years), with LAN costs decreasing much more rapidly than WAN costs. The principal trend in LAN technology centers on the highly cost-effective, tried and true Ethernet family of transmission systems, ranging from 10 Mbps to 10 Gbps and using twisted pair at the lower speeds and optical fiber at the higher speeds. Although running fiber to the desktop and home costs only a few cents per foot, it now dominates the overall cost. But here again, creative innovators have found ways to squeeze even more bandwidth out of the existing twisted pairs deployed to major enterprises and many households through cable modem and DSL technologies, as discussed in Chapter 11.

Bandwidth in the WAN, as many pundits projected, is not yet “free” (and never will be), but it is becoming less expensive when delivered at very high volumes between points directly connected by optical fiber. That is, faster circuits are generally less expensive when measured on a cost per bit per second basis. At the time of writing, “fast” means terabit per second speeds (10^{12} bps) across a single fiber optic strand, as achieved through Dense Wavelength Division Multiplexing (DWDM). However, don’t forget that you must fill up such a high-speed circuit to achieve this economic benefit. But aggregating many lower-speed flows into a larger stream is a fundamental concept of network infrastructure, such as ATM or MPLS. Therefore, the enhanced cost effectiveness of increasingly higher-speed digital transmission circuits at rates ranging from millions to billions of bits per second is an important enabler for MPLS and ATM network infrastructures, since this helps a service provider achieve economy of scale in the core of a network.

The Accelerating Bandwidth Principle

As discussed earlier, Moore’s law is based upon the historical trend that a computer’s central processing unit (CPU) speed, as measured in millions of instructions per second (MIPS), doubles every 18 months, with this performance available at a relatively constant

price. (Of course, you have to buy twice as much capacity to get the newest model with the latest features.) During the mainframe era, Gene Amdahl postulated that the average application requires processing cycles, storage, and data communication speeds in roughly equal proportion. For example, an application requiring 1 MIPS also needs 1 megabyte of storage along with 1 Kbps of peak communications bandwidth. Robert Metcalfe (the inventor of Ethernet) postulated that communications demand grows as the square of the number of devices that are in communication with each other. Many experts believe that Metcalfe's prediction is too aggressive, but they agree that demand for communications capacity is growing faster than processing speed. The number of devices in communication has been growing over time due to the change from centralized computing to the client/server paradigm, and most recently the worldwide embrace of the Internet. One example of this trend is the number of hosts connected to the Internet, which has been growing at a historical rate of 55 percent per year [Hobbes 02]. The increases in available computing power from Moore's law and the commensurate need for bandwidth as posited by Amdahl, magnified by the increased communications of an expanding community predicted by Metcalfe, combine to result in what we call the *accelerating bandwidth principle*.

The accelerating bandwidth principle shows that the need for communication grows at a rate faster than the growth of processing speed (and storage capacity) predicted by Moore's law due to the increased communication within an expanding community of interest. Figure 2-1 illustrates an example of the accelerating bandwidth principle, with values plotted on the logarithmic scale on the y-axis versus years on the x-axis. The starting point is a representative set of numbers starting with minicomputers in 1980 of a 1 MIPS CPU requiring 1 Kbps of communication. We assume that the community of communication rate increases at 30 percent per year as a slower acceleration than that postulated by Metcalfe to the linear tracking of computer capacity postulated by Amdahl's law. The curve labeled MIPS/CPU represents the nonlinear doubling in computing power every 18 months predicted by Moore's law. The curve labeled Mbps/MIPS represents the nonlinear growth of the required data communication of approximately 30 percent per year. The curve labeled Mbps/CPU, which is the product of the MIPS/CPU and Mbps/MIPS curves, represents the data communications bandwidth predicted by the accelerating bandwidth principle. The growth rate for the Mbps/CPU value is hyperexponential because the exponent grows at a faster than linear rate due to the combined nonlinear increase in interconnection and computer communications bandwidth. This simple model provides an explanation for how the Web drove Internet and enterprise data network traffic growth experienced beginning in the mid-1990s.

Worldwide Cooperation for Standards

Rapid realization and industry-wide agreement on a common set of standards are essential to achieve interoperability at all levels of the network and across many hardware and software platforms. Interoperability is important for a service provider or enterprise network operator because it enables a choice between vendors. Participation in standards bodies by vendors, service providers, and end users is essential to achieve this goal, as described in the next chapter.

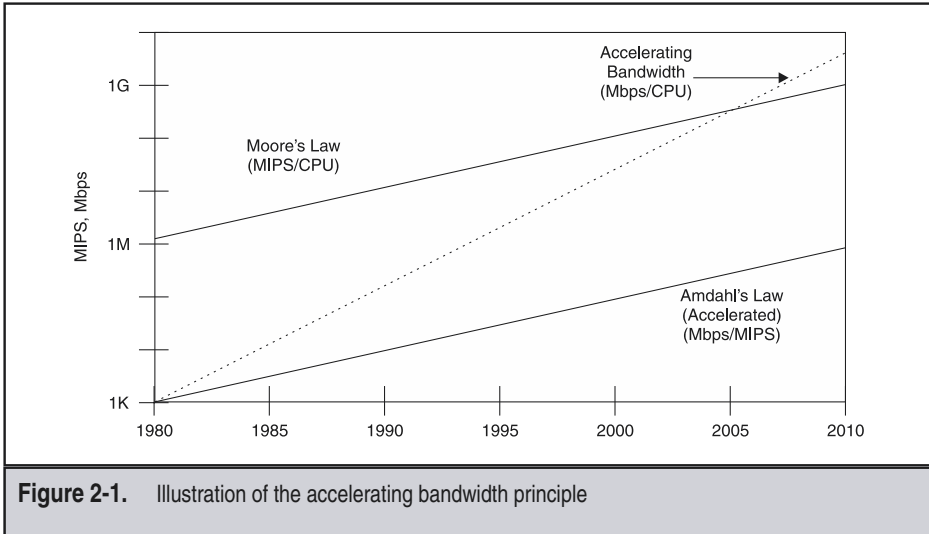


Figure 2-1. Illustration of the accelerating bandwidth principle

Most industry sectors and associated standards bodies now support both ATM and MPLS. This began with the telecommunications industry defining initial B-ISDN and ATM standards in the late 1980s, which saw the development of early, prototype ATM switches. The traditional customer premises multiplexer and switch vendors then adopted ATM in the early 1990s. Next, the Internet standards folks got involved and router, bridge, and workstation manufacturers began building standard ATM interfaces into their products. In fact, it was ATM-standard interfaces that enabled service providers to scale the Internet in the mid-1990s by connecting routers with ATM switches. Vendors developed ATM interface cards for workstations and servers in the latter part of the 1990s, which in retrospect was the peak of ATM development. Now, fewer new standards are being developed for ATM, and much of the focus of ATM is in support of Frame Relay, voice, and circuit emulation.

As described in Chapter 10, the drive that created MPLS was to make a bigger and better Internet. The Internet standards body, service providers, and router vendors were the ones to drive forward this important effort, which took several years. The basic MPLS standards are now in place, and a number of interoperability labs help ensure that real-world implementations work together. But, a good standard is often a process of continuing refinement, and MPLS standards continue to evolve. As described in this book, although MPLS was initially conceived as an infrastructure better suited than ATM for the Internet, the standards efforts of the early twenty-first century have focused on making MPLS support multiple services, essentially trying to solve the same problem that telephony standards had done for ATM. In fact, the standards groups who created ATM are now trying to get involved in the MPLS standards process!

REVIEW

The chapter first summarized how ATM and MPLS are communications of a digital and asynchronous form, similar to the postal services and telegraphy of the past. Also, the change in computer communications from centralized mainframes to distributed personal computers and servers has impacted organizations as well as fueled the demand for communication. There are two broad classes of communication users: residential and commercial, who have different characteristics and needs. In general, applications and networks that let people do what they have been doing, but more productively, are the most successful; the Web and e-mail of the Internet are a case in point. Demand also has a geographical aspect, with a community of interest generally matching that of a user's locality. The end result is the explosive growth of Internet-related traffic. The chapter concludes with a summary of the foundational technology trends of processing, protocols, and transmission that enable the high-speed networking capabilities of ATM and MPLS. Of course, in order to do all of this cost effectively, there must be cooperation that results in interoperable standards, a topic covered in the next chapter.

CHAPTER 3


ATM- and MPLS-Related Standards Bodies



ATM & MPLS Theory & Application: Foundations of Multi-Service Networking

DAVID **MCDYSAN**
DAVE **PAW**

McGraw-Hill/Osborne
New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto



International Telecommunications Union (ITU)

The United Nations founded the International Telecommunications Union (ITU) in 1948 to produce telegraphy and telephone technical, operating, and tariff issue recommendations. Formerly known as the Consultative Committee International Telegraph and Telephone (CCITT), the Telecommunications standardization sector, referred to as the ITU-T, produces recommendations on B-ISDN in which ATM is but one component in an overall set of services. During a two-year study period, the ITU-T assigns questions to study groups. These groups organize into lower-level committees and produce working documents and draft recommendations. Study group 11, for example, covers ATM signaling protocols; while study group 13 is the lead group for IP-related functions, for example, MPLS. For details and to obtain further information about the ITU-T, see their Web site at <http://www.itu.int>.

The ITU-T is a members-only standards organization, with fees charged based upon the class of membership. The ITU-T standards documents are called "Recommendations" but have a much stronger meaning, particularly when interworking between service providers in different countries. Recommendations are available to members, or to the general public for a fee. Hard copy, CD-ROM, and downloadable forms of these documents are available at www.itu.int.

The ITU-T organizes recommendations covering various aspects of B-ISDN in a set of alphabetically identified series. The following series are relevant to ATM and MPLS:

- ▼ **Series E** Telephone network and N-ISDN
- **Series F** Non-telephone telecommunication services
- **Series G** Transmission systems and media
- **Series H** Transmission of non-telephone signals
- **Series I** Integrated services digital network
- **Series M** Maintenance
- **Series Q** Switching and signaling
- ▲ **Series Y** MPLS and IP related work

In this book, ITU-T Recommendations are cited as [ITU a.xxxx], where *a* is one of the preceding series letters and *xxxx* is the specific Recommendation number. As can be seen from the References section, the ITU-T has approved a large number of ATM-related recommendations, and a few MPLS ones.

ATM Forum

The ATM Forum started up in January 1992 as a members-only organization with the charter of accelerating the pace of ATM standardization, interoperability, and education.

There are several categories of membership, which have different levels of rights and privileges. The principal focus of this book is the output of the technical committee, which consists of specifications that once finalized are available for free download at www.atmforum.com. The ATM Forum organizes its documents according to technical “subject matter expert” subcommittees, which define the document naming and numbering convention. The groups of related technical subcommittees covered in this book and their commonly used acronyms are:

- ▼ Control Signaling (CS), Routing and Addressing (RA), Private Network-Network Interface (PNNI), and Broadband InterCarrier Interface (B-ICI)
- ATM-IP Collaboration (AIC), formerly called LAN Emulation (LANE), and MultiProtocol Over ATM (MPOA)
- Network Management (NM), Integrated Local Management Interface (ILMI) and testing (TEST)
- Physical layer (PHY) and User-Network Interface (UNI)
- Service Aspects and Applications (SAA) and Security (SEC)
- Traffic Management (TM)
- Voice Telephony over ATM (VTOA)
- ▲ Frame-based ATM (FBATM) and Data Exchange Interface (DXI)

The documents on www.atmforum.com are grouped according to the preceding topics, with each document identified by the subcommittee acronym, a document number, and a revision number. In this book, we refer to ATM Forum documents as [AF xxx], where AF stands for ATM Forum and xxx is either an acronym or a number. These subcommittees meet several times per year, with attendees paying a meeting fee in addition to an annual fee. They also conduct their business over e-mail and a restricted members-only Web page. As seen from the ATM Forum Web site or the Reference section of this book, the ATM Forum has produced many important specifications; and earned a reputation as a fast-paced group covering a broad scope of topics.

Internet Engineering Task Force (IETF)

Commensurate with the mandate for global interoperability of the Internet is the need for standards, and lots of them. The Internet Activities Board (IAB) initially produced standards, but by 1989 the Internet had grown so large that it delegated the work of developing interoperability specifications to an Internet Engineering Task Force (IETF), split into several areas, each with an area director. Each area has several working groups focusing on different issues in the same area. The IETF is not membership based. It is an open forum that accepts only individual contributors, giving no official preference based upon organizational affiliation. Specifications are drafted by working groups in documents called Requests For Comments (RFCs), identified by a serially assigned number. As of

2002, the IETF had produced over 3000 RFCs since 1969. We cover the outputs of many working groups in this book, including the following:

- ▼ Multiprotocol Label Switching (MPLS)
- Provider Provisioned Virtual Private Networks (PPVPN)
- Pseudo-Wire Edge-to-Edge Emulation (PWE3)
- Integrated Services (INTSERV)
- ▲ Differentiated Services (DIFFSERV)

Not all RFCs are standards, and many are obsolete or are only of historical interest. A standards-track RFC passes through a draft stage and a proposed stage prior to becoming an approved standard. Another possible outcome of an RFC is archival as an experimental RFC. The IETF also publishes informational RFCs. As a housekeeping matter, the IETF archives out-of-date RFCs as historical standards. The archiving of all approved (as well as historical or experimental RFCs) serves as a storehouse of protocol and networking knowledge available to the world—accessible, of course, over the Web. The IETF has issued many MPLS- and ATM-related RFCs, as detailed by citations in this book and listed in the References section at the end of the book. Information on the IETF and free RFC downloads are available from the home page at www.ietf.org/home.html.

Frame Relay Forum

Many aspects of the Frame Relay Forum (FRF) are similar to those of the ATM Forum. It is a members-only industry organization, requiring membership fees. Since both organizations deal with connection-oriented protocols based upon ISDN-based signaling using similar network management functions, there is a technical need to define how they can interwork with each other. In fact, the ATM Forum and the Frame Relay Forum corroborated closely in the production of FR/ATM interworking specifications, as described in Chapter 17. There are also some relationships between FR and MPLS that we discuss in this book. You can download FR Forum (FRF) implementation agreements at frforum.com. We refer to these documents using the convention [FRF *xx.y*], where *xx* is the FR Forum document number and *y* is the version.

MPLS Forum

One of the newest players on the standards scene is the MPLS Forum, founded in early 2000. Similar to most other industry forums, it is a members-only organization with several levels of membership. Its charter is to drive worldwide deployment of multivendor MPLS networks, applications, and services through interoperability initiatives, implementation agreements, and education programs. At publication time, the technical committee had two working groups: Interoperability and Applications & Deployment, which had produced only the voice over MPLS document summarized in Chapter 16. For information on

the MPLS Forum or to download approved implementation agreements for free, see www.mplsforum.org.

DSL Forum

In response to the booming demand for broadband access to the Internet of the late 1990s, an important technology that made use of existing twisted wire pairs called digital subscriber line (DSL) was a subject of intense activity. As described in Chapter 11, ATM was one of several methods for carrying information over DSL. The DSL Forum is a members-only organization responsible for creating interoperability specifications for such implementations. You can download these specifications from www.adsl.com for free.

Other B-ISDN/ATM Standards Bodies

There are also national and regional standards bodies involved in the specification of B-ISDN and ATM. The official B-ISDN/ATM standards organization in the United States is the American National Standards Institute (ANSI), and in Europe it is the European Telecommunications Standards Institute (ETSI). These bodies cover aspects not covered in ITU-T or ATM Forum specifications, such as specific aspects of physical interfaces, network management, performance, or specific service aspects. However, over time, many of these aspects are now contained in ITU-T Recommendations. Both ANSI and ETSI are members-only organizations. You can buy ANSI standards at ansi.org. ETSI standards are available for limited free downloads or purchase at etsi.org.

CREATING STANDARDS: THE PLAYERS

Perhaps the single most important driving factor to achieve successful standards and industry specifications is responsiveness to deliver what users actually need (and will pay for), and not what engineers think is technically elegant. Some of the most important questions a user can present to a vendor are “Does it conform to industry standards, which ones, and how?” Standards play a critical role in an age when national and international interoperability is a requirement for successful communications networking. People from many nationalities, diverse cultures, and differing value systems must find a consensus (or a meeting of the minds) in international standards bodies. That these groups of earnest people are able to produce such highly successful, interoperable results testifies to the power of human cooperation.

Vendors

Standards present a dilemma to vendors: on the one hand they must consider the standards, while on the other hand they must develop something proprietary (often leading-edge) to differentiate their products from the rest of the competitive pack. Being the

market leader and being standards-compliant are often contradictory objectives. In the emerging era of MPLS, successful vendors frequently lead standards development and forego proprietary differentiation. This occurs because users are often unwilling to risk future business on a vendor-proprietary system.

Vendors can also drive standards, either by de facto industry standardization, through formal standards bodies, or through industry forums. De facto standardization occurs when a vendor is either an entrepreneur or the dominant supplier in the industry and wants to associate the new technology with their name, such as IBM with SNA or Microsoft with Windows. De facto standards in high-technology areas, however, do not last forever. Sometimes the dominant vendor is not the only one in the market with a product, but the market share, quality, or technology makes them the de facto standard around which other vendors must design.

Users

Generally, users benefit when they purchase equipment conforming to industry standards rather than proprietary solutions, since they can competitively shop for products and services with the assurance of some level of interoperability. Standards, remember, are of paramount importance in the context of international interconnectivity. Also, users play an important role in helping develop standards because the use of standard equipment (as well as vendor acceptance) may well determine the success or failure of the standard. Ubiquitous deployment is often required for success. Vendors say: “We will provide it when customers sign up.” Customers say: “We will sign up when it is universally available at the right price, unless we see something else better and less expensive.” Users usually do not play a very active part in the standardization and specification process. Instead they signal their approval with their purchases—in other words, users vote with their money.

Network Service Providers

Network service providers also actively participate in the standard-making process. In a very real sense, they are major users of vendor equipment. Service providers often select vendors that adhere to industry standards but that still provide some (usually nonstandard) capability for differentiation. This approach does not lock a service provider into one vendor’s proprietary implementation, and it allows for a multiple-vendor environment. Providers must not only make multiple-vendor implementations interoperate within their networks, but they must also interface with other networks. Additionally, they must also ensure the availability of industry standard interfaces to provide value-added services to users. Some service providers utilize single-vendor networks since full implementation of standards is not complete. Furthermore, an example where standards are still incomplete is notably in the management and administrative areas, requiring proprietary solutions to meet essential business needs.

CREATING STANDARDS: THE PROCESS

This section reviews the general standards and specification process illustrated in Figure 3-1. The process begins with the standards organization defining a plan or charter to work on a certain area. Technical meetings and/or e-mail dialog progress the work through a series of written contributions, debates, and drafting sessions. The result is usually a document drafted and updated by the editor in response to contributions and agreements achieved in the meetings. The group reviews the drafts of this document, often progressing through several stages of review and eventually approval—eventually resulting in a final standard or specification. Business and political agendas often influence the standards process: sometimes the industry accepts a de facto standard, while at other times the standards bodies form a compromise and adopt incompatible standards performing the same function. If your company has a vested interest in the outcome of the standard, they should have input and involvement in its creation. Of course, the final measure of success for any standard is the degree of user acceptance and the volume of interoperable implementations produced by the industry, as indicated at the bottom of the figure. We now discuss each of these steps in more detail.

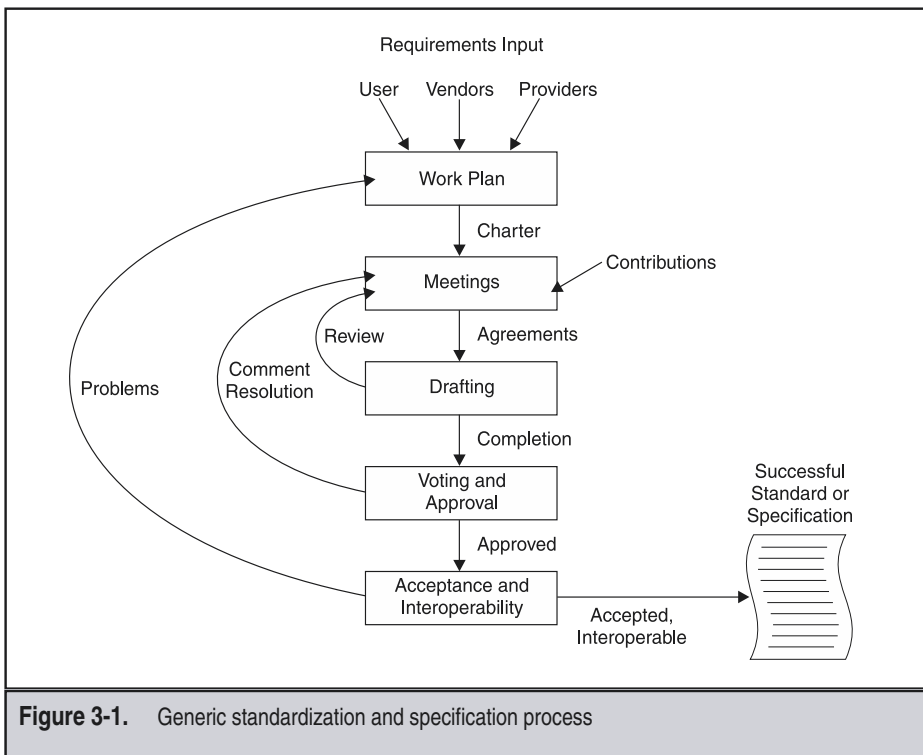


Figure 3-1. Generic standardization and specification process

Charter and Work Plan

Most standards and specifications groups must first agree on a work plan. An exception is a vendor or provider bringing in a fully documented proposal to a standards body. A work plan defines the topics addressed, a charter for the activity, an organization for performing the work, and usually a high-level set of objectives. User input and involvement most likely occur at this stage, either indirectly or sometimes through direct participation. This is the time that users and service providers often voice their high-level requirements to vendors. The work plan for updating an existing standard usually includes some changes resulting from user feedback or interoperability issues that have arisen. Often, the work plan sets an approximate schedule for completion of the standard or specification.

Meetings and Contributions

The majority of the work occurs at technical meetings, which last from several days to a week or more or via e-mail interaction. Typically, participants submit contributions in advance of meeting face to face. Contributions take on many flavors, ranging from proposing specific text and drawings for the draft standard, presenting background information, articulating arguments for or against a particular approach, or serving as liaison with other standards or specification bodies. Usually, smaller subcommittees discuss these detailed contributions, except when a contribution proposes something of interest to the entire standards body. If the contribution proposes adding or changing text to a baseline document, then the subcommittee employs a process to determine whether the proposal is accepted, amended, or rejected. Formal standards bodies usually attempt to achieve consensus before agreeing to include a contribution's input, while other industry forums employ a straw vote method to accept or reject proposals.

The ITU-T, the ATM Forum, and the IETF are all large committees, with hundreds to thousands of members attending each meeting. These large committee meetings normally have a plenary session where representatives from all the subcommittees attend. Outside the plenary meeting, multiple subcommittee meetings usually occur in parallel. The subcommittees are granted some autonomy. However, they usually review major changes or key decisions in the plenary session or through some other formalized process. Meetings also are used to resolve issues that arise from the drafting, review, voting, or approval process described in the next couple of sections.

Drafting and Review

Key individuals in the development of a standard or specification are the editors. Standards would never exist without the efforts of these dedicated individuals. Many standards documents explicitly give credit to the editor and acknowledge key individuals who contributed to the overall standard or specification. This inspires extra effort and participation. As indicated in the Reference section of each chapter, the IETF explicitly acknowledges these individuals as authors of an RFC. Many other standards documents also acknowledge contributors.

The editors draft text based upon the contributions, as amended in a meeting or in response to e-mail comments. The working group usually trusts the editor to research related standards and specifications and align the document accordingly. A key part of any standards or specification technical activity is the ongoing review, correction, and improvement of the draft document, which provides the basis for contributions for the next meeting.

Approval and Consensus

Once a particular document has reached a “final draft” status, the committee usually distributes it for approval via consensus or voting. Comments that members believe must be addressed in order to approve the document as a standard or specification are often addressed via a comment resolution process at meetings or over an e-mail list, resulting in more drafting for the editor and the subcommittee. The voting step of the process differs in various bodies in the number of members required to approve a change. If complete concurrence is the objective, then the process can be lengthy; if only a majority vote is required, then progress may be more rapid, but that can possibly increase the risk. Since human beings work on the standards, the occasional instance of human error is inevitable. After completing the comment resolution process, the standard or specification then goes to final approval. Again, depending upon the rules governing the standards or specification body, anything ranging from a simple majority to unanimous approval is necessary for the body to release the document as an approved standard or specification. Often a supervisory board reviews the proposed standard for consistency with the format, style, scope, and quality required by that body in the final approval stage.

User Acceptance and Interoperability

Some users have business problems today that can be solved only by proprietary implementations *prior* to the development of standards. Waiting for an approved standard could put these users out of business. Therefore, the user is caught in the dilemma of adopting an emerging standard now, or else waiting for it to mature. Users primarily determine the success of standards by creating the demand for specific capabilities, and even technology, by purchasing various implementations from vendors and service providers supporting that standard.

The key technical measure of a standard’s or specification’s success is whether implementations from multiple vendors or carriers interoperate according to the details of the documentation. Remember too that specifications also point out where systems will *not* interwork. The documents should specify a minimum subset of interfaces, function, and protocol to achieve this goal. In support of interoperability, additional documentation, testing, and industry interoperability forums may be required.

If users do not accept a standard, or if significant interoperability issues arise, then this feedback is provided back into the standards process for future consideration. Acceptance by the vendor community also plays a key role in the success or failure of standards—if no implementation of the standard is built, no user or provider can buy it!

OTHER ASPECTS OF STANDARDS

There are several other important aspects of standards that are not often written about. This section summarizes these considerations.

Business and Politics

Standards organizations and industry forums have had increased participation and scope in recent years. With this increased number of people working on a plethora of problems there comes the inevitable burden of bureaucracy. Service providers, vendors, and to some extent, users view the chance to participate in the standard-setting process as an opportunity to express and impress their views upon the industry. This is a double-edged sword: while participation is necessary, biases are brought to the committees that can tie up decision making and bog down the process for years in making standards. The impact of this type of situation depends on whether the committee operates on a complete consensus basis or some form of majority rule. A consensus-based approach can end up with multiple, incompatible options stated in the standard. There is then a need to further subset a standard as an interoperability specification to reduce the number of choices, and to translate the ambiguities of the standard into specific equipment requirements.

Standards can also have omissions or “holes” left to vendor interpretation because they simply weren’t conceived originally as issues. These “holes” may exist because no agreement could be reached on how the requirement should be standardized, or merely because of oversight. Typically, standards identify known “holes” as items “for further study” (sometimes denoted as “ffs”) just to point out that there is an awareness of a need for a function or element that isn’t yet standardized.

Some vendors play a game of supporting their proprietary solution to make it a standard before their competitor’s proprietary solution becomes a standard. Increasingly, you will see intellectual property rights cited in draft or approved standards for this reason. In some cases, this makes sense because a proprietary solution has a number of attractive attributes. However, trying to rework a solution to avoid intellectual property considerations can draw out the standards process for many months or even years. While standards organizations take their time to publish standards, some vendors try to take the lead and build equipment designed around a draft standard and then promise compliance with the standard once it is finally published. This is savvy marketing if a vendor guesses right, but if they miss the mark, a significant investment could be lost in retooling equipment and/or rewriting software to meet the final standard.

Measures of Success and Proven Approaches

Standards-conscious players should always keep the mission in mind, regardless of the dangers present. Those bodies that keep the user’s application foremost in mind—while dealing with the problems and dangers of the standards process—will live long and prosper, or at least live longer than those who concentrate primarily on the development of standards. User trial communities and university test beds are another method employed

by these forums to help speed up the testing and acceptance of new technologies. The Internet protocols are an example of the work by university and academia to successfully lead standards development. Indeed, the notion of “deployment propagation” created by the IETF is that a proposed standard must first demonstrate interoperability before final ratification begins catching on in other standards bodies. Standards development seems to have been improved by the adoption of this free market approach, just as the same free market of ideas has stimulated the world.

Predicting the Future of Standardization

Sometimes standardization moves at a very slow pace, while at other times the pace is quite rapid. The flurry of ATM standards has slowed since the heyday of the mid-1990s, and now as described in this book, much vendor and service provider standardization focus has now shifted to MPLS in several standards bodies. As we shall see, MPLS has some unique characteristics given its strong IP heritage, but in other respects it is following a similar path to that of ATM.

Already, there are cases of competition between standards bodies in MPLS in a manner similar to that which occurred between the ATM Forum and the ITU-T in the early 1990s. Hopefully, these groups will cooperate to produce standards that are coordinated with minimal overlap. This is important because multiple standards for essentially the same function create little benefit for the industry. Vendors incur additional costs to implement multiple standards in order to sell to as many service providers as possible. Service providers with multiple standards in place as a result of mergers and acquisitions must implement and test interworking, or migrate to one standard. In turn, service providers must pass these additional costs on to users. The principal winners in developing multiple standards for comparable function appears to be only the standards bodies themselves.

REVIEW

This chapter identified the organizations taking an active role in standardizing and specifying ATM and MPLS equipment and services. The ITU-T and ATM Forum are the principal standards bodies for ATM, while the IETF is the principal standards body for MPLS. There are so many standards that we have relegated the listings to the References section at the end of this book, with specification citations called out in the text that covers a particular topic. We then covered the role of the various players in the standards process: users, vendors, and service providers. The chapter then described the standards development process using a flowchart with narrative for each major step along the way. The chapter concluded with a discussion of other aspects of standards, such as business and politics, as well as competition between standards bodies.

PART

II



Networking and Protocol Fundamentals

This part describes the topologies, technologies, protocols, and services used in modern digital communications networks. The following chapters cover these subjects to give the reader a background on the successful concepts from prior protocols proven in real-world digital communication networks that serve as the foundation for understanding both ATM and MPLS. Chapter 4 provides a comprehensive introduction to these topics by reviewing common network topologies and circuit types, including an in-depth explanation of the principles of multiplexing and switching. Chapter 5 introduces the concepts of layered protocols and the architectures that influenced the history of networking: the Internet Protocol (IP) suite, IBM's Systems

Network Architecture (SNA), the OSI Reference Model (OSIRM), and the IEEE's 802.x LAN protocols. Chapter 5 also introduces the key concepts of connection-oriented and connectionless network services. Chapter 6 then defines the basics of the digital transmission hierarchy as the foundation for the Integrated Services Digital Network (ISDN) reference model. Chapter 7 reviews the major connection-oriented services: X.25 packet switching and Frame Relay. Chapter 8 then surveys key characteristics of the most commonly used connectionless service, namely, the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. Finally, Chapter 9 provides a comprehensive summary of local area networks, bridging, and routing.

CHAPTER 4



Networks, Circuits, Multiplexing, and Switching

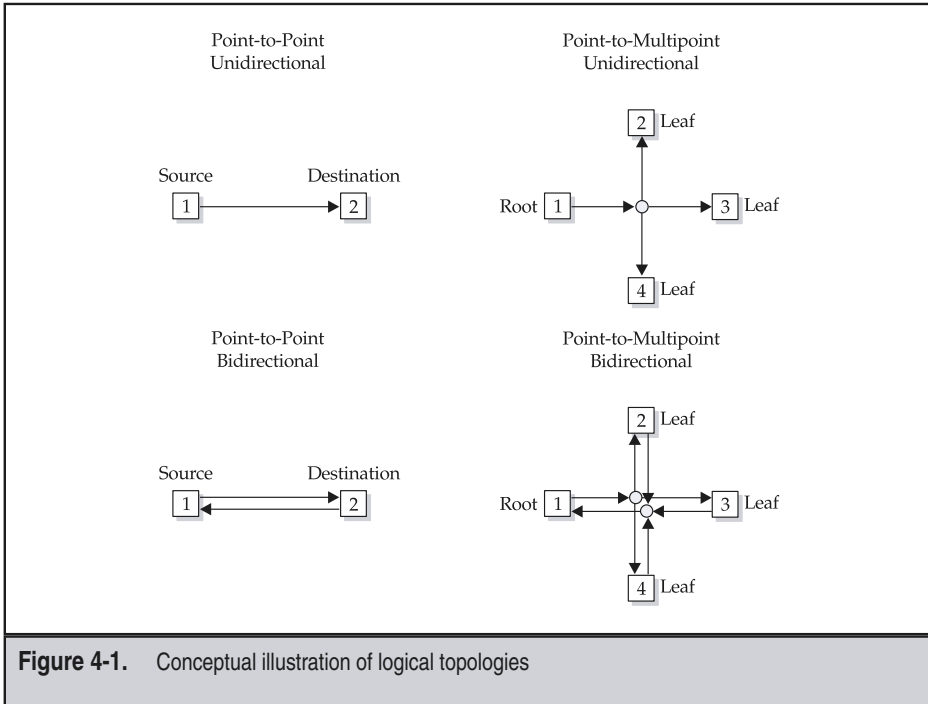
This chapter provides an overview of common network topologies, circuits, and transmission types that form the basis for most network designs. We define the key characteristics of the links or circuits that connect nodes in these topologies, such as the direction of data flow, bit- or byte-oriented transmission, and the broadcast nature of the media. Next, the text reviews the several meanings of synchronous and asynchronous transmission. Coverage then moves to an explanation of multiplexing techniques—methods of combining and separating units of capacity. Finally, the discussion covers the five major methods of switching data: in space, in time, in frequency, by address, or by codes. We point out that address (or label) switching is the foundation of packet switching, which includes ATM and MPLS. The reader will then have background to understand the concepts underlying common networking protocols, such as Frame Relay, IP, ATM, and MPLS, that are rooted in these multiplexing and switching techniques.

GENERAL NETWORK TOPOLOGIES

Physical topology defines the interconnection of physical *nodes* by physical transmission *links*. A *node* is a network element, such as an ATM switch, an IP router, an MPLS label switching router, or a multiplexer. We also refer to a node as a *device*, and to a link as a *transmission path*. A *link* represents a connection between two nodes, either physical or logical. Therefore, a link may be either a physical connection, such as a dedicated private line, or a logical, or virtual, connection, such as a permanent virtual connection (or circuit) (PVC).

Logical topology defines connections between two or more logical nodes (or simply interfaces), which may be of either a *point-to-point* or a *point-to-multipoint* configuration in ATM. Furthermore, each connection may be either *unidirectional* or *bidirectional*. A *leaf* is the terminating point of a unidirectional point-to-multipoint topology with originations at the *root*. A spatial point-to-multipoint connection has at most one leaf per physical port, while a logical point-to-multipoint connection may have multiple leaves on a single physical port. When all nodes have a point-to-multipoint connection, then a broadcast logical topology results. Figure 4-1 illustrates each of these logical topologies. Other technologies, such as Ethernet, support a broadcast medium where *all* other stations receive any one station's transmission. Additional protocols and configurations are required to support the broadcast logical topology. Part 4 describes how ATM supports broadcast via LAN Emulation (LANE) and IP Multicast over ATM using sets of point-to-multipoint connections. For point-to-point connections, ATM uses the bidirectional topology while MPLS uses the unidirectional topology.

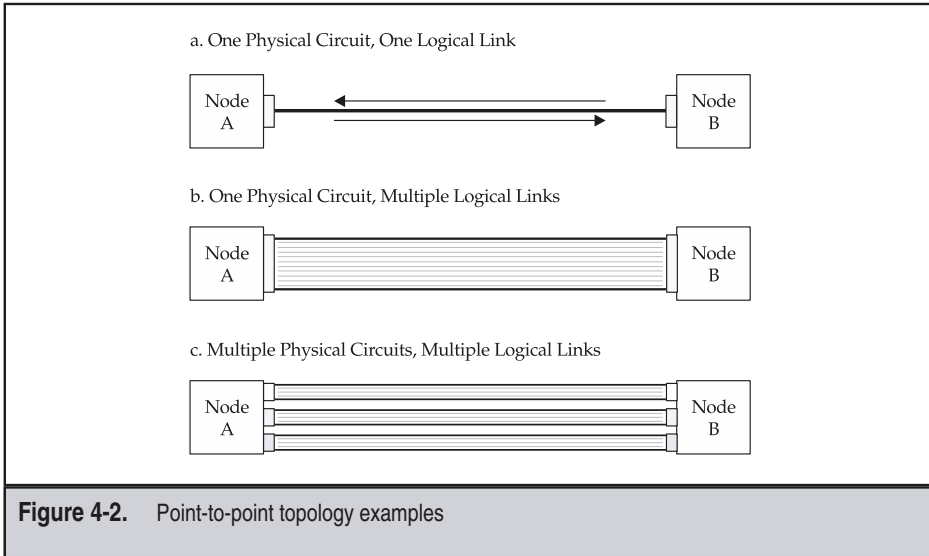
The most commonly used physical topologies for computer and data communications networks are point-to-point, multipoint (or common bus), star, ring (or loop), and mesh. The following sections provide illustrated examples of each network topology.



Point-to-Point

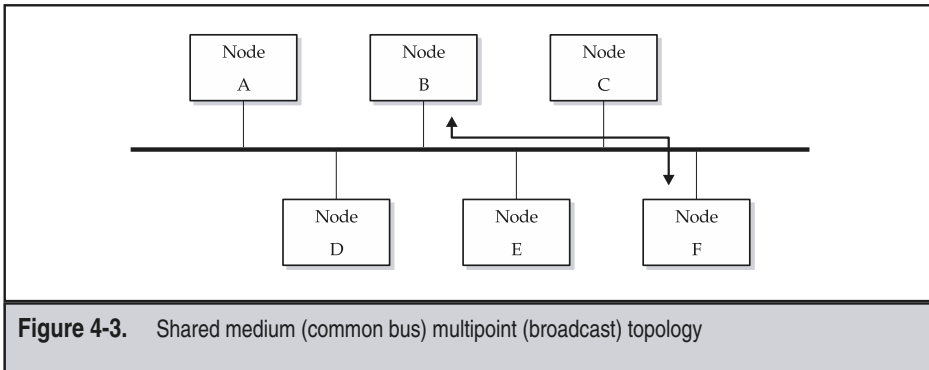
The *point-to-point* topology is the simplest, consisting of a single connection between two nodes composed of one or more physical or logical circuits. Figure 4-2 shows three examples of a point-to-point topology. The first example shows a single physical circuit connecting Node A and Node B. The second example depicts a single physical circuit between Node A and Node B carrying multiple logical links. The third example depicts a single connection path between Node A and Node B with multiple physical circuits, each carrying multiple logical links. Typically, network designers employ this configuration when the separate physical circuits traverse diverse routes, in which case any single physical link or circuit failure would not completely disconnect nodes A and B.

Point-to-point topologies are the most common method of connectivity in metropolitan area networks and wide area networks (MANs and WANs). User access to most MAN or WAN network services has some form of point-to-point topology. Examples of the point-to-point topology are private lines, circuit switching, and dedicated or dial-up access lines to packet-switched services, Frame Relay, ATM, and MPLS.



Multipoint and Broadcast

A common realization of the *multipoint* topology is a network where all nodes physically connect to (and logically share) a common broadcast medium. Figure 4-3 shows the multipoint topology, where Nodes A through F communicate via a shared physical medium. Sometimes the shared medium is also called a common bus. Most local area networks (LANs) utilize a broadcast (or multipoint) topology. Indeed, the IEEE 802.4 Token Bus, the IEEE 802.3 Ethernet, and the IEEE 802.6 Distributed Queue Dual Bus (DQDB) protocols define different means of logically sharing access to the common physical me-

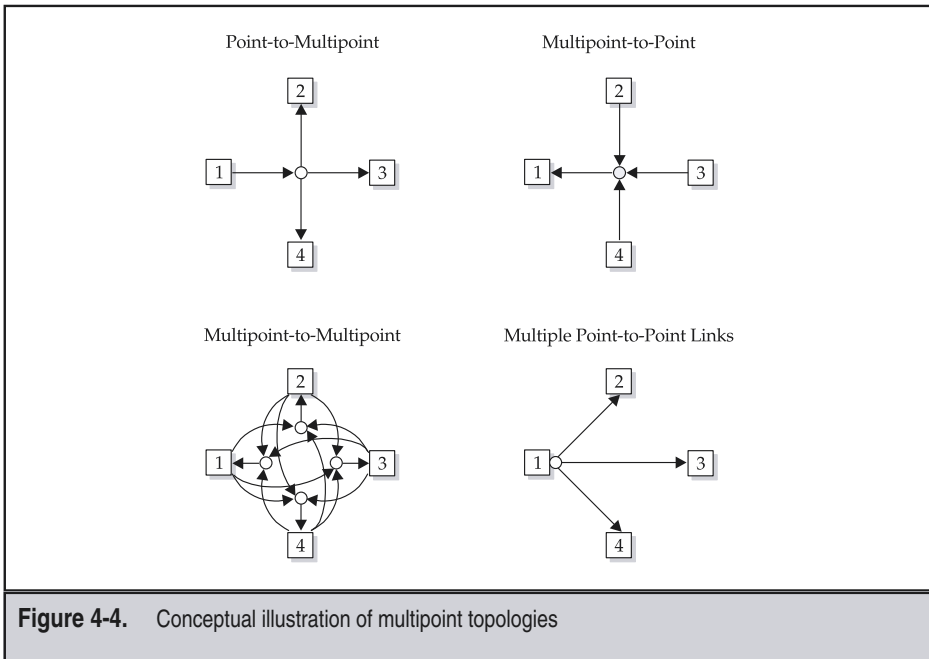


dium topology, as do other proprietary vendor architectures. Radio and satellite networks also implicitly employ the broadcast topology due to the inherent nature of electromagnetic signal propagation.

A multidrop analog line is commonly used for legacy SNA Synchronous Data Link Control (SDLC) loop access, further described in Chapter 7. In this example, an analog signal is broadcast from a master station (usually a mainframe front end processor) to all slave stations. In the return direction, the slaves share the common broadcast medium of the multidrop line. The SNA SDLC polling protocol involves the host polling the slave stations in a round-robin manner, thus preventing any two slaves from transmitting at the same time. (See References [Cypser 78] and [Ranade 89] for more information on the SNA SDLC protocol.)

Other networks, notably the Ethernet protocol, also work on a broadcast medium but don't provide for orderly coordination of transmissions as the SNA SDLC loop does. Instead, these protocols empower stations to transmit whenever they need to as long as another station isn't already sending data. When a collision does occur, a distributed algorithm uses the bandwidth at approximately 50 percent efficiency. Chapter 9 covers Ethernet and related local area networking protocols.

Figure 4-4 illustrates other conceptual examples of the multipoint topology. Another commonly used multipoint topology is that of broadcast, or multipoint-to-multipoint, which is the case where many other nodes receive one sender's data. Yet another example



is that of “incast,” or multipoint-to-point, where multiple senders’ signals are received at one destination—as in a slave-to-master direction. Some of the foundational work in MPLS embraced the multipoint-to-point concept with the aim of reducing the number of connection points within a network. In this conceptual illustration, note that the multipoint-to-multipoint (i.e., shared medium, or multicast) topology is effectively a full mesh of multipoint-to-point connections between each of the four nodes. The figure also illustrates emulation of a point-to-multipoint topology via multiple point-to-point links for comparison purposes.

Star

The *star* topology developed during the era when mainframes centrally controlled most computer communications. The voice-switched world also employs a star topology when multiple remote switching nodes, each serving hundreds to even thousands of telephone subscribers, home in on a large central switch. A variation on this topology is a dual star, which achieves enhanced resilience. This type of network radiates in a star-like fashion from the central switch through the remote switches to user devices. The central node performs the communication switching and multiplexing functions in the star topology. Nodes communicate with each other through point-to-point or multipoint links radiating from the central node. The difference between this topology and the multipoint topology is that the central node provides only point-to-point connections between any edge node, on either a physical or logically switched basis.

Figure 4-5 shows a star topology, where Node A serves as the center of the star and Nodes B through E communicate via connections switched to and through the central Node A. An example of a star topology is many remote terminal locations, or clients, accessing a

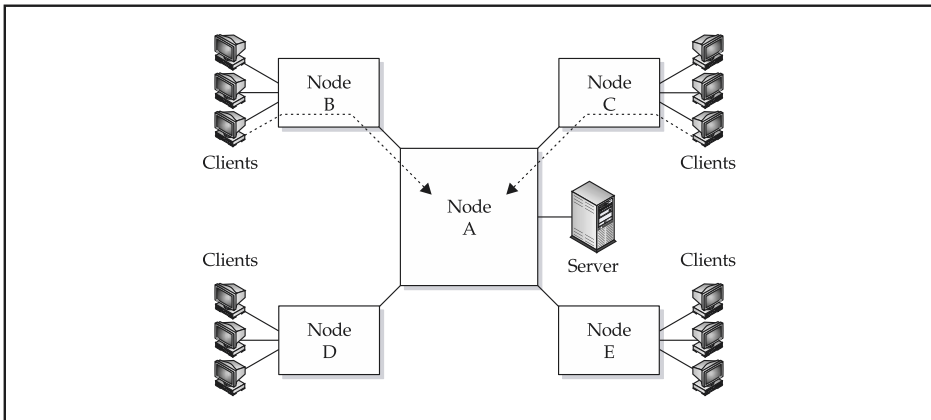


Figure 4-5. Illustration of a network with a star topology

centralized server through the central node as illustrated in the figure. The physical star topology is widely used to connect devices to a central hub in LANs, and thus it is often called a “hub and spoke” topology. The central hub may logically organize the physical star as a logical bus or ring, as is commonly done in LAN wiring hubs. A key benefit of the physical star topology is superior network management of the physical interfaces. For example, if a single interface fails in a physical star topology, then the management system can readily disable it without affecting any other stations. Conversely, in a broadcast topology, a single defective switch can take down the entire shared-medium network. As we shall see, many wide area networks also have a star topology, driven by the client/server computing paradigms.

Ring

The *ring* topology utilizes a shared transmission medium that forms a closed loop. Such networks utilize protocols to share the medium and prevent information from circulating around the closed physical transmission circuit indefinitely. A ring is established, and each device passes information in one direction around the ring.

Figure 4-6 shows a ring network where in step 1, Node A passes information addressed around the ring through Node D in step 2. Node C removes this frame from the ring and then returns a confirmation addressed to Node A in step 3 via Node B, at which point Node A removes this data from the ring in step 4. Note that actual implementations of ring structures for LAN protocols use a more complicated protocol than that described here. Rings reuse capacity in this example because the destination removes the information from the ring so that other stations can utilize the ring bandwidth. Examples of the ring topology protocols are the IEEE 802.5 Token Ring and the Fiber Distributed Data Interface

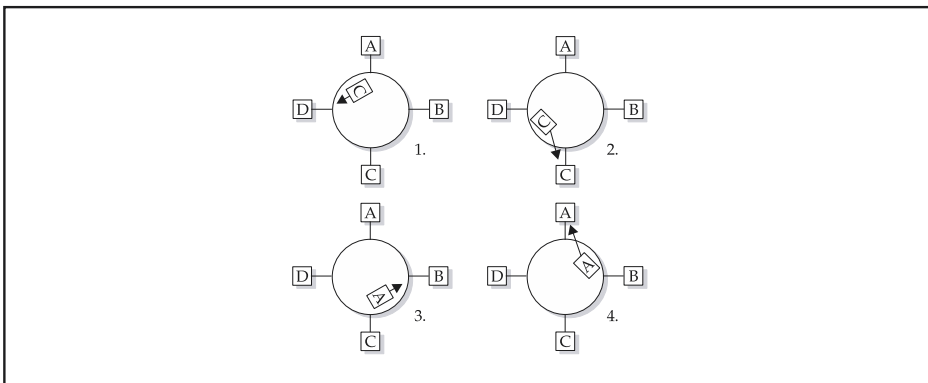


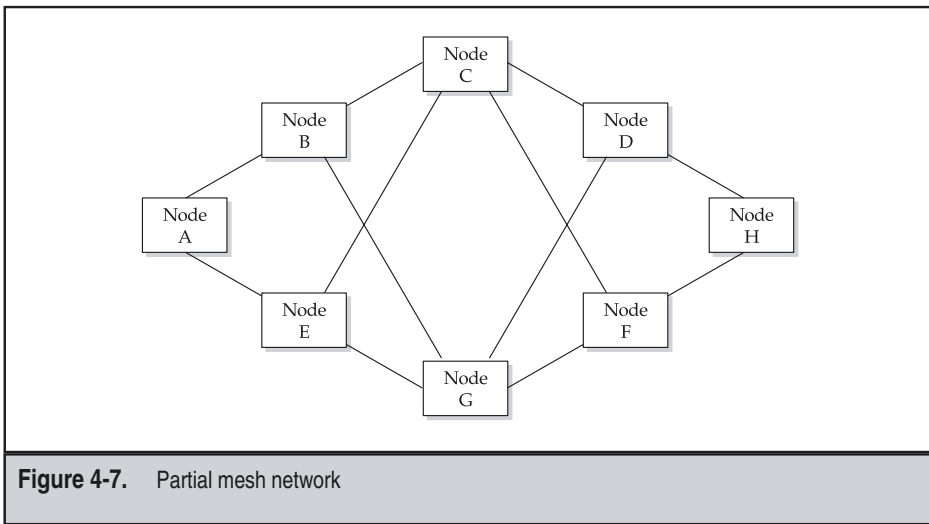
Figure 4-6. Ring or loop topology

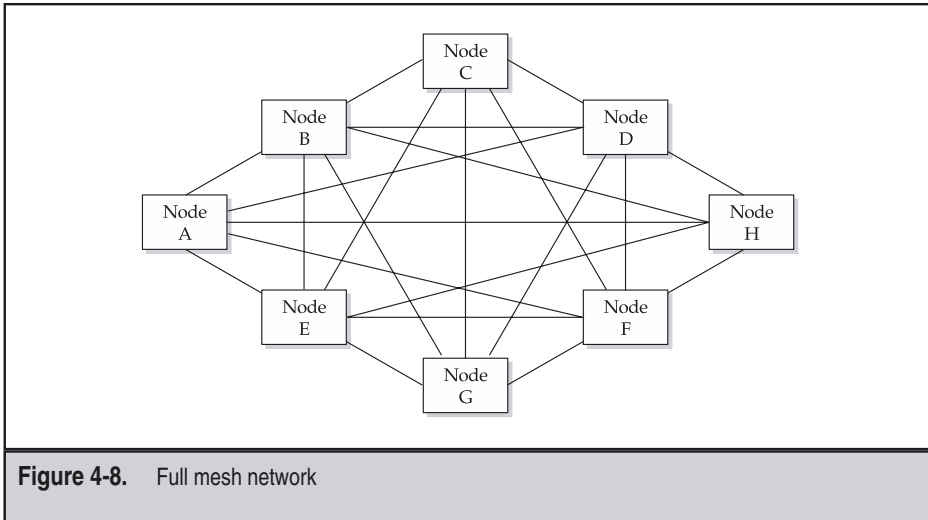
(FDDI). Although the ring topology looks like a special case of a mesh network, it differs because of the switching action performed at each node. SONET protection rings also use the ring topology, and they are also distinguished from a mesh by the difference in nodal switching action from that of a mesh of circuit switches.

Mesh

Many switched, bridged, and routed networks employ some form of mesh architecture. Mesh networks have many nodes, which are connected by multiple links. If each node is directly connected to every other node, then the network is fully meshed; otherwise, the network is only partially meshed. Figure 4-7 shows a partial mesh network where Nodes B, C, D, E, F, and G have a high degree of connectivity by virtue of having at least three links to any other node, while Nodes A and H have only two links to other nodes. Note that Nodes C and G have four links. The number of links connected to a node is that node's degree (of connectivity). For example, Node C has degree 4, while node H has degree 2.

Figure 4-8 shows a *full mesh* network where each node has a link to every other node. Almost every major computer and data communications network uses a partial mesh topology to give alternate routes for backup and traffic loads. Few use a full mesh topology, primarily because of cost and/or complexity factors associated with having a large number of physical and/or logical links. This is because a full mesh N -node network has $N(N-1)/2$ links, which is on the order of N^2 for large values of N . For N greater than 4 to 8 nodes, most real-world networks employ partial mesh connectivity.



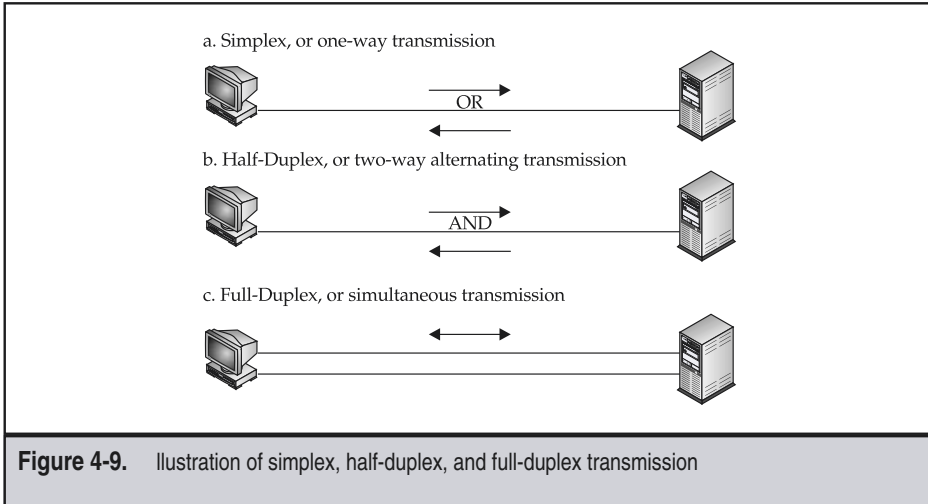


DATA COMMUNICATIONS AND PRIVATE LINES

This section takes a detailed look at the characteristics of connections used in real networks. First, we introduce the notion of simplex, duplex, and half-duplex communications. The treatment then introduces the concepts of data terminal and communications equipment. We then put these concepts together and apply them to private lines, which form the fundamental component of connectivity for most data communications, multiplexing, and switching architectures. Carriers offer private lines as tariffed services, or as access lines to other data communications services, such as Frame Relay, ISDN, ATM, the Internet.

Simplex, Half-Duplex, and Full-Duplex Transmission

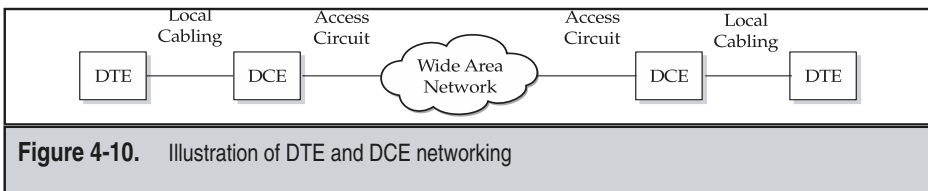
Figure 4-9 illustrates the three types of transmission possible in data communications: simplex, half-duplex, or full-duplex [Held 95]. In the *simplex* transmission example in Figure 4-9a, one physical communications channel connects the terminal to the computer, in which case transmission occurs in only one direction. Examples of simplex communication are a radio broadcast or an alarm system. In the *half-duplex* transmission example in Figure 4-9b, only one physical communications channel connects the terminal and computer; however, a protocol allows communication to occur in both directions, but not simultaneously. Half-duplex communication is also called two-way alternating communication. An example of half-duplex operation is Citizen Band (CB) radio transmission where a user can either transmit or receive, but not do both at the same time on the same channel. Finally, in the *full-duplex* example in Figure 4-9c, two



transmission channels connect the terminal and the computer. In full-duplex operation, communication can take place in both directions simultaneously.

DTE-to-DCE Connections

DTE-to-DCE connections provide a local, limited-distance physical connection between data terminal equipment (DTE) or terminal equipment (TE), such as a computer or PC, and data communications equipment (DCE), such as a modem, designed to connect to a wide area network, as illustrated in Figure 4-10. A DCE is equipment that provides functions required to establish, maintain, and terminate a connection between a DTE and a wide area network. Typically, the local cabling between the DTE and the DCE is a multistrand cable, but it may consist of coaxial, fiber optic, or twisted pair media. The DCE connects to an access circuit via twisted pair, coaxial cable, or fiber optic media. DTE-to-DCE and DCE-to-network communication is a particular example of a point-to-point topology. DTE-to-DCE communication can operate in any one of the transmission modes defined in the previous section: simplex, half-duplex, or full-duplex.



The DCE-to-DTE connections are the main reason we need so many different types of cables. It is an RS-232 concept that unfortunately was inherited by the Ethernet cabling standards. The original idea behind DCE and DTE was that there were two types of equipment—"terminal"-type of equipment that generates and/or receives data of its own, and "communication"-type equipment that relays only data generated by someone else. It was decided that the 25-pin RS-232 connectors on these two types of equipment actually needed to be wired differently, with the result being that you then needed two different types of cables: one for connecting a DTE to a DCE and another for connecting two DTEs directly to each other.

A DTE-to-DCE connection in a multistrand cable electrically conveys multiple interchange circuits of one of two types: balanced or unbalanced. What's the difference? One wire. Specifically, a balanced interchange circuit uses a pair of wires, while an unbalanced interchange circuit uses a single wire. The practical difference between these techniques is that unbalanced circuits require fewer wires in a cable but are more susceptible to electrical interference. Hence, unbalanced circuits operate over shorter distances than balanced ones, which require twice as many wires in the cable.

A multistrand cable connecting a DTE to a DCE carries multiple interchange circuits, some of which carry data, while others that carry control information. The example of Figure 4-11 show the commonly used control signals in the RS-232 standard. The widely used RS-232 standard commonly used for serial port connections in personal computer applications employs unbalanced (one-wire) interchange circuits with a common ground as shown in the figure. Each of these circuits appears as a signal associated with a physical pin on the serial port connector. The control circuits (shown by dashed lines in the figure) allow the DTE and DCE to communicate status and request various services from each other. For example, the DCE uses signaling on the control leads between the DTE and DCE at the physical layer to control the direction of transmission in half-duplex communication. When a single pair of wires, a coaxial cable, or a fiber optic cable connects a DTE and DCE, special modulation techniques provide for a functional separation of data and control signals to yield the equivalent of separate interchange circuits for control and data.

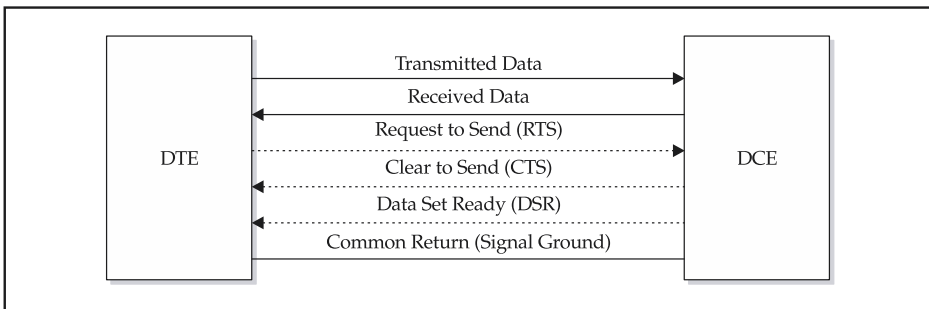


Figure 4-11. Example of a DTE-to-DCE connection using the RS-232 interface

Private Lines

A private line, or leased line, is a dedicated physical circuit leased from a carrier for a pre-determined period of time, usually in increments of months. A private line may be upgraded by paying extra for a defined quality of service, such that special conditioning is performed to ensure that a better error rate is achieved—resulting in a tremendous improvement in data communications performance. As carriers install all-fiber networks, digital private lines are replacing the old voice-grade analog circuits, at lower prices.

When users purchase leased lines to gain access to other services, such as Frame Relay, the Internet, or ATM, we call this application an *access line*. Users may lease access lines either through the local telephone company or, in an increasing number of locations, through alternative access providers. In some cases, end users own their access facilities; for example, through construction of a fiber optic ring. Generally, access from these alternative sources is less expensive than the local telephone company prices.

Private lines in Europe and Pacific Rim countries were very expensive, but installation of more transoceanic fiber routes and the advent of competition has driven prices down. Although prices are dropping, a carrier must make a significant investment to achieve the benefit. Historically, the high cost of international private lines justified the expense of sophisticated statistical multiplexers to utilize the expensive bandwidth as efficiently as possible. ATM, IP, and MPLS offer a way to achieve efficient use of expensive transmission facilities.

Another form of special-purpose private line operating over a four-wire circuit is the high-rate digital subscriber line (HDSL). HDSLs eliminate the cost of repeaters every 2000 ft for the first repeater and 6000 ft for subsequent ones in a standard T1 repeater system, and they are not normally affected by bridge taps (i.e., splices). They need to be within 12,000 ft of the serving central office, which covers over 80 percent of the DS1 customers in the United States. Digital subscriber lines (DSLs) are also becoming available that offer higher speeds and better performance. The goal of the DSL technology is to deliver Internet access, and potentially telephone service over a majority of the existing copper, twisted pairs currently connected to small and medium businesses as well as residences. Part 3 covers the topic of DSL as it applies as a physical medium for ATM.

DATA TRANSMISSION METHODS

A digital data transmission method is often characterized as being either asynchronous or synchronous. The terms *asynchronous* and *synchronous* are used in different contexts where, unfortunately, they have entirely different meanings. We first describe definitions different than the ones used in this book, commonly known as asynchronous and synchronous character message transmission [Held 95]. The meaning used in this book is that of Synchronous versus Asynchronous Transfer Mode (STM and ATM). These two entirely different meanings of the same two terms can be confusing. This section presents them together so that you can appreciate the differences and understand the context of each. The traditional meaning of the terms asynchronous and synchronous apply at the

character or message level. On the other hand, the same adjectives used in the STM and ATM acronyms define different transmission system paradigms for carrying characters and messages.

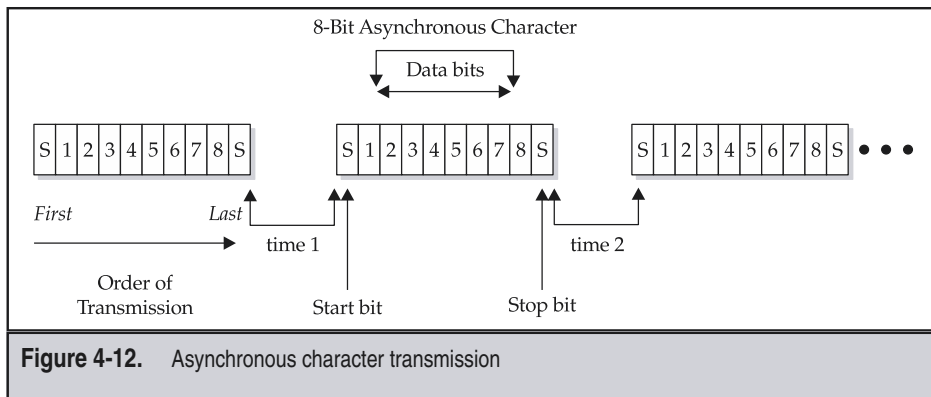
Asynchronous and Synchronous Data Transmission

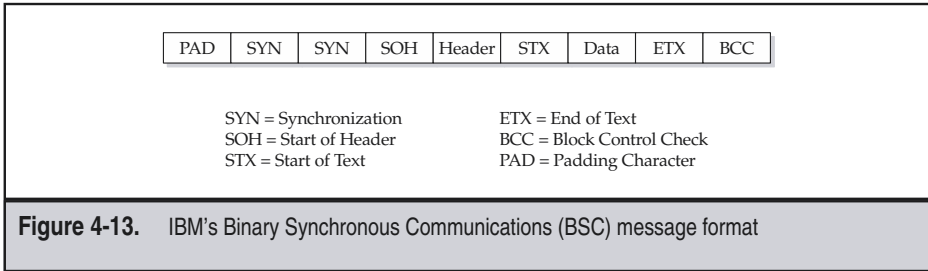
Asynchronous character transmission has no clock either in or associated with the transmitted digital data stream. Instead of a clock signal, start and stop bits delimit characters transmitted as a series of bits numbered 1 through 8, as illustrated in Figure 4-12. There may be a variable amount of time between characters. Analog modem communication employs this method extensively. Chapter 23 explains the notion of baud rate, which is the number of discrete signals transmitted on the communication channel per unit time.

Asynchronous character transmission usually operates at lower speeds ranging from 300 bps up to 28 Kbps. Asynchronous interfaces include RS232-C and D, as well as X.21.

On the other hand, synchronous data transmission clocks the bits at a regular rate set by a clocking signal either associated with or derived from the transmitted digital data stream. The motivation for synchronous signaling is to eliminate the extra time required to send the start and stop bits in asynchronous transmission, thereby increasing efficiency. Since the start and stop bits are at least one unit of time long, they comprise at least 20 percent of the line transmission rate. Therefore, in synchronous transmission, the sender and receiver must have a means to derive a clock within a certain frequency tolerance.

Figure 4-13 shows a typical synchronous data stream from the Binary Synchronous Communications (BSC) protocol employed by IBM in the 1960s for the System 360 [Cypser 78]. The message begins with a PAD character, followed by two synchronization (SYN) characters and a Start-of-Header (SOH) character. The header supports functions such as addressing and device control. A Start of Text (STX) character then precedes the textual (data). Textual data cannot contain any control characters in BSC, unless another character—Data Link Escape (DLE)—is sent prior to the STX character. If the data





contains a DLE character, then the transmitter inserts an additional DLE character. The receiver strips one DLE character for each pair of DLE characters received. This DLE stuffing operation allows users to send transparent data containing other control characters. The End of Text (ETX) character delimits the message. The Block Control Check performs a simple parity check to detect errors in the data characters. For various reasons, other data link control methods eventually replaced the BSC protocol.

On a parallel DTE-to-DCE interface, a separate clock interchange circuit (or connector pin) conveys the synchronous timing. Synchronous data interfaces include V.35, RS449/RS442 balanced, RS232-C, RS232-D, HSSI, and X.21. Synchronous data transmission usually runs at higher speeds than asynchronous. For example, the High Speed Serial Interface (HSSI) operates at speeds up to 51.84 Mbps.

Asynchronous Versus Synchronous Transfer Modes

This section introduces the meanings of the adjectives *synchronous* and *asynchronous* as applied to transfer modes used in this book. Fundamentally, a *transfer mode* defines the means for conveying sequences of bits between multiple sources and destinations over a single digital transmission system. In effect, a transfer mode is a means for multiplexing multiple data streams onto a single higher-speed bit stream, and then sorting it all out at the destination. Chapter 6 describes Synchronous Transfer Mode (STM), or synchronous time division multiplexing, in detail. Asynchronous Transfer Mode (ATM), or asynchronous time division multiplexing, is a different concept with roots in packet switching (covered in detail in Part III). The following example provides a high-level introduction to the basic difference between the STM and ATM multiplexing methods.

Figure 4-14 shows examples of STM and ATM. Figure 4-14a illustrates an STM stream where each time slot represents a reserved piece of bandwidth dedicated to a single channel, such as a DS0 in a DS1. Each frame contains n dedicated time slots per frame; for example, n is 24 8-bit time slots in a DS1. Overhead fields identify STM frames that often contain operations information as well. For example, the 193rd bit in a DS1 delimits the STM frame. Thus, if a channel is not transmitting data, the bits in the time slot remain reserved without conveying any useful information. If other channels have data to transmit,

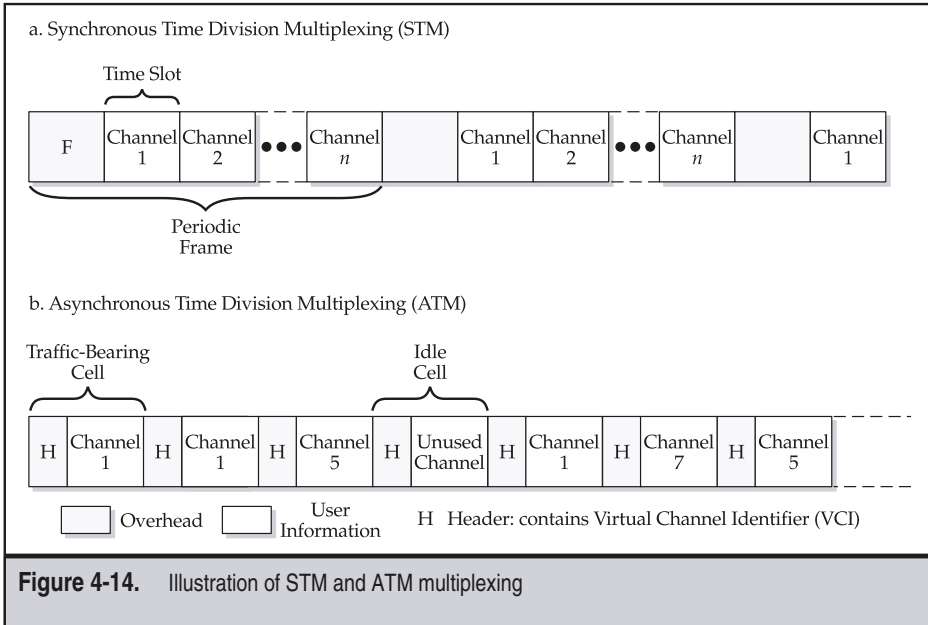


Figure 4-14. Illustration of STM and ATM multiplexing

they must wait until their reserved, assigned time slot occurs in turn again. If time slots are frequently empty, then STM results in low utilization.

ATM uses a completely different approach. Figure 4-14b illustrates the concept. A header field prefixes each fixed-length payload channel, identifying the virtual channel. The combination of the header and payload is a *cell*. The time slots (or cells) are available to *any user* who has data ready to transmit. The traffic need not wait for its next reserved time slot as in STM. If no users are ready to transmit, then ATM sends an empty, or idle, cell. Traffic patterns that are not continuous are usually carried much more efficiently by ATM as compared with STM. The current approach is to carry ATM cells over very-high-speed STM transmission networks, such as SONET and SDH. As we shall see, the match between the high transmission speeds of SONET and SDH and the flexibility of ATM is a good one. In general, packet switching uses a similar concept, except that the “slots” are of variable length.

PRINCIPLES OF MULTIPLEXING AND SWITCHING

A close family relationship binds the concepts of multiplexing and switching. *Multiplexing* defines the means by which multiple streams of information share a common physical transmission medium. *Switching*, on the other hand, takes information from an

input multiplexed information stream and directs this information to other outputs. In other words, a switch takes information from a particular physical link in a specific multiplexing position and connects it to another output physical link, usually in a different multiplexing position. Multiplexing positions are defined by space, time, frequency, address, or code. Since a switch is in essence an interconnected network of multiplexers, this section first reviews basic multiplexing methods and then covers point-to-point and point-to-multipoint switching functions. All networking devices use one or more of these multiplexing and/or switching techniques. Small, simple devices may use only a single technique, while a large, complex device may combine several of these techniques to implement very-high-capacity systems.

Multiplexing Methods Summarized

There are five basic multiplexing methods: space, frequency, time, address, and code. This sequence is also the historical order in which communications networks employed these techniques. Space, frequency, and time division multiplexing all occur at the physical level. Address switching, or label swapping, and code division multiplexing occur at a logical level. Address switching is the foundation of packet switching, Frame Relaying, ATM cell switching, and MPLS label switching.

Space Division Multiplexing (SDM)

An example of space division multiplexing is where multiple, physically separate cables interconnect two pieces of equipment. “Space” implies that there is physical diversity between each channel. The original telephone networks, where a pair of wires connected each end user to communicate, is an example of one of the first uses of space division multiplexing. This approach quickly becomes impractical, as evidenced by old photographs of the sky of major metropolitan cities blackened out by large numbers of wire pairs strung overhead using space division multiplexing. Early data communications ran a separate cable from every terminal back to the main computer, which is another example of space division multiplexing. When there were only a small number of terminals, this was not too much of a burden. Obviously, since each interconnection requires a separate physical cable, SDM does not scale well to large networks.

Frequency Division Multiplexing (FDM)

As transmission technology matured, engineers discovered how to multiplex many analog conversations onto the same cable, or radio spectrum, by modulating each signal by a carrier frequency. Modulation translated the frequency spectrum of the baseband voice signal into a large number of distinct frequency bands. This yielded a marked increase of efficiency and worked reasonably well for analog signals. However, FDM relied on analog electronics that suffered from problems of noise, distortion, and crosstalk between channels that complicated data communications. FDM also made a brief foray into LANs

with Wang Laboratories' Wang Net. This technology required constant tweaking and maintenance, eventually losing out to Ethernet, Token Ring, and FDDI.

Time Division Multiplexing (TDM)

The next major innovation in multiplexing was motivated by the need in the late 1950s to further increase the multiplexing efficiency in crowded bundles of cables in large cities. This entirely digital technique made use of emerging solid-state electronics. *Time division multiplexing (TDM)* first converts analog voice information to digital information prior to transmission using pulse code modulation (PCM). Although this technique was relatively expensive, it did cost less than replacing existing cables or digging larger tunnels in New York City. Since then, TDM has become the prevalent multiplexing method in all modern telecommunications networks. We now take for granted the fact that the network converts every voice conversation to digital data, transmits it an arbitrary distance, and then converts the digits back to an audible analog signal. The consequence is that the quality of a voice call carried by digital TDM is now essentially independent of distance. This performance results from digital repeaters that decode and retransmit the digital signal at periodic intervals to achieve extremely accurate data transfer. Data communications is more sensitive to noise and errors than digitized voice but reaps tremendous benefits from the deployment of TDM infrastructure in public networks. In theory, TDM may also be applied to analog signals; however, this application was never widely used.

Address or Label Multiplexing

Address, or label, multiplexing was first invented in the era of poor-quality FDM analog transmission. A more common name for address multiplexing is *asynchronous time division multiplexing (ATDM)*, of which an example appears later in this chapter. Transmission was expensive, and there was a need to share it among many data users. Each "packet" of information was prefixed by an address that each node interpreted. Each node decided whether the packet was received correctly and, if not, arranged to have it resent by the prior node until it was received correctly. SNA, DECNET, and X.25 are early examples of address multiplexing and switching. More recent examples are Frame Relay, ATM, and Multiprotocol Label Switching (MPLS). The remainder of this book covers the address multiplexing method in great detail.

Code Division Multiple Access (CDMA)

Code division multiplexing, also called *spread spectrum communications*, utilizes a unique method to allow multiple users to share the same broadcast communications medium. Also called Code Division Multiple Access (CDMA), this technique works well in environments with high levels of interference. CDMA transmissions are also difficult to detect because the energy transmitted is spread out over a wide frequency passband. This means that others cannot detect the transmission easily, or else the interference between transmitters stays at a very low level.

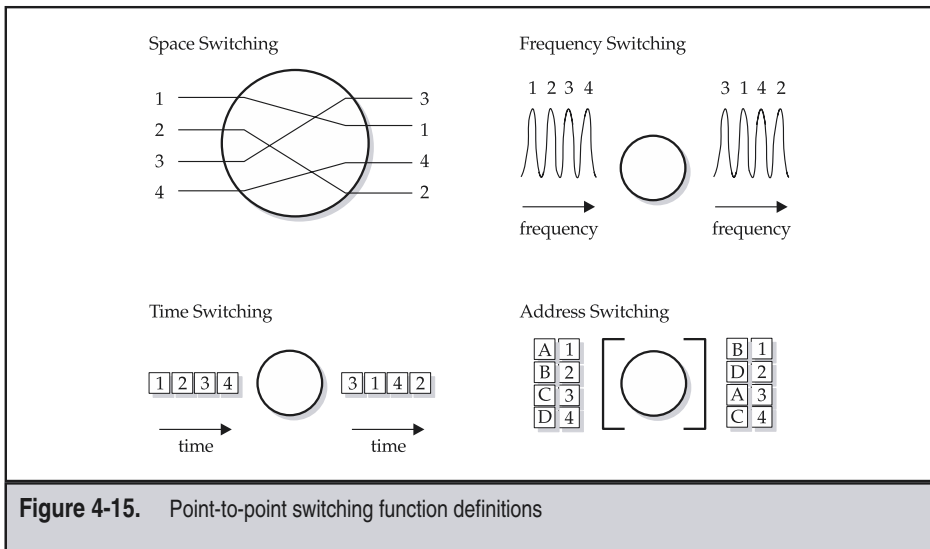
Point-to-Point Switching Functions

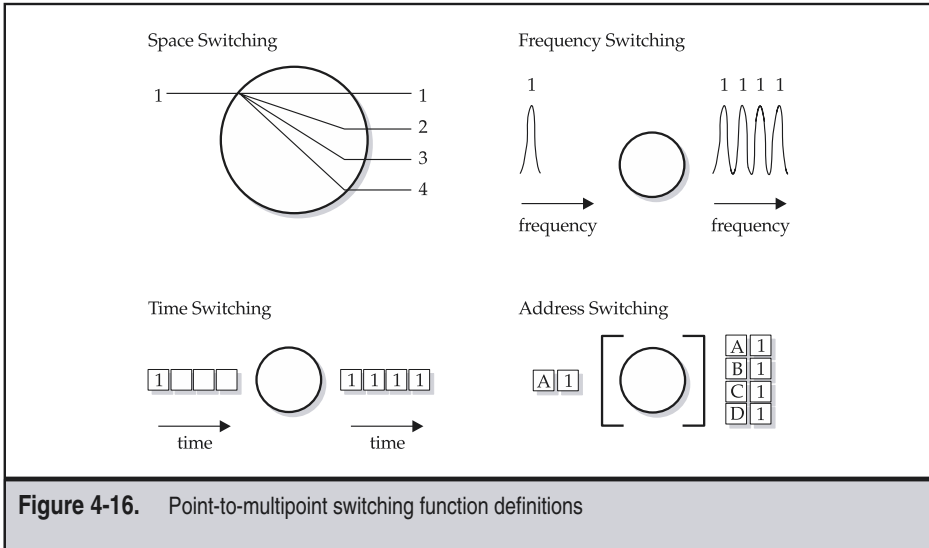
Figure 4-15 illustrates the four basic kinds of point-to-point connection functions that can be performed by a multiplexer or switch.

Space division switching delivers a signal from one physical (i.e., spatial) interface to another physical interface. One example is a copper crosspoint switch. Time division switching changes the order of time slots within a single spatial data stream, organized by the time division multiplexing (TDM) method. Frequency (or wavelength) switching translates signals from one carrier frequency (wavelength) to another. Wavelength division multiplexing (WDM) in optical fiber transmission systems uses this method. Finally, address switching changes the address field in data packets, which may be further multiplexed into spatial, time, or frequency signals. This book focuses on this switching method, as applied to packet, frame, and cell switching.

Point-to-Multipoint Switching Functions

Figure 4-16 illustrates the extension of switching from the case of point-to-point to the broadcast, or point-to-multipoint case. A space division broadcast switch replicates a single input signal on two or more outputs. A simple example is a coaxial television signal splitter that delivers the same signal to multiple outputs. TDM broadcast switching fills multiple output time slots with the data from the same input. FDM broadcast switching replicates the same signal on multiple output carrier frequencies. Address broadcast switching fills multiple packets with different addresses with identical information from the same input packet.





Examples of Multiplexing

A *multiplexer* is essentially a very simple switch consisting of a multiplexing function and a demultiplexing function connecting a single trunk port in the network to many access ports connected to individual traffic sources, as illustrated in Figure 4-17. The parallelogram symbol with the small end on the side of the single output (called the trunk side) and the large end on the side with multiple interfaces (called the access side) frequently denotes a multiplexer in block diagrams. The symbol graphically illustrates the many-to-one relationship from the many ports on the access side to the single port on the trunk side, as well as the one-to-many relationship from the trunk side to the access side.

The multiplexing function shares the single output among many inputs. The demultiplexing function has one input from the network, which it distributes to many access outputs. The multiplexing and demultiplexing functions can be implemented by any of the generic switching functions described in the previous section. Usually, the same method is used for both the multiplexing and demultiplexing functions so that the multiplexing method used on each of the interfaces is symmetrical in each direction. Generally, the overall speed or capacity of each port on the access side is less than that on the trunk side. For example, different levels in the time division multiplexing (TDM) hierarchy operate at increasingly higher speeds by aggregating multiple lower-speed TDM signals together. We give more detailed examples for each of the generic methods described in the preceding sections.

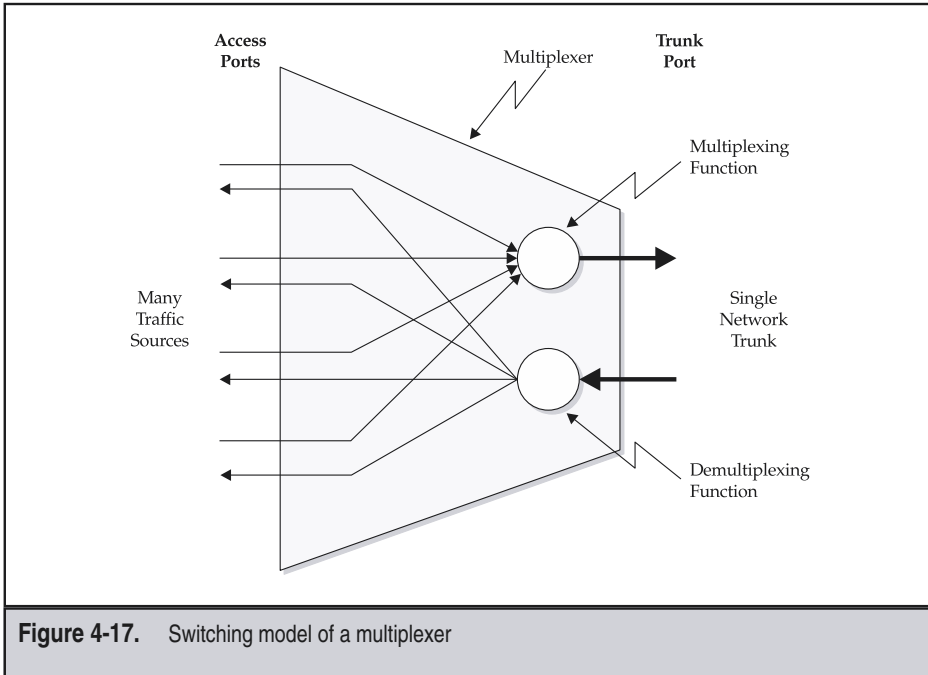


Figure 4-17. Switching model of a multiplexer

Multiplexers share a physical medium between multiple users at two different sites over a private line, with each pair of users requiring some or all of the bandwidth at any given time. Many simple multiplexers statically assigned a fixed amount of capacity to each user. Other multiplexing methods statistically assign capacity to users according to demand to make more efficient use of the transmission facilities that interface to the network. You'll see these called *statistical multiplexers* in the technical literature. TDM is often used to reduce the effective cost of a private access line or international private line by combining multiple lower-speed users over a single higher-speed facility.

Frequency Division Multiplexing (FDM)

Analog telephone networks made extensive use of *frequency division multiplexing (FDM)* to aggregate multiple voice channels into larger circuit groups for efficient transport. FDM multiplexes 12 voice-grade, full-duplex channels into a single 48 kHz bandwidth group by translating each voiceband signal's carrier frequency. These groups are then further multiplexed into a mastergroup made up of 24 groups. Multiple mastergroup analog voice signals are then transmitted over analog microwave systems. A lower-frequency analog microwave spectrum was used to frequency division multiplex a DS1 digital data stream in a technique called Data Under Voice (DUV).

Wavelength division multiplexing (WDM) on optical fibers is analogous to FDM in coaxial cable and microwave systems. Optical fiber is most *transparent* in two windows centered around the wavelengths of 1300 and 1550 nm (10^{-9} m) as shown in the plot of loss versus wavelength in Figure 4-18 [Personick 85]. The total bandwidth in these two windows exceeds 30,000 GHz. Assuming 1 bps per Hertz (Hz) would result in a potential bandwidth of over 30 *trillion* bps per fiber!

Recall the basic relationship from college physics $\lambda v = c$, where λ is the wavelength in billionths of a meter (i.e., a nanometer or nm), v is the frequency in billions of cycles per second (gigahertz or GHz), and c is the speed of light in a vacuum (3×10^8 m/s). Applying this formula, the carrier frequency v at the center of the 1300 nm window is 2300 GHz and 1900 GHz in the 1550 nm window. The available spectrum for signal transmission is 18,000 GHz in the 1300 nm window and 12,500 GHz in the 1550 nm window, as shown in the figure. Chapter 23 defines the concepts of optical signals and their frequency spectra. This means that at a spectral efficiency of 1 bps per Hertz, a single fiber pair carrying these two bands could theoretically carry approximately 30 Terabits (1×10^{12}) per second of duplex traffic. The sharp attenuation peak at 1400 nm is due to residual amounts of water

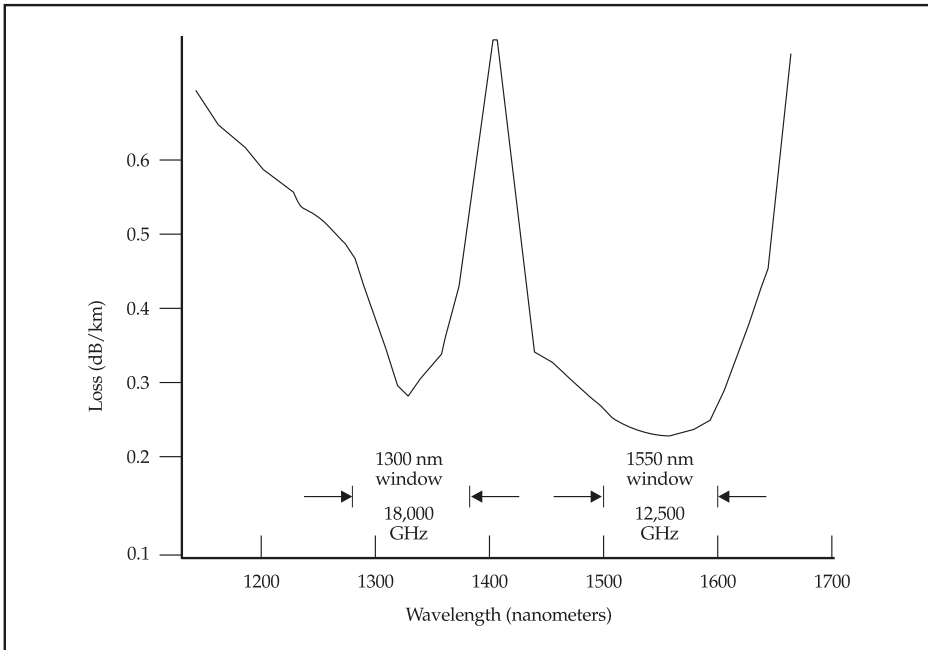


Figure 4-18. Optical fiber transfer characteristic

(an OH radical) still present in the glass. Continuing improvements in optical fiber manufacturing will likely make even more optical bandwidth accessible in the future. Commercial long-haul fiber optic transmission is now using between two and eight wavelengths per fiber, in what is called wideband WDM, in these two windows. Implementations of dense WDM (DWDM), supporting up to 100 optical carriers on the same fiber, were available at the time of writing.

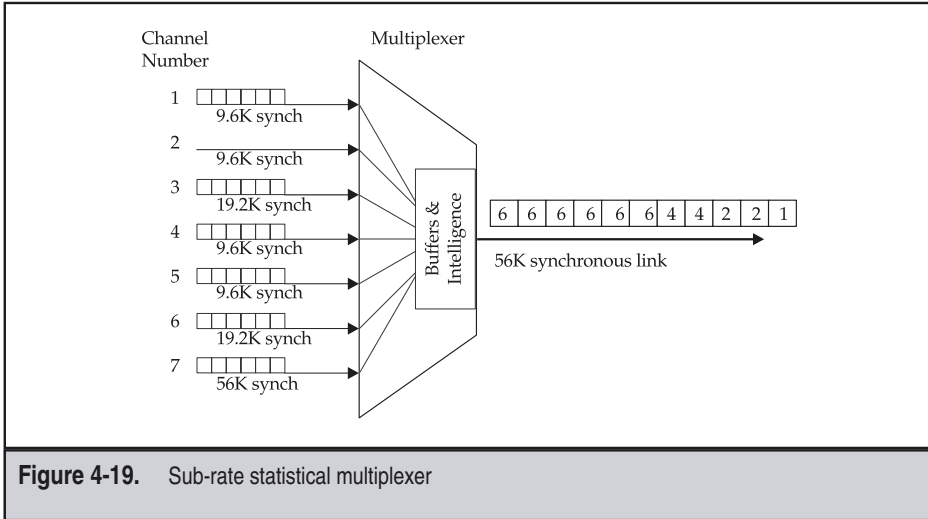
Time Division Multiplexing (TDM)

Time division multiplexing (TDM) was originally developed in the public telephone network in the 1950s to reduce costs in metropolitan area networks. It also eliminated FDM filtering and noise problems when multiplexing many signals onto the same transmission medium. In the early 1980s, TDM networks using smart multiplexers began to appear in some private data networks, forming the primary method to share costly data transmission facilities among users. In the last decade, time division multiplexers have matured to form the basis of many corporate data transport networks. The premier example of TDM is DS1 and E1 multiplexing; Chapter 6 describes this for the ISDN Primary Rate Interface (PRI).

Address Multiplexing

An early application of *address (or label) multiplexing* is found in statistical multiplexing equipment, also called statistical time division multiplexing (STDM), or asynchronous time division multiplexing (ATDM). These devices operate similar to TDM, except they dynamically assign the available time slots only to users who need data transmission. Gains of up to 2:1 are achieved for voice transmission by utilizing all available time slots, rather than wasting them on users who are not speaking. Higher or lower statistical multiplex gains can be obtained for data traffic depending upon the burstiness (peak-to-average statistics) of the data traffic. Part 6 covers these tradeoffs in detail. The net effect is a potential increase in overall throughput for users since time slots are not “reserved” or dedicated to individual users—thus, dynamic allocation of the unused bandwidth achieves higher overall throughput. Figure 4-19 shows an example of a statistical multiplexer that takes multiple low-speed synchronous user inputs for aggregation into a single 56 Kbps synchronous bit stream for transmission. The methods used to interleave the various channels in statistical multiplexers include bit-oriented, character-oriented, packet-oriented, and cell-oriented techniques, each requiring buffering and more overhead and intelligence than basic time division multiplexing.

Figure 4-19 shows an excerpt from the output of a statistically multiplexed data stream. In a statistical multiplexer, the output bandwidth is *less* than the aggregate input bandwidth. This is done by design, assuming that not all input channels will be transmitting at the same time when each channel is sampled for transmission. Thus, the output synchronous data stream allocates bandwidth only to users who require it. It does not waste time slots by dedicating bandwidth to users who do not require it at the moment. Note in the example in Figure 4-19 that channels 1, 2, 4, and 6 are transmitting, together utilizing 48 Kbps of the available 56 Kbps trunk bandwidth. Using the same example, if



channels 3 (19.2 Kbps) and 7 (56 Kbps) were also to transmit data at the same instant, the total peak transmission rate of 123.2 Kbps exceeds the 56 Kbps trunk speed of the multiplexer. The statistical multiplexer overcomes brief intervals of such demand by buffering information and transmitting it when other sources fall idle. As we shall see, all packet switching technologies utilize the concept of statistical address multiplexing in one way or another.

Space Division Multiplexing

Space division multiplexing essentially reduces to the discipline of cable management. This can be facilitated by mechanical patch panels, or increasingly so by automatically controlled optical and electronic patch panels. Largely, space division multiplexing is falling out of favor; space division switching or other, more efficient types of multiplexing typically replace multiple parallel cables in many network designs.

Code Division Multiplexing

In *code division multiplexing*, each user bit is modulated by a high-rate “chipping” signal. Each user transmits a unique pseudo-random coded signal at the chip rate for each user data bit. The pseudo-random sequences are chosen so that they are easily generated in hardware. The sequences of ones and zeros cancel each other out for all except the transmitting user. The most commonly used pseudo-noise sequences are generated by maximal-length shift registers [Proakis 83]. These particular sequences have the useful property that each unique user code is a cyclic shift of another sequence.

Code division multiplexing performs well on channels with high levels of interference, either arising naturally, generated by malicious jammers, or just resulting from simultaneous transmissions by multiple users. Hence, satellite communications, military communications, wireless networks, and Personal Communications Service (PCS) telephones employ code division multiplexing. Some wireless LAN and wireless ATM systems employ code division multiplexing.

Examples of Switching

This section gives an example for each of the major switching techniques: space, time, address, and frequency. The examples chosen define terminology and illustrate concepts as background for material in subsequent chapters. Furthermore, as we shall see, fundamental limits of the technique limit the maximum sizes and highest port speeds of devices built using the space, time, frequency, and address multiplexing techniques.

Space Division Switching

Figure 4-20 illustrates a simple two-input, two-output crossbar network, using the crosspoint nodal function. An example connection is shown by the boldface lines and control inputs. Notice that this 2×2 switch matrix requires four switching elements.

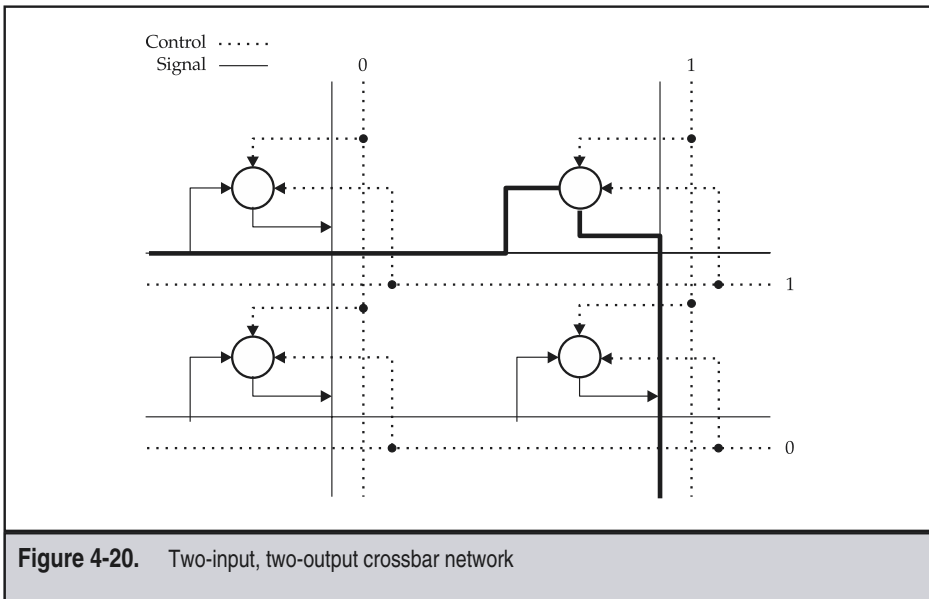


Figure 4-20. Two-input, two-output crossbar network

Classical space division switch fabrics are built from electromechanical and electronic elements to provide the crosspoint function. Electrical cross-connect switches operate at speeds of many Gbps by running multiple circuit traces in parallel.

Future technologies involving optical crosspoint elements with either electronic or optical control are being researched and developed. Optical switches generally have higher port speeds but probably smaller total numbers of ports than their electrical counterparts. Electrical and optical space division switches currently scale to capacities greater than 1 trillion bits per second (Tbps).

Examples of space division switches are matrix switches, high-speed digital cross-connects, and optical switches that switch the entire spectrum of an incoming optical fiber to an outgoing fiber, for example, using electronically controlled mirrors. The usage and control of optical switches is a topic of tremendous interest in scaling large IP networks. Many space division switches employ multiple stages of crosspoint networks to yield larger switch sizes.

Time Division Switching

The operation of current digital telephone switches may be viewed as an interconnected network of special-purpose computers called *time division switches (TDSs)* [Keiser 85]. Recall that a switch is basically a set of interconnected multiplexers; hence, time division switches use time division multiplexing (TDM) as the interface protocol. Our description of time division switching operation references Figure 4-21. Each TDM frame has M time slots. The input time slot m , labeled $I(m)$, is stored in the input sample array $x(t)$ in position m . The output address memory $y(t)$ is scanned sequentially by increasing t from 1 to M each frame time. The contents of the address array $y(t)$ identify the index into the input time slot array x that is to be output during time slot t on the output line. In the example in Figure 4-21, $y(n)$ has the value m , which causes input time slot m to be switched to output time slot n . Note that the input sample array must be double-buffered in an actual implementation so that time slot phase can be maintained for inputs and outputs with different frame clock phases.

This TDS function is performed for M time slots, that is, once every frame time. This must occur in less than $\tau = 125 \mu\text{s}$ ($1/8000$) for all slots, $n = 1, \dots, M$. The maximum TDS size is therefore determined by the TDS execution rate, I instructions per second (or equivalently I^{-1} seconds per instruction); then the TDS switch size M must satisfy the inequality $M \leq \tau I$.

The TDS is effectively a very special-purpose computer designed to operate at very high speeds. For I ranging from 100 to 1000 MIPs, the maximum TDS switch size M ranges from 12,500 to 125,000, which is the range of modern single-stage time division switches (TDSs). Larger time division switches can be constructed by interconnecting TDS switches via multiple-stage crosspoint-type networks [Keiser 85].

Usually, some time slots are reserved in the input frame in order to be able to update the output address memory. In this way, the update rate of the switch is limited by the usage of some slots for scheduling overhead.

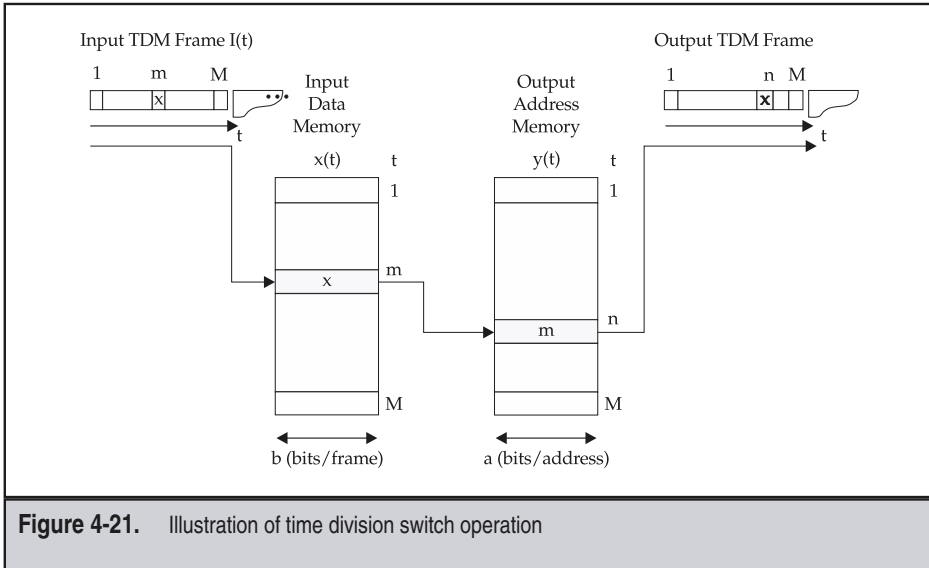


Figure 4-21. Illustration of time division switch operation

Address Switching

Address (or label) switching operates on a data stream in which the data is organized into packets, each with a header and a payload. The header contains address information used in switching decisions at each node to progress the packet toward the destination on a hop-by-hop basis. The address determines which physical output the packet is directed to, along with any translation of the header address. All possible connection topologies can be implemented within this switching architecture: point-to-point, point-to-multipoint, multipoint-to-point, and multipoint-to-multipoint. We illustrate these topologies in the following example.

Figure 4-22 illustrates four interconnected address switches, each with two inputs and two outputs. Packets (either fixed or variable in length) arrive at the inputs as shown on the left-hand side of the figure with addresses indicated by letters in the header symbolized by the white square prior to each shaded payload. The payload shading is carried through the switching operations from left to right to allow the reader to trace the switching result of the address switches visually. The input address indexes into a table using the column labeled In@, which identifies the address for use on output in the column Out@, and the physical output port on which the packet is sent in the column labeled Port. For example, the input packet addressed as A is output on port 1 using address M. Conceptually, each switch functions as a pair of busses that connects to the output port buffers. Each switch queues packets destined for a particular output port prior to transmission. This buffering reduces the probability of loss when contention occurs for the

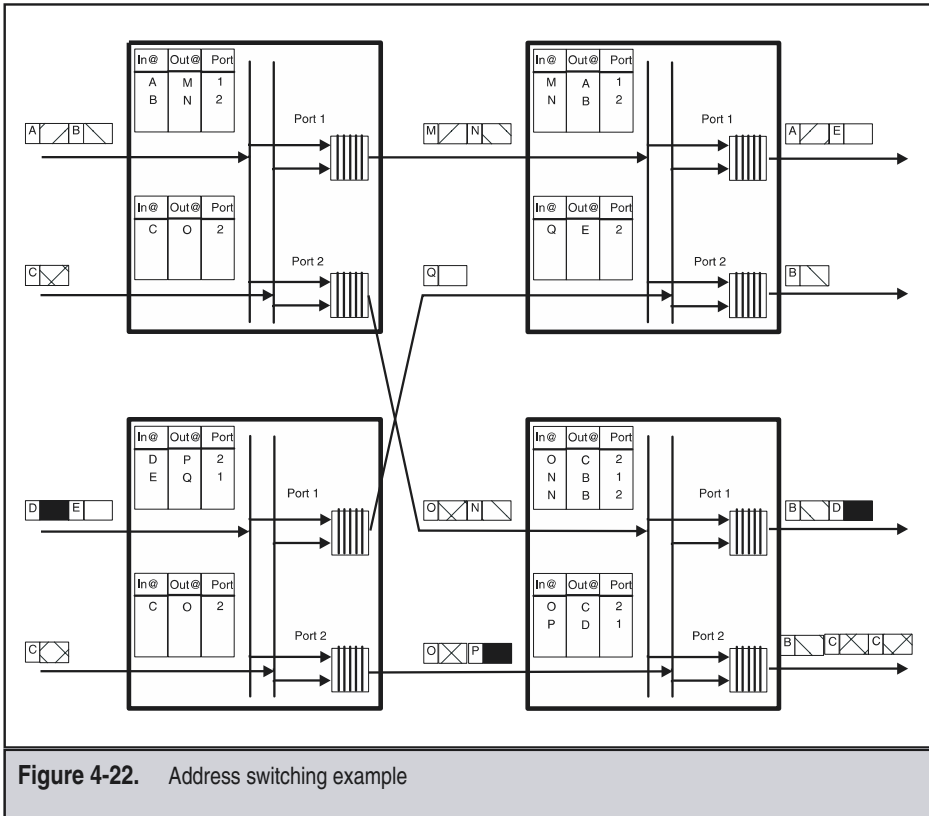


Figure 4-22. Address switching example

same output port. Chapter 24 presents an analysis of the loss probability due to buffer overflow for the main types of switch architectures used in real-world ATM switches and MPLS switching routers. At the next switch, the same process occurs until the packets are output on the right-hand side of the figure.

The packets with header addresses labeled A, D, and E form point-to-point connections. The packets labeled B form point-to-multipoint connections. The packets labeled C form multipoint-to-point connections. Currently, address switching operates at electrical link speeds of up to 2.4 Gbps for both ATM and packet-switching systems. Of course, address switching and multiplexing are at the heart of ATM and MPLS, which subsequent chapters cover in detail. Specifically, Part 8 discusses ATM switch and MPLS switching router fabric designs that scale to total capacities in excess of one trillion bits per second using combinations of address and space switching.

Frequency/Wavelength Switching

A significant amount of research on all-optical networks [Jajczyk 93], [Green 92] using wavelength division multiplexing (WDM) has resulted in commercial optical multiplexing and switching systems. The basic concept is a shared medium, all-photonic network interconnecting a number of optical nodes or “end systems,” as shown in Figure 4-23.

The optical end system nodes transmit on at least one wavelength and receive on at least one wavelength. The wavelength for transmission and reception may be tunable, currently in a time frame on the order of milliseconds, with an objective of microseconds. The end systems may also be capable of receiving on more than one wavelength. The wavelengths indicated by the subscripts on the character λ are used in the next example of a multiple-hop optical network.

If the end system cannot receive all of the other wavelengths transmitted by other nodes, then the network must provide some means to provide full interconnectivity. One early method proposed and implemented was that of multiple-hop interconnections. In a multiple-hop system, each end system also performs a routing function. If an end system receives a packet that is not destined for it, it forwards it on its transmit wavelength. Eventually the packet reaches the destination, as shown in the trellis drawing of Figure 4-24. In this example, each node transmits and receives on only one wavelength. For example, Node 1 transmits on wavelength λ_1 and receives on λ_4 . For example, in order for station 1 to transmit to station 4, it first sends on wavelength λ_1 , which Node 2 receives. Node 2 examines the packet header, determines that it is not the destination, and retransmits the

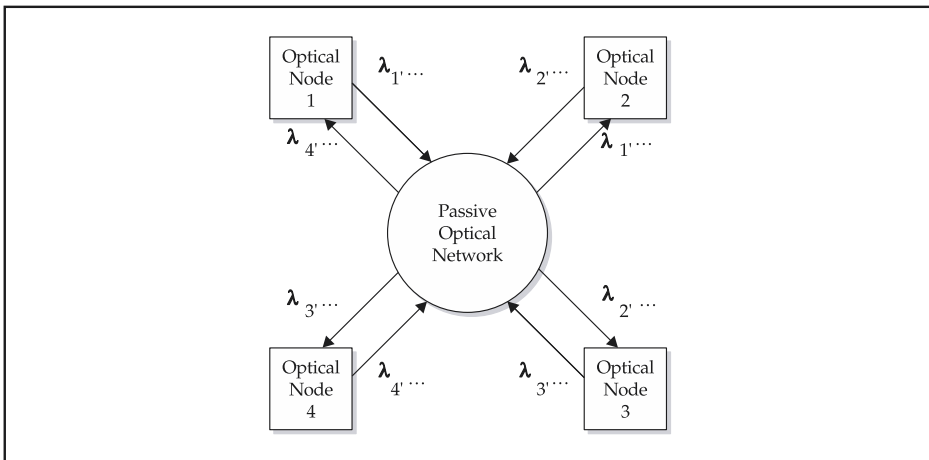
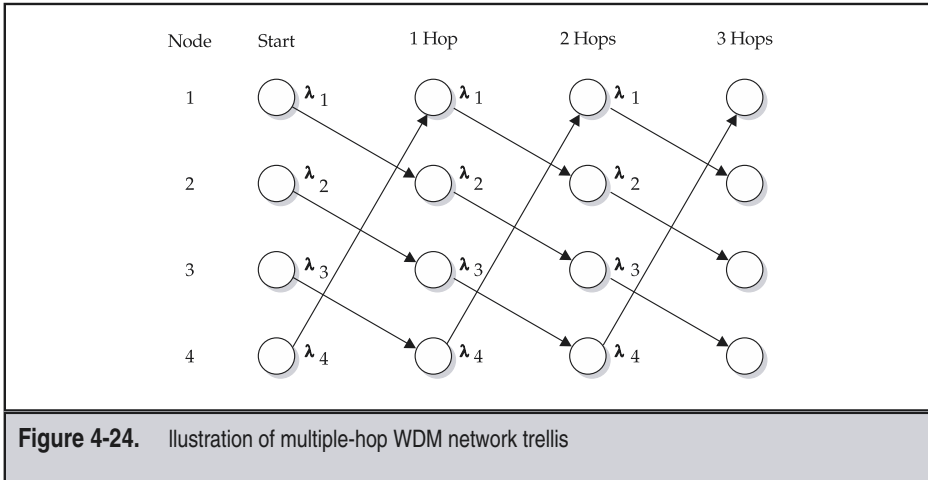


Figure 4-23. Illustration of optical WDM network



packet on wavelength λ_2 . Node 3 receives the packet, examines the packet header, and forwards it on λ_3 , which the destination Node 4 receives after taking three hops across the network.

This multiple-hop process makes inefficient use of the processing power of each node, especially when the number of nodes is large; therefore, research has focused on single-hop designs. In these designs, the tunable transmitter and receiver are often employed. There is a need for some means to allocate and share the bandwidth in the optical network. Connection-oriented signaling has been tried, but the user feedback is that packet switching, and not circuit switching, is required because the connection setup time is unacceptable. Fixed allocation of bandwidth in a time-slotted manner is also not desirable. Dynamic scheduling and collision avoidance hold promise for a solution to the demands of very-high-speed networking.

REVIEW

This chapter began with a discussion of the five major network topologies: point-to-point, multipoint, star, ring, and mesh. The treatment then moved to a discussion of the relationship between DTE and DCE connections. We clarified the usage of the terms *asynchronous* and *synchronous* in data communications. The text compared the commonly used definition of asynchronous data in modem communications with the meaning of asynchronous in ATM. We showed that ATM and MPLS are specialized forms of address switching derived from packet switching, as described by several examples in this chapter. The major principles of multiplexing and switching were also introduced, followed by a number of examples illustrating the major digital communications technologies in use today.



CHAPTER 5

Basic Protocol Concepts

Protocols shape our everyday lives. A *protocol* is similar to a language, conveying meaning and understanding through some form of communication. Computer communication protocols are sets of rules governing the exchange of messages that define the way machines communicate and behave. Much as people require a shared language to conduct intelligent discourse, in order for one computer to talk to another, each must be able to understand the other's protocol. Protocols play an important role in data communications; without them, islands of users would be unable to communicate.

This chapter begins with an introduction to packet switching, articulating the reasons for this new suite of protocols, defining some basic principles, and discussing how changes in the telecommunication environment impacted the evolution of packet-switching protocols. Next, the text explains the key concept of layered models, where lower-layer protocols provide services to the next higher layer. The concept of layered protocols was developed in parallel by the Internet Research Committee and the International Organization for Standardization (ISO) in the 1970s. This is the source of terminology widely used in networking communications, including the TCP/IP protocol suite and the ISO's seven-layer Open Systems Interconnection (OSI) Reference Model. We then summarize how protocol suites collect sets of layered protocols into a single group; for example, IBM's well-known Systems Network Architecture (SNA), the IEEE 802 series of local area networking specifications, and the Internet Protocol (IP) suite. The computer communication community predominantly adopted the LAN and IP protocol suites, but the numbered OSI layer terminology is also widely used in a generic sense. Furthermore, the concept of layering enabled the entire industry of multiprotocol routing. The chapter concludes with a comparison of connection-oriented network services (CONS) and connectionless network services (CLNS) as an introduction to the world of packet switching.

A BRIEF HISTORY OF PACKET SWITCHING

Data communications is inherently bursty and has multiple destinations, and therefore setting up a circuit-switched connection for each data transfer is inefficient and time consuming. In response to this basic problem, packet-switched networks have evolved for over 40 years and form the basis of most advanced data communications networks today. Packet switching initially provided the network environment needed to handle bursty, terminal-to-host data traffic over noisy analog telephone network facilities. Packet switching has been widely implemented, especially in Europe, where it constitutes the majority of public and private data services. Most of the current telecommunications infrastructure consists of parallel networks largely optimized for one specific service. The majority of voice services are offered over the traditional TDM voice circuit-switched network. The Internet supports primarily best-effort IP service; whereas frame relay, ATM, and MPLS support traffic with more specific quality assurances. Convergence of more service types onto fewer networks is one of the current challenges, driven by new service opportunities and a promise of realizing significant economic network management and operational efficiencies. We will return to the topic of convergent networks when we discuss ATM and MPLS functionality and standards in Part 3.

Early Reasons for Packet Switching

Paul Baran and his research team at the RAND Corporation invented the concept of packet switching in the early 1960s as a secure, reliable means of transmitting military communications that could survive a nuclear attack. The solution was to segment a longer message into many smaller pieces and then wrap routing and protocol information around these pieces, resulting in data “packets.” The routing and control information ensured the correct and accurate delivery and eventual reassembly of the original message at the end-user destination. Early systems had packets with a fixed maximum size assigned, typically 128 or 256 bytes. Through the use of multiple independent packets, the entire message could be transmitted over multiple paths and diverse facilities to a receiver that reassembled the original message.

The next step in packet-switch history was taken when the Advanced Research Projects Agency (ARPA) of the United States Department of Defense (DoD) implemented packet switching to handle computer communications requirements, thus forming the basis for the network called the ARPANET. Packet switching was chosen as the method to implement WAN computer communications, which mainly consisted of connecting large computing centers. Soon after ARPANET, many commercial companies also developed packet-based networks.

The early days of computing also saw the development of new interfaces and data communication protocols by each major computer manufacturer. Large computer manufacturers, such as IBM and DEC, developed protocols that were standardized, but only across their own product line. This tactic often locked a user into a single, proprietary protocol. Indeed, a key objective of the OSI standardization effort was to enable standard computer communication interfaces and protocols in a multiple vendor environment.

Early packet-switching systems targeted terminal-to-host communications. The typical transaction involved the user typing a few lines, or even just a few characters, and then sending a transaction to the host. The host would then return a few lines, or possibly an entire screen’s worth of data. This terminal-host application was very bursty; that is, the peak transmission rate of each terminal was much greater than its average rate. Packet-switching equipment statistically multiplexed many such bursty users onto a single expensive transmission facility.

As the number of computers, applications, and people using computers increased, the need for interconnection increased, creating the accelerating need for bandwidth. Similar to the response to growth within telephony, change was necessary to optimize networking, and it quickly became absurd to have a dedicated circuit to connect every pair of computers that needed to communicate. Packet-switching and routing protocols were developed to connect terminals to hosts, and hosts to hosts.

Principles of Packet Switching

Several factors created the need for packet switching: the need to create standard interfaces between computing devices, the challenge of extending computer communication over noisy analog transmission facilities, a requirement to make more efficient use of

expensive transmission bandwidth, and the demand for a means to enable the interconnection of a large number of computing devices.

Packet switching is an extremely important special case of the general address multiplexing and switching method described in Chapter 4. Packet switching provides a service in which blocks of user data are conveyed over a network. User data, such as files, are broken down into blocks of units called “payload” information. Packet switching adds overhead to the user data payload blocks, resulting in a combination called a *packet*. All of the protocols studied in this book have this characteristic, including SNA, IEEE 802.X LAN protocols, X.25, IP, frame relay, SMDS, Ethernet, FDDI, Token Ring, ATM, and MPLS.

The functions implemented using the packet overhead are either data link layer, packet layer, or transport layer from the OSI Reference Model (OSIRM), as described later in this chapter. Older protocols, such as X.25 and IP, perform both the data link layer and packet layer functions. Newer protocols, such as frame relay, SMDS, ATM, and MPLS, perform only a subset of data link layer functions, but with addressing functions that have a network-wide meaning.

Data link layer functions always have a means to indicate the boundaries of the packet, perform error detection, provide for multiplexing of multiple logical connections, and provide some basic network management capability. Optional data link layer functions are flow control, retransmission, command/response protocol support, and data link-level establishment procedures.

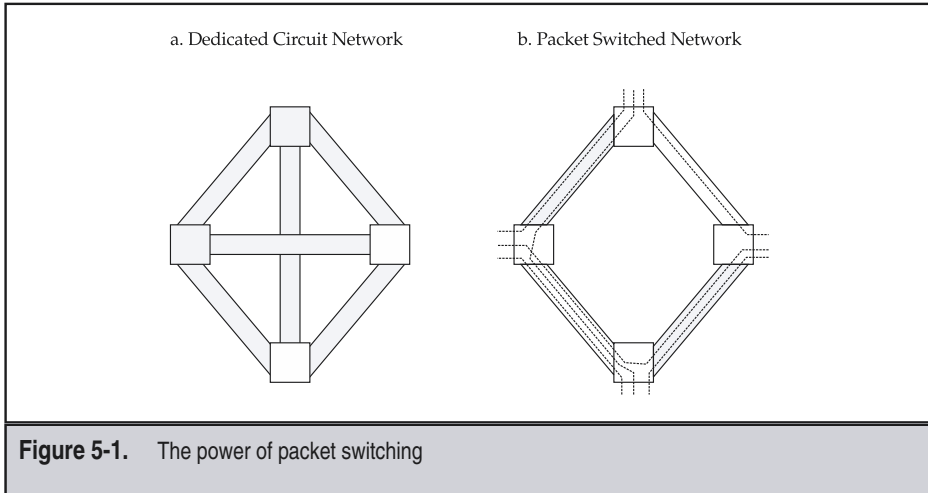
Packets were designed with cyclical redundancy check (CRC) fields that detected bit errors. Early packet switches (e.g., X.25) retransmitted packets that were corrupted by errors on a link-by-link basis. The advent of low bit-error rate, fiber optic transmission media made implementation of such error detection and retransmission cost-effective in the end system, since errors rarely occurred, freeing protocol implementations like FR, ATM, and MPLS from performing link-level error detection and correction.

Network layer functions always have a means to identify a uniquely addressed network station. Optional network layer functions include retransmission, flow control, prioritized data flows, automatic routing, and network layer connection-establishment procedures.

Furthermore, packet switching enables statistical multiplexing by allowing multiple logical users to share a single physical network access circuit. Buffers in the packet switches reduce the probability of loss during rare intervals when many users transmit simultaneously. Packet switches control the quality provided to an individual user by allocating bandwidth, allocating buffer space, policing the traffic offered by users, or affording flow control. Part 6 covers the application of these methods to ATM and MPLS.

Packet switching also extends the concept of statistical multiplexing to an entire network. In order to appreciate the power of packet switching, compare the full-mesh network of dedicated circuits in network Figure 5-1a with the packet-switched network in Figure 5-1b. The dedicated-circuit network has three lines connected to every host, while the packet-switched network has only one, connecting the host to the packet switch. A virtual circuit connects every user through the packet-switched network, as shown by the lines within the physical trunks.

The dedicated-circuit network has higher overall throughput but will not scale well. The packet-switched network requires additional complexity in the packet switches and



has lower throughput, but it reduces circuit transmission costs, as shown in the example in Figure 5-1 with the nodes placed on the corners of a square. Sharing of network resources allows savings over the cost of many dedicated, low-speed communications channels, each of which is often underutilized the majority of the time. Virtual circuits are a concept that carries through into frame relay, ATM, and MPLS networking.

Packet switching employs queuing to control loss and resolve contention at the expense of added, variable delay. The packet may take longer to reach its destination with packet switching, but the chances of loss are lower during periods of network congestion, assuming a reasonable buffer size. Packet data protocols employ two types of flow and congestion control: implicit and explicit congestion notification.

Implicit congestion notification usually involves a layer 4 transport protocol, such as the Transmission Control Protocol (TCP), in either the network device or the user premises equipment. These protocols adaptively alter the rate at which packets are sent into the network by estimating loss and delay.

Explicit congestion notification occurs when the protocol notifies the sender and/or receiver of congestion in the network. If the sender or receiver reacts to the explicit indication of congestion quickly enough, it avoids loss entirely. Part 5 covers the subject of implicit and explicit flow and congestion control in detail.

Darwin's Theory and Packet-Switching Evolution

Darwin spent nearly as much time to arrive at his theory of evolution as it took for engineers to conceive of implementing packet switching throughout the world of data communications. The basic tenets of Darwin's theory of evolution are natural selection, survival of the fittest, and the need to adapt to a changing environment. In the communications jungle, packet switching has all of these attributes.

This section takes the reader through a brief summary of the genealogy of packet switching with reference to Figure 5-2. The genesis of packet switching began with two proprietary computer communication architectures: IBM's Systems Network Architecture (SNA) and DEC's Digital Network Architecture (DNA). Standards bodies refined the Synchronous Data Link Control (SDLC) protocol from SNA, resulting in the High-Level Data Link Control (HDLC) protocol—which begat X.25 and Link Access Procedure D (LAP-D) within ISDN. Frame relay evolved as a leaner, meaner LAP-D protocol. OSI adopted the X.25 protocol as the first link and packet layer standard. Combining the conventions from preceding protocols and the concepts of hardware-oriented Fast Packet Switching (FPS) resulted in the Distributed Queue Dual Bus (DQDB) protocol, which is the basis of the Switched Multimegabit Data Service (SMDS) followed by ATM, and then MPLS. On top of ATM, a number of ATM adaptation layers (AALs) support not only data, but voice and video as well, as described in Part 4. MPLS standards development is following a similar track by defining how various services, including voice, video, and ATM, can be carried over MPLS.

Around the time that the ISO was developing the OSI protocol suite, the U.S. Advanced Research Projects Agency (ARPA) was working on a network, together with universities and industry, that eventually resulted in the suite of applications and higher-level protocols that are based on the highly successful Internet Protocol (IP) version 4.

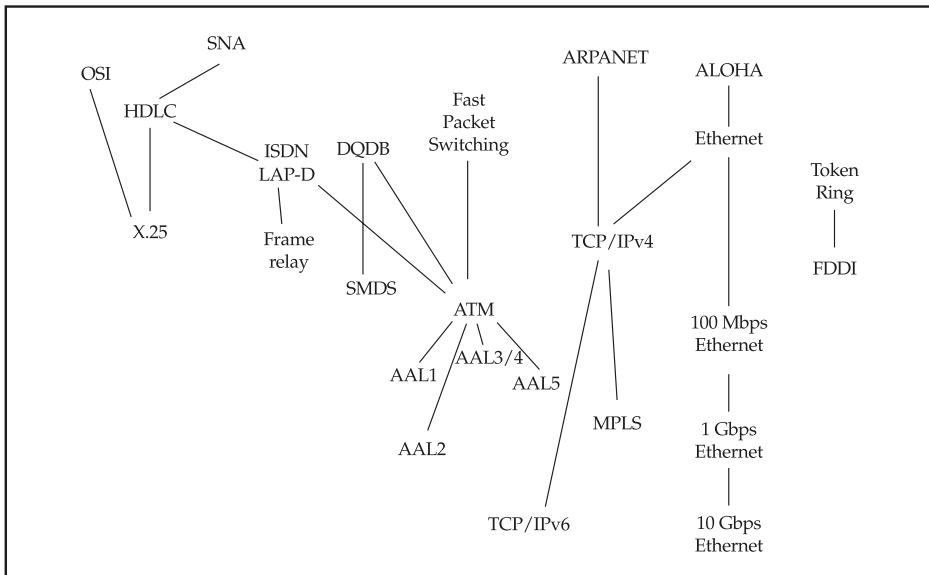


Figure 5-2. Genealogy of packet switching

Ethernet also sprung up at this time as a result of an experimental packet radio communication network in Hawaii called ALOHA. Ethernet then evolved into 100 Mbps Fast Ethernet, and then Gigabit Ethernet with speeds to over one billion bits per second, which was followed shortly thereafter by full standardization of 10 Gbps Ethernet. TCP/IP also adds a much larger address space with version 6, as well as a means to group application flows with similar performance requirements and allocate bandwidth. As we shall see, MPLS evolved initially as a means to more efficiently engineer IP networks, but it also spawned some new applications. Token Ring was also developed shortly after Ethernet, and it has evolved into the higher-speed FDDI.

Packet switching plays an increasingly important role in the rapidly changing environment of distributed processing of the 1990s. Several environmental factors drive the direction of data communications evolution. There is an accelerating need for more bandwidth driven by increasing computing power, increasing need for interconnectivity, and the need to support ever-larger networks where any user or application can communicate with any other. The low error rate of modern fiber optic, satellite, and radio communications enables more cost-effective implementation of higher-speed data communications. The same technology that increases computer power also increases packet-switching performance.

This changing environment creates new opportunities for new species of data communications protocols. The improved quality of transmission facilities alone was a major force in the evolution of IP, frame relay, ATM, and MPLS. These newer protocols are streamlined in that they do not perform error correction by retransmission within the network. The fixed slot and cell size of SMDS and ATM enabled cost-effective hardware implementation of powerful switching machines in the 1990s. As TCP/IP dominated the applications space, the industry responded with hardware that efficiently handled variable-length packets and prefix-based lookups. The increasing capabilities of high-speed electronics are an essential ingredient in IP, ATM, and MPLS devices.

Now that we've surveyed the history, background, and directions of packet switching, let's take a more detailed look at some of the underlying concepts and terminology involved in layered protocols.

BASIC PROTOCOL LAYERING CONCEPTS

Webster's New World Dictionary defines a *protocol* as "a set of rules governing the communications and the transfer of data between machines, as in computer systems." Seeking a means to divide and conquer complex protocols, system designers arrange the communications and data transfers between such machines into logical layers that pass messages among themselves. A convention created by the OSI Reference Model refers to the messages passed between such layers as an *interface*. Each layer has a specific interface to the layer above it and the layer below it, with two exceptions: the lowest layer interfaces directly with the physical transmission medium, and the highest layer interfaces directly with the end-user application. The study of computer communication networking covers primarily the lowest three layers: physical, data link, and network. Devices implementing these protocols may realize these layers in either software or hardware, or a combination of

the two. Typically, hardware implementations achieve much higher speeds and lower delays than software ones do; however, with the increasing performance of microprocessors, this distinction blurs at times. Figure 5-3 illustrates the basic concept of protocol layering that is relevant to the protocols described in this book. Let's now look at these protocol-layering concepts in more detail.

The term *interface* is used in two ways by different standards bodies. First, primarily in the CCITT/ITU view, physical interfaces provide the physical connection between different types of hardware, with protocols providing rules, conventions, and the intelligence to pass data over these interfaces between peer protocol layers. In summary, the CCITT/ITU view is that bits flow over physical interfaces, as shown at the bottom of Figure 5-3. Second, in the OSI view, interfaces exist between protocol layers within end and intermediate systems as indicated in the figure. In this view, *protocol data units (PDUs)*, or

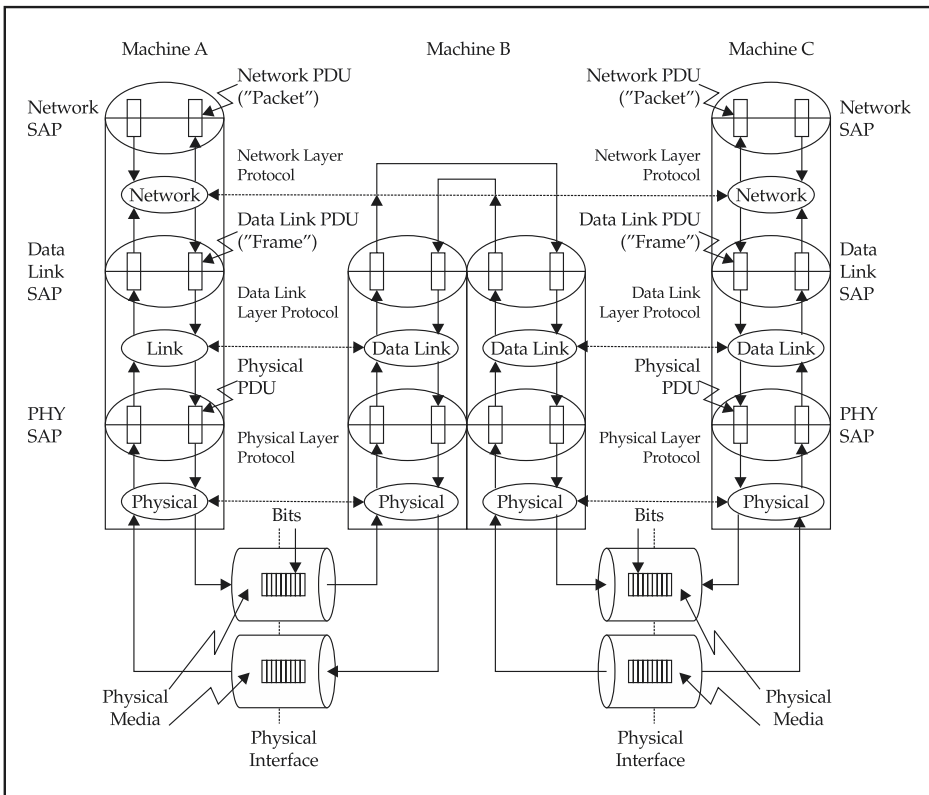


Figure 5-3. Protocol model of the physical, data link, and network layers

messages, pass over protocol interfaces. OSI standards also call the interfaces between layers Service Access Points (SAPs) because they are the points where the higher-layer protocol accesses the service provided by the lower-layer protocol. ATM protocol models use the OSI concept of SAPs extensively. Stated another way, physical interfaces provide the path for data flow between machines, while protocols manage that data flow across this path using SAPs (or protocol interfaces) between the layers within the machines traversed by PDUs across a network. Obviously, compatibility of both the protocol and physical interfaces is essential in the implementation of communications networks involving machines manufactured by different vendors. Indeed, the original motivation for the OSI Reference Model was to drive toward a multivendor standard that would enable competition with IBM's proprietary Systems Network Architecture (SNA).

The concepts behind the use of multiple protocol layers are important. A communication architecture, also often called a network architecture, is usually viewed as a hierarchy of protocol layers. The division of the communication architecture into layers helps to explicitly identify and isolate functional requirements, support well-defined service interfaces, and allow network designers and administrators to better understand and manage communication systems that employ a number of protocols. Layered architectures are also essential to manufacturers of communication hardware and software. Layers provide them an ability to create modular components within any specific layer and to later alter and improve upon these components without a need to change the other components of different layers. Insofar as the module conforms to the interface design specifications of the adjacent layers, those components can still interoperate. We will now summarize the OSI and other important layered communication architectures, such as TCP/IP, Ethernet, and ISDN.

OPEN SYSTEMS INTERCONNECTION REFERENCE MODEL

The Open Systems Interconnection Reference Model (OSIRM) effort strove to define the functions and protocols necessary for any computer system to connect to any other computer system. The ISO created the OSIRM in the ISO Technical Subcommittee 97 (TC97). Starting in 1977, with Subcommittee 16 (SC16) and officially documenting the protocol architecture in 1983 as ISO standard 7498, the OSIRM provided a framework for organizing various data communications functions occurring between disparate devices working together. This model is used as a guideline for developing standards that can allow the interoperation of equipment produced by various manufacturers. Systems that conform to these standards with interoperability as a goal are referred to as *open systems*. Figure 5-4 depicts the basic OSIRM showing end system (A), intermediate system (B), and end system (C), along with the protocol stack within each. The layers are represented starting from the bottom at layer 1, which has a physical interface to the adjacent node, to the topmost seventh layer, which usually resides on the user end device (workstation) or host that interacts with or contains the user applications. Each of these seven layers represents one or more protocols that define the functional operation of communications between user and network elements. All protocol communications between layers are

“peer-to-peer”—depicted as horizontal arrows between the layers. Standards span all seven layers of the model, as summarized in the text that follows. Although OSI has standardized many of these protocols, only a few are in widespread use. The layering concept, however, has been widely adopted by every major computer and communications standards body and most proprietary implementations as well.

Figure 5-5 illustrates the basic elements common to every layer of the OSI Reference Model. This is the portion of the OSIRM that has become widely used to categorize computer and communications protocols according to characteristics contained in this generic model. Often the correspondence is not exact or one-to-one; for example, ATM and MPLS are often described as embodying characteristics of *both* the data link and network layers. MPLS is often characterized as not fitting the OSIRM very precisely at all, but instead as existing between layer 2 and layer 3.

Referring to Figure 5-5, a layer (N + 1) entity communicates with a peer layer (N + 1) entity by way of a service supported at layer (N) through a Service Access Point (SAP). The layer (N) SAP provides the primitives between layer (N) and (N + 1) of request, indicate, confirm, and response. Parameters are associated with each primitive. Protocol data units (PDUs) are passed down from layer (N + 1) to layer (N) using the request primitive, while PDUs from layer (N) are passed up from layer (N) to layer (N + 1) using the indicate primitive. Control and error information utilizes the confirm and response primitives.

This book utilizes the shorthand notation illustrated in Figure 5-6 to graphically express the concept of protocol layering. This simple syntax represents the PDU structure

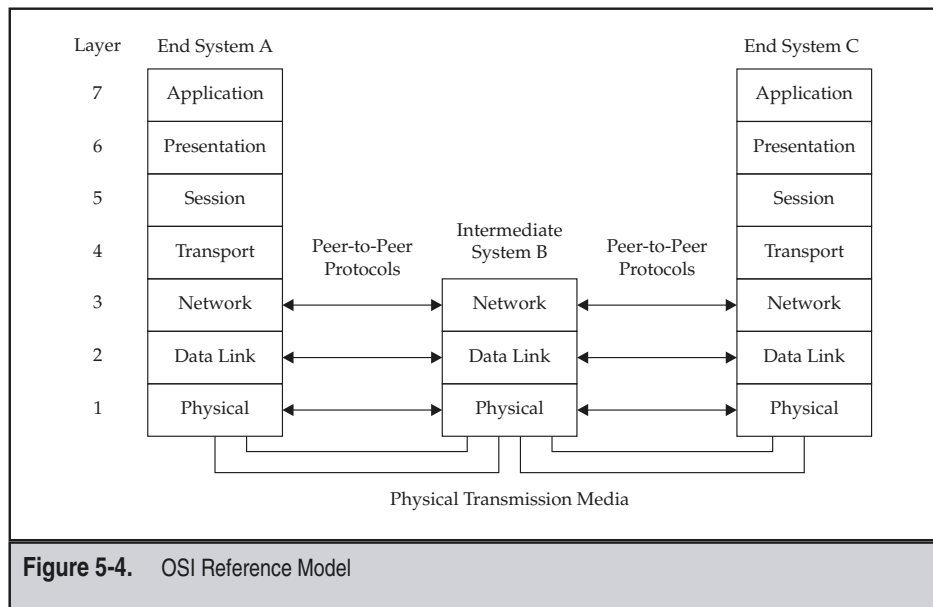


Figure 5-4. OSI Reference Model

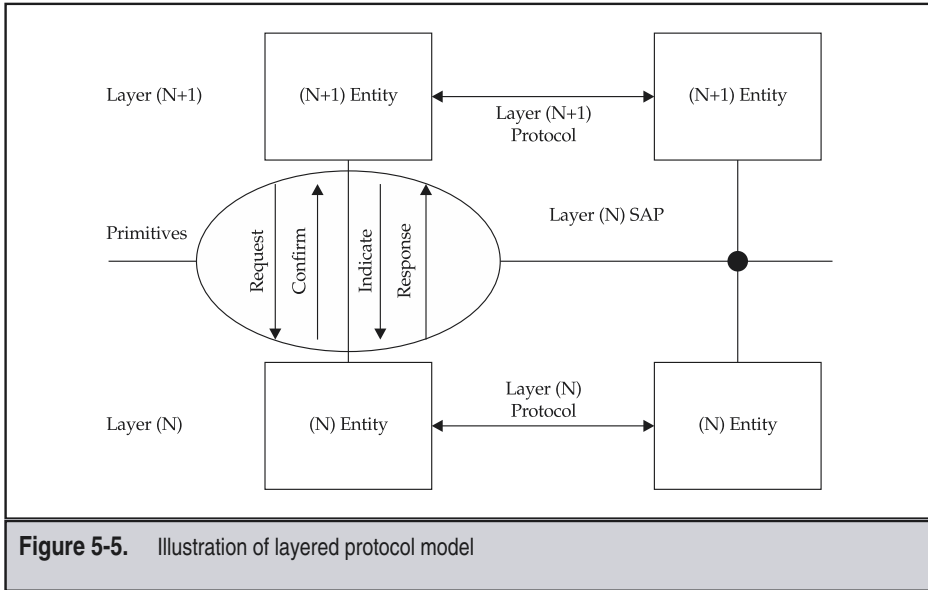


Figure 5-5. Illustration of layered protocol model

passing between layers via stacks that contain only the name or acronym of the protocol. Separate portions of the text define the bit-by-bit protocol data unit (PDU) structures. Starting at the left-hand side, Node A takes data at layer $(N + 1)$, which is connected to Node B by a layer $(N - 1)$ protocol. On the link between Nodes A and B, we illustrate the resultant enveloping of the layer headers (HDR) and trailers (TRLR) that are carried by the layer $(N - 1)$ protocol. Node B performs a transformation from layer (N) to the correspondingly layered, different protocols called layer $(N)'$ and layer $(N-1)'$. The resultant action of these protocol entities is shown by the layer $(N-1)'$ PDU on the link between nodes B and C.

Since the model of Figure 5-6 is somewhat abstract, let's take a closer look at a real-world example to better illustrate the concept. Figure 5-7 illustrates an example of a workstation connected via an IEEE 802.5 Token Ring (TR) LAN to a bridge, which is connected via an 802.3 Ethernet LAN to a server. Both the workstation and the server are using the Internet Protocol (IP) at the network layer. Over the Token Ring physical layer connection, the workstation and the bridge use the Token Ring link layer. Hence, the protocol data unit on the Token Ring LAN begins and ends with a Token Ring header and trailer that envelop an IP header (since IP has no trailer) and also the workstation's data. The bridge takes in the Token Ring link layer and converts this to the IEEE 802.3 Ethernet header and trailer, while also converting this to the 10 Mbps rate running over unshielded twisted pair using the 10 Base T standard. The resulting PDU sent over the wires to the server via the bridge is illustrated in the lower right-hand corner of the figure.

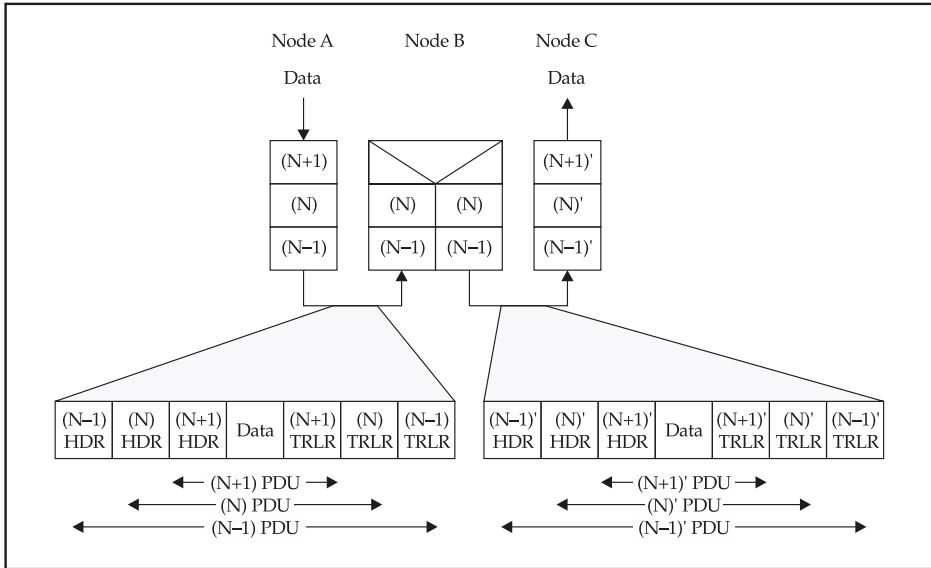


Figure 5-6. Shorthand protocol model notation

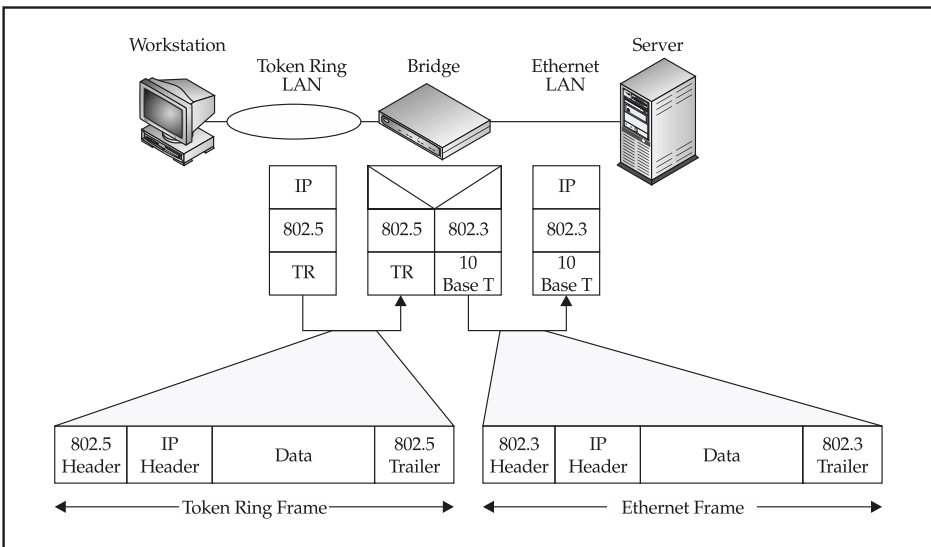


Figure 5-7. Example of layered protocol model and resulting protocol data units

An 802.3 Ethernet header and trailer envelop the same IP header and user data. Communication from the server back to the workstation basically reverses this process. This example serves as a brief introduction only. The curious reader can find more details on the IP protocol in Chapter 8, and on the IEEE 802.3 Ethernet and IEEE 802.5 Token Ring protocols in Chapter 9.

LAYERS OF THE OSI REFERENCE MODEL

We now cover each layer of the OSIRM in more detail. The OSIRM outlines a layered approach to data transmission: seven layers, with each successively higher layer providing a value-added service to the layer above it. Data flows down from layer 7 (application layer) at the originating end system to layer 1 (physical layer), where it is transmitted across a network of intermediate nodes over interconnecting physical medium, and back up to layer 7 of the destination end system. Not all seven levels need be used. The specific OSI protocols for each of the seven layers have not been widely adopted in practice, particularly at the application, presentation, and session layers. The following sections summarize the generic functions of all seven layers, starting with the physical layer, which is the one closest to the physical transmission medium.

Physical Layer

The first layer encountered is the physical layer (L1), which provides for the transparent transmission of a bit stream across the physical connection between network elements. The intelligence managing the data stream and protocols residing above the physical layer is transparently conveyed by the physical layer.

The physical layer connections are either point-to-point or multipoint. The physical layer operates in simplex, half-duplex, or full-duplex mode, as described in Chapter 4. *Simplex* means that transmission is in one direction only. *Half-duplex* involves the use of physical layer signaling to change the direction of simplex transmission to support bidirectional communication, but at any one point in time data flows only in one direction. *Full-duplex* means that transmission occurs in both directions simultaneously. Furthermore, the bit stream may be transmitted serially or in parallel.

The physical layer includes specification of electrical voltages and currents, optical pulse shapes and levels, mechanical connector specifications, basic signaling through connections, and signaling conventions. The physical layer can also activate or deactivate the transmission medium, as well as communicate status through protocol primitives with the layer 2 data link layer. The physical medium can be either an electrical or optical cable, or a satellite or radio transmission channel. Examples of commonly encountered physical layer specifications are EIA-RS-232-C, EIA-RS-449, and the HSSI interface.

The terms *data terminal equipment (DTE)* and *data communications equipment (DCE)* refer to the hardware on either side of a communications channel interface. DTE typically refers to a computer or terminal that acts as an end point for transmitted and received data via a physical interface to a DCE. DCE typically refers to a modem or communications device,

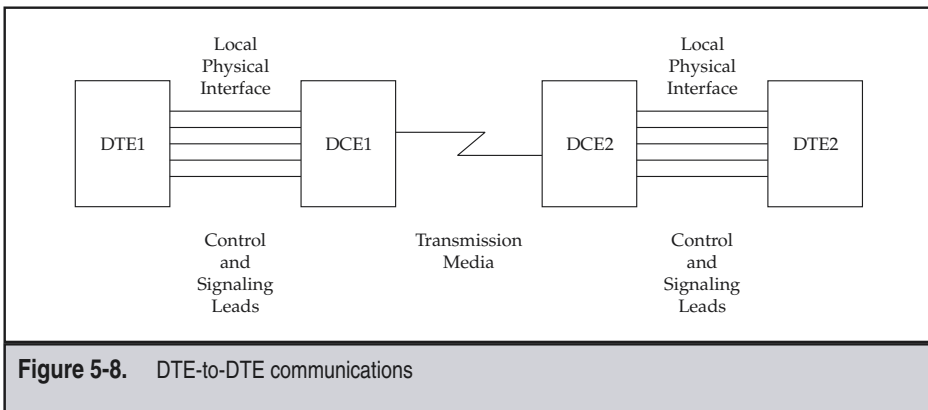
which has a different physical interface than that of the DTE. One commonly used type of DCE is a channel service unit/data service unit (CSU/DSU); it converts the DTE/DCE interface to a telephony-based interface.

Figure 5-8 shows a common end-to-end network configuration where DTE1 talks to DCE1, which, in turn, formats the transmission for transfer over the network to DCE2, which then interfaces to DTE2. Some devices can be configured to act as either a DTE or a DCE.

Data Link Layer

The data link layer is the second layer (L2) in the seven-layer OSIRM, and the second layer in most other computer architecture models as well. The primary function of the data link layer is to establish a reliable protocol interface across the physical layer (L1) on behalf of the network layer (L3). This means that the data link layer performs error detection and, in some cases, error correction. Toward this end, the data link control functions establish a peer-to-peer relationship across each physical link between machines. The data link layer entities exchange clearly delimited protocol data units, which are commonly called *frames*. The data link layer may use a limited form of addressing, such that multiple data link layer protocol interfaces can be multiplexed across a single physical layer interface. There may be a flow control function to control the flow of frames such that a fast sender does not overrun a slow receiver.

Computer communications via local area networks utilize special functions of the data link layer called the Media Access Control (MAC) and Logical Link Control (LLC) layers. The MAC layer protocols form the basis of LAN and MAN standards used by the IEEE 802.X LAN protocol suite introduced later in this chapter, which includes Ethernet, Token Ring, and Token Bus. Examples of data link layer protocol standards include ISO 7776, ISDN LAP-D, ISO HDLC, and MAC-layer protocols such as the ISO 9314-2 FDDI Token Ring MAC.



Some of the new services, such as frame relay, ATM, and MPLS, can be viewed as using only the first two layers of the OSIRM. They rely heavily on reducing the data link layer services to increase speeds at lower costs because of the resulting protocol simplification.

Network Layer

The third layer (L3) encountered in the OSIRM is the network layer. The principal function of the network layer is to provide delivery of protocol data between transport layer entities. In order to do this, the network layer must have an end-to-end, globally unique addressing capability. A unique network layer address is assigned to each network layer protocol entity. A network layer protocol may communicate with its peer over a route of intermediate machines with physical, data link, and network layers. The determination of this route is called the *routing function*. Chapter 9 covers the important subject of routing protocols and their operation. Network layer PDUs are often called *packets*.

The network layer may also perform end-to-end flow control, undertake the segmentation and reassembly of data, and even provide a reliable delivery service. The network layer is the most protocol-intensive portion of packet networks. Some examples of protocols used in the network layer are the ITU X.25 and X.75 packet level and gateway protocols; the Internet Protocol (IP); CCITT/ITU-T Q.931, Q.933, and Q.2931; and the OSI CLNP.

The network layer is also used to define data call establishment procedures for packet- and cell-switched networks in ISDN and B-ISDN. For example, ATM signaling utilizes a layer 3 protocol for call setup and disconnection. SMDS also employs a layer 3 protocol to provide an end-to-end datagram service using E.164 (i.e., telephone numbers) for addressing. We cover each of these concepts in subsequent chapters. Since MPLS interacts closely with IP routing, there is an analogous relationship between establishment of an MPLS label switched path (LSP) and layer 3 control signaling and routing.

Transport Layer

The fourth layer (L4) encountered is the transport layer. The principal function of the transport layer is to interconnect session layer entities. Historically, it was also called the host-to-host layer. Principal functions that it performs are segmentation, reassembly, and multiplexing over a single network layer interface. The transport layer allows a session layer entity to request a class of service, which must be mapped onto appropriate network layer capabilities. Frequently, the transport layer manages end-to-end flow control. The transport layer may often perform error detection and correction as well. This has become increasingly important because it provides a higher-level error correction and retransmission protocol for services that usually don't provide reliable delivery, such as frame relay, IP, ATM, and MPLS. Often, frame relay users ask what happens when frames are lost. The answer is that the transport layer retransmits these lost packets.

One example of the transport layer is the ITU X.224 OSI transport protocol TP4. Another widely used example of a transport type of protocol is the Internet Transmission Control Protocol (TCP).

Session Layer

The fifth layer (L5) encountered is the session layer. The session layer is essentially the user's interface to the network, which may have some data transformations performed by the presentation layer. Sessions usually provide connections between a user, such as a terminal or LAN workstation, and a central processor or host. So-called peer-to-peer session layer protocols can directly connect user applications. Session layer protocols are usually rather complex, involving negotiation of parameters and exchange of information about the end user applications. The session layer employs addresses or names that are meaningful to end users. Other session layer functions include flow control, dialog management, control over the direction of data transfer, and transaction support.

Some examples of the session layer are terminal-to-mainframe logon procedures, transfer of user information, and the setup of information and resource allocations. The ISO standard for the session layer is the ISO 8327/ITU X.225 connection-oriented session protocol.

Presentation Layer

The sixth layer (L6) is the presentation layer, which determines how data is presented to the user. Official standards are now complete for this layer. Many vendors have also implemented proprietary solutions. One reason for these proprietary solutions is that the use of the presentation layer is predominantly equipment dependent. Some examples of presentation layer protocols are video and text display formats, data code conversion between software programs, and peripheral management and control, using protocols such as ITU X.410 and ITU X.226.

Application Layer

The seventh and final layer (L7) is the application layer. This layer manages the program or device generating the data to the network. More important, this layer provides the actual interface to the end user. The application layer is an "equipment-dependent" protocol and lends itself to proprietary vendor interpretation. Examples of standardized application layer protocols include ITU X.400, X.420, and X.500–X.520 directory management, ISO 8613/ITU T.411–419 Office Document Architecture (ODA), and ISO 10026 distributed transaction processing (TP).

Mapping of Generic Devices to OSI Layers

In the mainframe and minicomputer era, the bottom three layers (network, data link, and physical) were implemented on different pieces of equipment than the next three higher layers (presentation, session, and transport). The first three layers were implemented on a front-end processor (FEP), while the higher four layers were implemented on a host. Current customer premises devices such as bridges, routers, and hubs usually manipulate the protocols of the first three layers: network, data link, and physical. They can often connect dissimilar protocols and interfaces. Many implementations of user software that cover the top three nonapplication layers (presentation, session, and transport) operate as a single program.

LAYERED DATA COMMUNICATION ARCHITECTURES

In addition to the OSIRM, several other data communication protocol architectures shaped and standardized the computer networking industry in the late twentieth century. These include the Internet Protocol (IP) suite, IBM's SNA, the IEEE 802.X LAN standards, and the ITU-T's ISDN. This section introduces these important architectures.

Internet Protocol (IP) Architecture

The Internet Protocol uses fewer layers than the OSI Reference Model, as seen from Figure 5-9. An Internet is composed of end systems (synonymously called *hosts*) interconnected by routers. The hosts run networked end-user applications over a small set of transport protocol services, such as the Hypertext Transfer Protocol (HTTP) commonly used on the World Wide Web. The transport layer interfaces to the Internet layer, which runs over a comprehensive range of data link layer protocols. Intermediate routers provide a connectionless datagram forwarding service to the transport layer. As we describe in the next section, connectionless forwarding is a simple, powerful concept that has proved to be extremely useful and scalable in data networking.

Routers do not perform any transport layer functions—as indicated in the figure. Unlike the OSI layered model, the IP architecture does not require a reliable data link or network layer. Instead, the important function of error detection and retransmission is performed by the Transmission Control Protocol (TCP) at the transport layer. Possibly because of inherent simplicity and elegance, and partly because of the World Wide Web's emergence as the dominant force in internetworking, the IP protocol suite has become the de facto worldwide standard for most end-user computer equipment networking. Chapter 8 covers the Internet Protocol suite in detail.

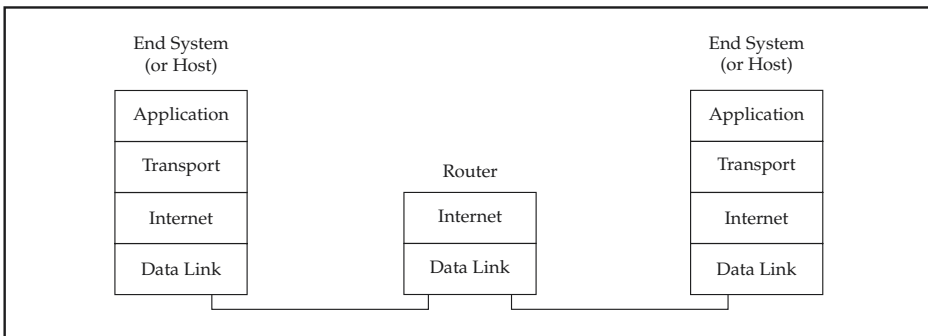


Figure 5-9. Internet Protocol (IP) layered architecture

IBM's Systems Network Architecture (SNA)

The introduction by IBM of Systems Network Architecture (SNA) in 1974 signaled the beginning of a vendor-proprietary architecture that remained prominent in the computing industry into the 1990s. SNA architectures and protocols are still widely used by many businesses and institutions. Many users have multiprotocol environments where designers either separate networks for their IP/IPX traffic and SNA traffic or encapsulate (or tunnel) their SNA traffic inside another network protocol (such as frame relay). This technique is not without problems. However, since SNA protocol timers assume a private line interconnection environment, the variable delay sometimes encountered with tunneling over another protocol may cause timeouts resulting in session losses. A common solution to this problem is protocol spoofing. This technique involves the device attached to the SNA network sending acknowledgments to the timeout-sensitive SNA device prior to receiving the distant acknowledgment. SNA was IBM's method of creating a computing empire through standardization that centered on the mainframe and (distributed) front-end processors. The problem that the huge IBM corporation faced in standardizing data communications among its own products was formidable. By providing a hierarchy of network-access methods, IBM created a network that accommodated a wide variety of users, protocols, and applications, while retaining ultimate control at the mainframe host and front-end processors. The move from centralized to distributed processing has had pronounced effects on SNA. IBM dubbed its latest evolution of SNA Advanced Peer-to-Peer Networking (APPN) using Advanced Program-to-Program Communication (APPC) in an attempt to preserve homogeneous SNA networks. However, many SNA users now look to less elegant (and less expensive)—yet functional—solutions.

The SNA architecture layers are shown in Figure 5-10, where an SNA terminal is connected via a front-end processor (FEP) to a mainframe computer [Cypser 78]. Since SNA preceded the OSIRM, most of the names for the layers differ, except for the physical and data link layers. The SNA layered stack is divided into two main components: node-by-node transmission services and end-to-end services. Every node implements the transmission services, while only end systems implement the end-to-end services as half-sessions, as shown in the figure. The common transmission services encompass the physical, data link, and path control layers. The physical and data link layers define functions similar to the OSIRM, with serial data links employing the SDLC protocol and channel attachments between front-end processors (FEPs) and mainframes employing the System 370 protocol. The path control layer provides connectivity between half sessions based upon the addresses of the source and destination network accessible units (NAUs). Hence, a key function of path control is to determine the node-by-node route from the source to the destination. SNA employs a hierarchical structure to enable scaling to large networks by grouping nodes into subareas. Path control utilizes precomputed explicit routes to route packets between sessions. Since the philosophy in SNA is per-session flow control, the path control layer basically provides only an indication back to higher layers about network congestion.

Fundamental to SNA is the concept of a *session*, where each system in an end-to-end connection implements half of the protocol. The transmission control layer establishes,

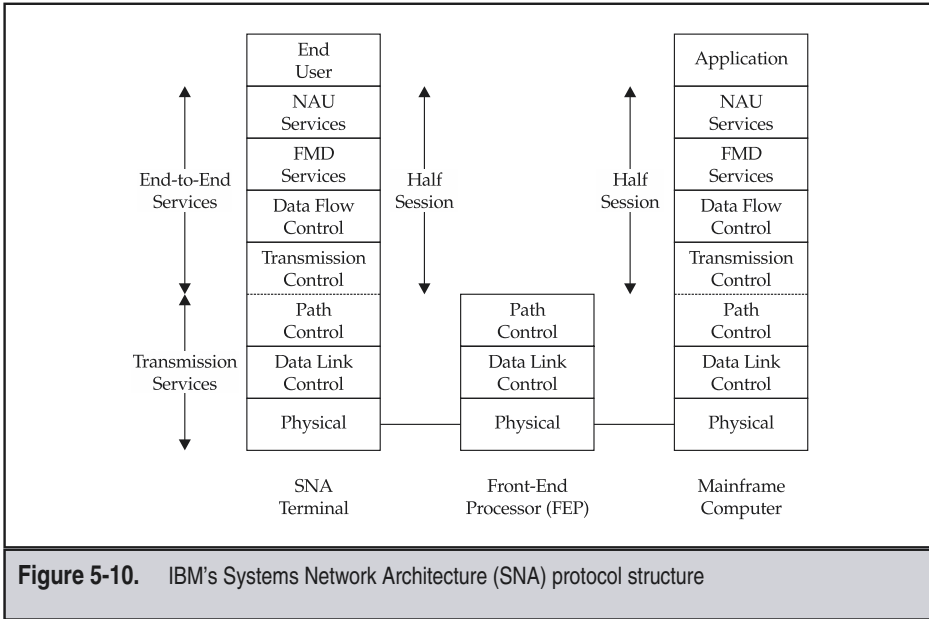


Figure 5-10. IBM's Systems Network Architecture (SNA) protocol structure

maintains, and terminates sessions between logical units (LUs) of various types. The transmission control layer also performs some routing functions, for example, within a mainframe where multiple simultaneous sessions exist. The data flow control layer involves the processing of request-response units exchanged over sessions. It provides functions controlling priority, grouping packets into larger logical sets (such as all the lines on a screen), and controlling operation over half-duplex links. Moving up the protocol stack, function management data (FMD) services and network accessible unit (NAU) services complete the SNA protocol stack. FMD services provide user-to-user as well as presentation services, including data compression and character set conversion. NAU services cover the concepts of system services control points (SSCPs), physical units (PU), and logical units (LUs). The NAU services are the interface to the end user and applications.

IEEE 802.X Series (LAN/MAN/WAN)

The Institute of Electrical and Electronics Engineers (IEEE) established the 802 working group to standardize local and metropolitan area networks (LANs and MANs). These standards have become so important that the ISO, the International Electrotechnical Commission (IEC), and the American National Standards Institute (ANSI) also publish these standards. Generically called the IEEE 802.X series of standards, these important specifications cover the physical and data link layers shown in Figure 5-11. The physical specification, commonly abbreviated as PHY, covers the interface to a variety of physical media, such as

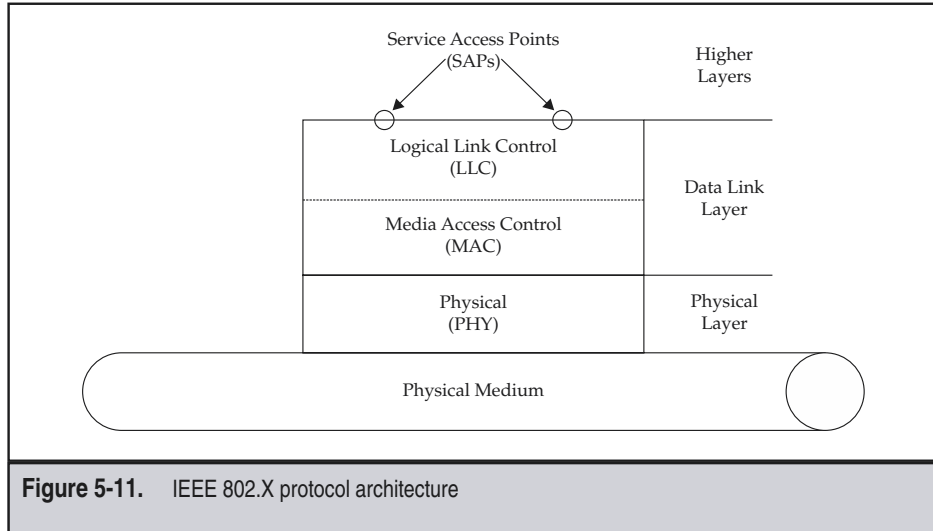


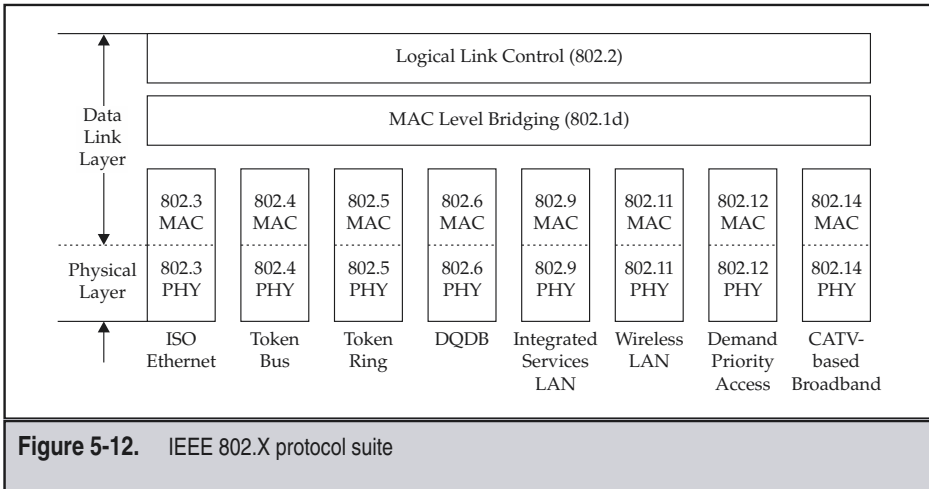
Figure 5-11. IEEE 802.X protocol architecture

twisted pair, coax, fiber, and radio frequencies. Above the PHY layer, the data link layer embodies two sublayers: a Media Access Control (MAC) sublayer, and a Logical Link Control (LLC) sublayer. The LLC layer provides one or more Service Access Points (SAPs) to higher-layer protocols, as indicated in Figure 5-11. A commonly encountered example of a SAP is the LAN driver in personal computer software. We cover this important series of standards in greater detail in Chapter 9.

The LLC layer operates the same way for all LAN architectures, but not for the 802.6 MAN architecture, which is a completely different beast, as discussed in Chapter 8. The MAC layer and physical layer operate differently for each of the local and metropolitan area network architectures. The LLC layer defines common procedures for call establishment, data transfer, and call termination through three types of services: connection-oriented, unacknowledged connectionless, and acknowledged connection-oriented.

Figure 5-12 depicts the IEEE 802.X protocol suite. The IEEE 802.2 standard [ISO 8802.2] defines the LLC sublayer. Also note that the 802.1d standard defines LAN bridging at the MAC level, a topic covered in depth in Chapter 9 as an introduction to LAN emulation (LANE) over ATM. Two major LAN architectures defined in the 802.X standards are commonly used: Ethernet and Token Ring, also defined in Chapter 9. Ethernet is by far the most commonly used LAN protocol. One major MAN protocol defined by the IEEE was implemented: the Distributed Queue Dual Bus (DQDB), as defined in IEEE 802.6. This protocol is used by the Switched Multimegabit Data Service (SMDS).

The IEEE 802.3 standard and other specifications form what is called the Ethernet standard. The first Ethernet products appeared in 1981, and now sales for Ethernet outpace all



other 802 protocols combined. Ethernet users contend for a shared medium after first sensing for a carrier transmission by other users. The original interface specified 10 Mbps over twisted pair or coaxial cable physical media. Now, fast Ethernet speeds are available at 100 Mbps, Gigabit Ethernet at 1 Gbps, as well as 10 Gbps Ethernet.

IBM invented the Token Ring architecture in its development labs in Zurich, Switzerland. The first Token Ring products appeared in 1986. The IEEE 802.5 standard defines the protocol that involves passing a “token” between stations to control access to the shared medium. Token Ring initially competed with Ethernet as a popular LAN standard but gave up significant market share to the economical Ethernet alternative in the 1990s. Today most estimates place Token Ring at less than 15 percent of LANs in use, compared with Ethernet use exceeding 80 percent.

Integrated Services Digital Network Protocol Architecture

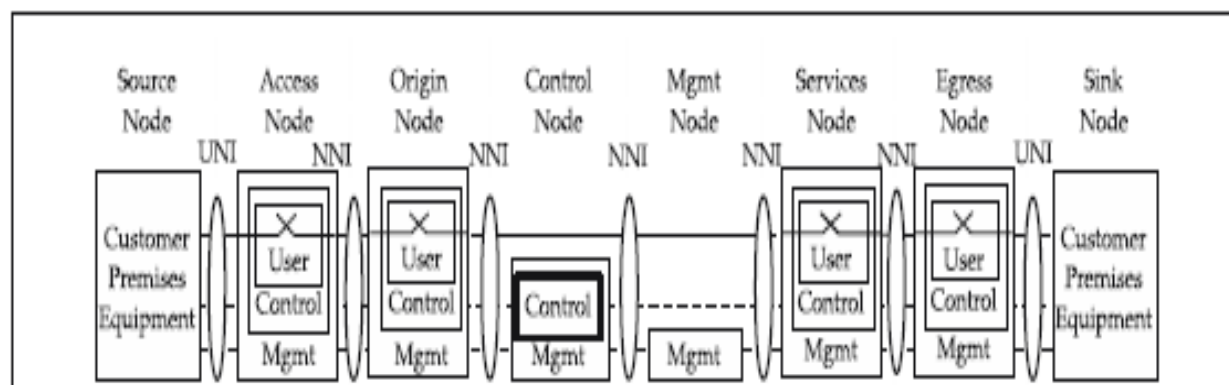
The ITU began work on the Integrated Services Digital Network (ISDN) standards in 1972, with the first documents published in 1984. ISDN’s initial goal was aimed at converting the entire telecommunications transmission and switching architecture to a digital architecture, providing end user-to-end user digital service for voice, data, and video over a single physical access circuit. Narrowband ISDN (N-ISDN) standards are the root of Broadband ISDN (B-ISDN) standards, of which ATM is the key protocol layer, as described in Chapter 10.

The basic concept of ISDN is of multiple types of terminal equipment (TE), such as phones and computers, connecting through an ISDN network termination point (called an NT) into the central office environment that provides access to a range of information. While all seven protocol layers in N-ISDN are the same as the OSIRM, the physical, data

link, and network layers define the lower-layer functions, also called bearer services. These layers define physical connectivity and transmission as defined in ITU-T Recommendations I.430, I.431, and I.432; data link management, flow, error, and synchronization control as defined in ITU-T Q.921(LAP-D); and network addressing, congestion control, end-to-end call establishment, routing or relaying, and switching as defined in Recommendations Q.931/I.451, Q.932/I.452, and Q.933/I.453. The transport, session, presentation, and application layers define the higher-layer functions, including the teleservices that define services such as messaging, telephone, and telex. Standards for these layers are host-to-host, as well as application-specific.

The ISDN architecture was the first to introduce the concepts of multiple-layered protocol planes. Figure 5-13 illustrates the user, control, and management planes of ISDN. Note how addition of this dimension to the overall protocol architecture provides for specialization of particular nodes, as well as operation at multiple protocol layers simultaneously. The ITU-T applied this same multiple plane model to the B-ISDN specifications, as covered in the next part.

The user protocol (or bearer service) is layer 1 for circuit-mode (shown in Figure 5-13), layer 2 for frame-mode, and layer 3 for packet-mode services. Teleservices and value-added services operate at higher layers. Intermediate nodes may provide only physical



connectivity. User-to-Network Interfaces (UNI) and Network-to-Network Interfaces (NNI) will be explained later. Conceptually, another application runs the control, or signaling, plane. The purpose of the control plane protocols is to establish, configure, and release the user plane (bearer) capabilities. Finally, the management plane is responsible for monitoring the status, configuring the parameters, and measuring the performance of the user and control planes. We cover the protocol structure and functional building blocks of ISDN in the next chapter.

NETWORK SERVICE PARADIGMS

The OSI Reference Model categorizes data network services by one of two paradigms: connection-oriented or connectionless. Connection-oriented network services (CONS) involve establishing a connection between physical or logical end points *prior* to the transfer of data. Examples of CONS are frame relay, TCP, ATM, and MPLS. Connectionless network services (CLNS), on the other hand, provide end-to-end logical connectivity *without* establishing any connection before data transfer. Examples of CLNS are the IP and LAN protocols. As we shall see repeatedly, the consequences of ATM and MPLS being connection oriented, whereas IP and LAN protocols are connectionless, are far-reaching indeed. Historically, wide area networks employed connection-oriented services, while local area networks used connectionless services. ATM, along with its supporting cast of adaptation layers and control protocols, strove to support *both* connection-oriented and connectionless services; while, on the other hand, as originally conceived, MPLS plays a subservient connection-oriented role within the overall IP architecture.

Connection-Oriented Network Service (CONS)

Connection-oriented services require establishment of a connection between the origin and destination before transferring data. The connection is established as a single path of one or multiple links through intermediate nodes in a network. Once established, all data travels over the same preestablished path through the network. The fact that data arrives at the destination in the same order as sent by the origin is fundamental to connection-oriented services.

If network management or provisioning actions establish the connection and leave it up indefinitely, then we call the result a Permanent Virtual Connection (or circuit) (PVC). If control signaling of any type dynamically establishes and takes down the connection, then it is called a Switched Virtual Circuit (or connection) (SVC). X.25, frame relay, and ATM all use the notion of PVC and SVC; and in different documents for the same protocol, the “C” in the acronym is spelled out as either “circuit” or “connection.” Although different words are used, the meaning is the same.

A PVC connection may be established by physical wiring, equipment configuration commands, service provider provisioning procedures, or combinations of these actions. These actions may take several minutes to several weeks, depending upon exactly what is required. Once the PVC is established, data may be transferred over it. Usually PVCs are

established for long periods of time. Examples of physical PVCs are analog private lines, DTE-to-DCE connections, and digital private lines. Examples of logical PVCs are the X.25 PVC, the frame relay PVC, the ATM PVC, and an MPLS label switched path (LSP).

In the case of an SVC service, only the access line and address for the origin and each destination point are provisioned beforehand. The use of a control signaling protocol plays a central role in SVC services. Via the signaling protocol, the origin requests that the network make a connection to a destination. The network determines the physical (and logical) location of the destination and attempts to establish the connection through intermediate node(s) to the destination. The success or failure of the attempt is indicated back to the originator. There may also be a progress indication to the originator, alerting for the destination, or other handshaking elements of the signaling protocol as well. Often the destination utilizes signaling to either accept or reject the call. In the case of a failed attempt, the signaling protocol usually informs the originator of the reason that the attempt failed. Once the connection is established, data can then be transferred. Networks may employ SVCs to efficiently share resources by providing dynamic connections and disconnects in response to signaling protocol instructions generated by end users. End users could use SVCs as a way to dynamically allocate expensive bandwidth resources without a prior reservation. In real-world ATM and MPLS networks, a semipermanent virtual circuit (SPVC) is initiated by network management system or operator command and uses SVC signaling to set up a PVC. Although extensive standards are in place to support end-user-initiated SVC capabilities, such services have not been successfully offered by most ATM service providers, and practically all ATM services today are established with PVC, and SPVC, connections. We will examine some of the limitations that may have contributed to the slow acceptance of SVC services when we discuss the ATM and MPLS protocols in Part 3.

Probably, the simplest way to explain an SVC is to compare it to a traditional telephone call. After ordering the service and receiving address assignments, the communications device “picks up the phone” and “requests” a connection to a destination address. The network either establishes the call or rejects it with a busy signal. After call establishment, the connected devices send data until one of the parties takes the call down. There is a direct analogy between establishing and taking down an SVC connection-oriented service and a normal telephone call, as illustrated in Table 5-1.

Connectionless Network Services (CLNS)

As the name implies, connectionless services never establish connections of any kind. Instead, network nodes examine the address field in every packet header to determine the destination. Network nodes provide connectionless service by forwarding packets along a path toward the destination. Each node selects an outgoing link on a hop-by-hop basis. Typically, the nodes run a distributed routing protocol that consistently determines the forwarding tables to result in optimized, loop-free, end-to-end paths. Therefore, unlike in a connection-oriented service, packets do not take a predetermined path through the network. Thus, connectionless services avoid the overhead of call establishment and management incurred by connection-oriented services. The origin node initiates the forwarding process, with each intermediate node repeating it until the packet reaches the

General Signaling Protocol	Voice Telephone Call
Provision access/address	Order service from phone company
Handshaking	Obtain dial tone
Origin request	Dial the destination number
Successful attempt indication	Ringing tone
Unsuccessful attempt indication	Busy tone
Destination acceptance	Answering the phone
Data transfer	Talking on the phone
Disconnect request	Hanging up the phone

Table 5-1. Comparison of General Signaling Terminology to a Telephone Call

destination node. The destination node then delivers the packet to its local interface. Pretty simple, right?

Yes, and no. The magic in the preceding simple description is the routing protocol that consistently determines the next hop at the origin and each intermediate node. Chapter 9 explains more details about different types of routing protocols, but they all achieve the same purpose stated in the previous sentence: routing protocols determine the contents of the next-hop forwarding table such that packets with the same destination address take the same path through the network. A bad routing protocol could create next-hop entries that cause endless loops where a packet never arrives at the destination but instead loops around the network indefinitely. On the other hand, a good routing protocol automatically chooses a path through the network optimized to a specific criterion, such as minimum cost. Note that if the routing protocol changes the next-hop forwarding table in the middle of data transfer between end systems (for example, if a physical circuit fails), then packets may arrive at the destination in a different order than sent by the origin.

Chapter 9 details common network node implementations of connectionless services, namely, bridges and routers. As we shall see, some aspects of connectionless services are truly plug and play, while others require address configuration, subnet mask definitions, and setting of other parameters. The connectionless paradigm requires that each network node (e.g., a router) process each packet independently. Common per-packet processing functions required in real networks include filtering (out) certain packets according to address and other fields, queuing different packet flows for prioritized service, and data link layer conversions. Older routers implemented this complex processing in software, which limits throughput. Practically, using filtering to implement a firewall limits router throughput significantly. However, hardware-based routers opened up this bottleneck in the late 1990s.

Connectionless services do not guarantee packet delivery; therefore, applications rely on higher-level protocols (e.g., TCP) to perform the end-to-end error detection/correction. Additionally, higher-layer protocols must also perform flow control (e.g., TCP) or admission control (e.g., RSVP), since the IP connectionless service typically operates on a best-effort basis without any notion of bandwidth allocation. As we discuss in Part 3, the use of protocols like RSVP to reserve bandwidth is a critical component of MPLS.

Connection-Oriented Versus Connectionless Services Analogy

One simple analogy for understanding the difference between CONS and CLNS is that of placing a telephone call compared with sending a telegraph message. To make a phone call, you pick up the phone and dial the number of the destination telephone. The network makes a connection from your house, through one or more telephone switches, to the destination switch and rings the phone. Once the called party answers, the telephone network keeps the connection active until one of the parties hangs up.

Now here is a CLNS example. Consider sending a telegraph message in the nineteenth century. A person visits the telegraph office and recites a message, giving the destination address as a city and country. The telegraph operator picks a next-hop telegraph station and keys in the entire message to that telegraph office. Since the originating telegraph operator does not know the status of telegraph lines being up or down except for those lines connected to his own station, he must rely on the other operators to forward the message toward the destination. If there is a path to the destination, then the persistent telegraph operators in this example eventually relay the message to the final destination, even if some telegraph lines on the most direct path are down. This example is not as dated as it may seem—Internet e-mail systems use essentially the same method proven over a century ago by telegraph networks to reliably forward and deliver messages.

REVIEW

This chapter began with an introduction to packet switching and frame relaying by exploring: their reasons for creation and basic principles, and their history. This chapter then introduced the concept of protocol layering and the notation used throughout the book. Next, a brief description covered the Open Systems Interconnection Reference Model (OSIRM) and its seven layers: physical, data link, network, transport, session, presentation, and application. The text also summarized the TCP/IP, IBM SNA, IEEE, and N-ISDN architectures, giving examples of standards and real-world equipment implementations. Finally, the chapter concluded with definitions of connection-oriented and connectionless network services, giving some foreshadowing as to the manner in which ATM and MPLS serves both requirements.

CHAPTER 6

Time Division Multiplexing and the Narrowband Integrated Services Digital Network

This chapter begins with an overview of circuit-switched network services, the predecessor to the signaling and connection control aspects of the Narrowband Integrated Services Digital Network (N-ISDN) control plane. This chapter then introduces the basics of time division multiplexing (TDM) used in the North American, European, and Japanese plesiochronous and synchronous digital transmission hierarchies. Begun as a means to more economically transport voice signals, TDM forms the underlying fabric of most wide area communications networks today. We then examine the services provided using the TDM networking paradigm in the N-ISDN.

CIRCUIT SWITCHING

The following section reviews the long history of circuit switching and how it continues to exert a strong influence on the design of modern communications networks.

History of Circuit Switching

Circuit switching originated in the public telephone network. The first telephone networks had a dedicated electrical circuit from each person to every other person that desired communication. This type of connectivity makes sense if you talk to very few people and very few people talk to you. Today, the typical person makes calls to hundreds of different destinations for friends and family, business or pleasure. It is unrealistic to think that in this environment each of these call origination and destination points would have its own dedicated circuit to all others, since it would be much too expensive and difficult to administer.

Historically, early telephone networks dedicated a circuit to each pair of callers until the maze of wires overhead on telephone poles began to block out the sun in urban areas. The next step toward switching was human telephone operators who manually connected parties wishing to communicate using patch cords on a switchboard. Callers identified the called party by telling the operator the name of the person to whom they wished to speak. This design relieved the problem greatly, as all the wires from each user went back to a central operator station instead of to every other user. However, once the number of users grew beyond what a single operator could handle, multiple operators had to communicate in order to route the call through several manually connected switchboards to the final destination. Interestingly, the reason Almon B. Strowger invented the first electromechanical circuit switch in 1889 [Bear 76] had nothing to do with engineering efficiency, but everything to do with basic capitalism. As the story goes, Strowger was actually an undertaker by trade in a moderate-sized town that had two undertakers. Unfortunately for Strowger, his competitor's wife was the switchboard operator for the town. As the telephone increased in popularity, when anyone died, their relatives called the telephone operator to request funeral services. The operator in this town routed the requests to her husband, of course, and not to Strowger. Seeing his business falling off

dramatically, Strowger conceived of the electromechanical telephone switch and the rotary dial telephone so that customers could contact him directly. As a result, Strowger ended up in an entirely different, but highly successful business. Now, we take for granted the ease of picking up the phone virtually anywhere in the world and dialing any other person in the world.

Digitized Voice Transmission and Switching

Bell Labs engineers faced a decision in the 1950s of either augmenting bundles of twisted pairs run in conduits under the streets in large metropolitan areas or multiplexing more voice conversations onto the existing bundles using a new digital technique. The high cost of adding conduits with more twisted pairs drove them to deploy a radically new entirely digital technique. It converted the analog voice band signals to digital information prior to transmission, as illustrated in Figure 6-1. Nyquist derived a theorem in 1924 proving that the digital samples of an analog signal must be taken at a rate no less than *twice* the bandwidth of that signal to enable accurate reproduction of the original analog signal at the receiver. Thirty years later, telephone engineers put the Nyquist sampling theorem into practice by sampling a standard 4000 Hz bandwidth voice channel at 8000 samples per second. Employing 8 (or 7) bits per sample yields the standard 64 Kbps (or 56 Kbps) digital data stream used in modern digital (TDM) transmission and switching systems for each voice channel. Engineers call the digitized coding of each analog voice sample pulse code modulation (PCM). The numerical encoding of each PCM sample uses a nonlinear companding (*compression/expanding*) scheme to improve the signal-to-noise ratio by providing greater granularity for larger amplitude values. Unfortunately, the methods used for representing PCM samples differ in networks around the world, with the μ -Law (pronounced “mew-law”) standard used in North America and an A-Law method used elsewhere. In fact, while many networks still transmit voice at 56 or 64 Kbps, more sophisticated modern digital encoding techniques now enable transmission of a voice channel at speeds as low as 5.3 Kbps (with some loss in quality, of course) when bandwidth is expensive or scarce, as described in Chapter 16. International networks and voice over limited bandwidth applications are the primary applications for these low-bit-rate techniques.

Transmission systems multiplex 24 such digitized and sampled voice channels (called a digital signal 0 (DS0) in North America) onto a single twisted pair using a T1 repeater signal according to a digital signal 1 (DS1) signal format. The DS1 transmission rate of 1.544 Mbps derives from multiplying the DS0 rate of 64 Kbps by 24 (i.e., 1.536 Mbps), plus an 8 Kbps framing and signaling channel derived from the framing bits interleaved with the DS0 streams. We review the DS1 signal format later in this chapter, since it forms the basis of the narrowband ISDN primary rate interface. This design decision resulted in an improvement of over 2400 percent in utilization of the scarce twisted pair resource—a tremendous gain in efficiency! International standards adopted a similar multiplexing technique to make better use of existing twisted pair plant but multiplexed together 32 64 Kbps channels instead of 24 in a standard called E1 operating at 2.048 Mbps. We also cover this format later in the chapter in the section on N-ISDN.

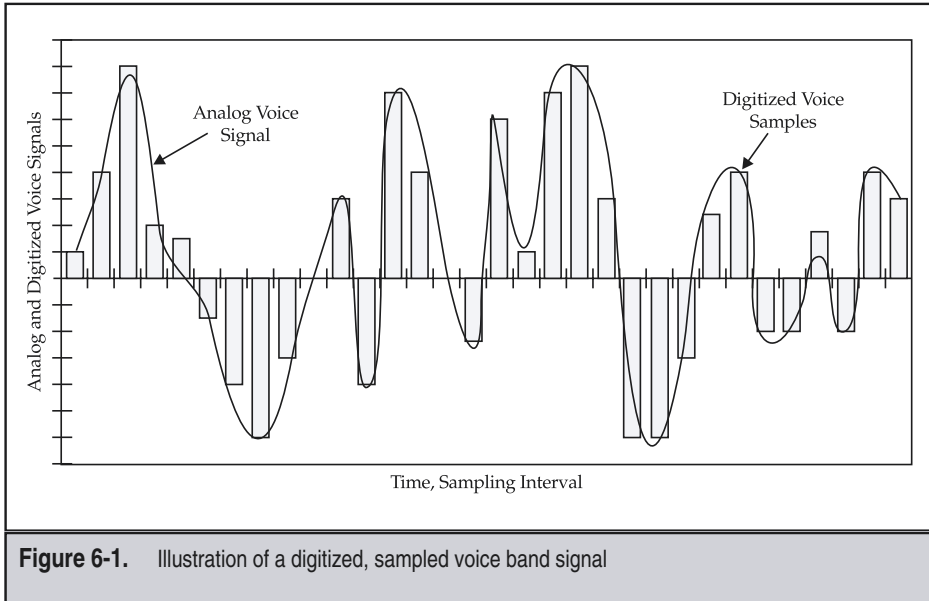


Figure 6-1. Illustration of a digitized, sampled voice band signal

Digital Data Circuit Switching

If the access line from the customer is digital instead of analog and the network has digital switches, then another set of digital data circuit-switching services apply that eliminate the need for analog-to-digital conversion in a modem. Of course, data circuit switching arrived only after carriers replaced older analog telephone switches with updated, entirely digital versions. Now, the availability of digital data service is limited primarily to whether the user's access line is digital or analog. Since TDM uses 8000 samples per second per DS0 channel, a difference arises from the fact that 56 Kbps uses only 7 bits per sample, while 64 Kbps uses all 8. The 56 Kbps rate resulted from the historical use by the North American telephone network of 1 bit per sample for robbed-bit signaling.

Switched 56 Kbps, or simply switched 56, is a service offered in both the private and public networking environments. Often, a channel service unit/data service unit (CSU/DSU) device attaches via a dedicated digital access line to a carrier's switched 56 Kbps service. The DSU side presents a standard DCE interface to the computer equipment, as described in Chapter 4. Users employ data circuit switching as a backup for private-line services or for on-demand applications. The price of circuit-switched data services is close to that of voice service, since that is basically what it is! This pricing makes it a cost-effective option to leased-line services if usage is less than several hours per day, or if multiple destinations require dynamic connectivity. The data communications user, however, needs up to three logical types of communication for one call: the data circuit; a signaling capability; and, optionally, a management capability.

Many carriers now offer data circuit-switched services ranging in speeds from 56/64 Kbps up to 1.5 Mbps, including nx56/64 Kbps. Applications that use high-speed circuit switching as an ideal solution are ones such as bulk data transport and/or those that require all the available bandwidth at predetermined time periods. Circuit switching can provide cost reductions and improve the quality of service in contrast to dedicated private lines, depending upon application characteristics.

The interface for switched services can be directly from the CPE to the interexchange carrier (IXC) point of presence (POP), or via a local exchange carrier's (LEC's) switched data service, as depicted in Figure 6-2. Dedicated access lines connect the customer equipment to the LEC and IXC carrier services. A trunk group of many circuits connects the LEC and the IXC, achieving an economy of scale by sharing these circuits between many users. Many of these trunk groups carry both digitized voice and data calls. A common use of this configuration involves use of the LEC switched service as a backup to the dedicated access line to the IXC circuit-switched data service.

Many users implement circuit-switched data services as a backup for private lines as shown in Figure 6-2, or else for occasional, on-demand transfer of data between sites. Some carriers also offer noncontiguous and contiguous fractional DS1 or nxDS0 reconfigurable or switched services. Reconfigurable services often utilize a computer terminal to rearrange digital cross-connects to provide a version of nxDS0 switching. Depending upon whether the control system utilizes semiautomated network management or signaling-based control, the circuit establishment times in these services range from seconds to minutes. The N-ISDN-based version of this service is called the Multirate Circuit-Mode Bearer Service (MRCMBS), which supports switched nxDS0. Videoconferencing is an example application that employs MRCMBS to combine multiple 56/64 Kbps circuits to form a single high-speed videoconference channel at higher speeds. Some examples of switched DS1 service traffic include video, imaging, large file transfers, and data center disaster recovery.

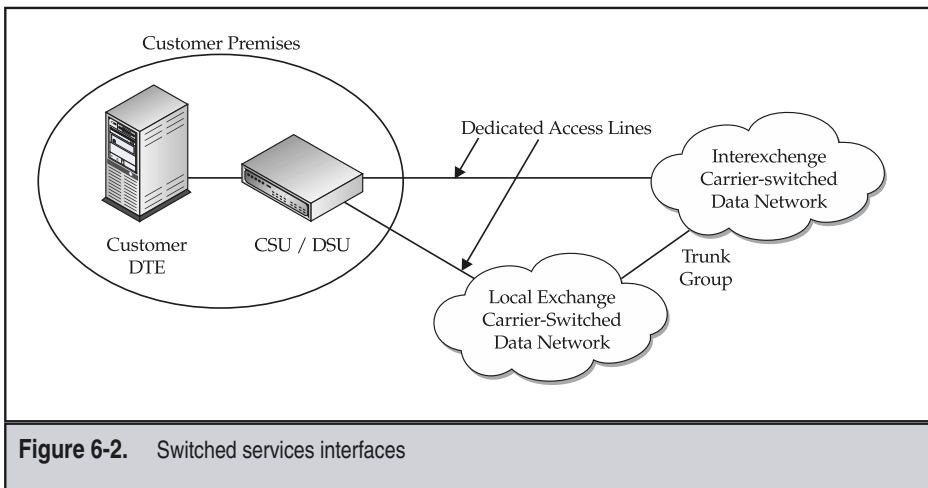


Figure 6-2. Switched services interfaces

PRIVATE-LINE NETWORKS

Users have three physical layer networking options: private-line networks, switched networks, and hybrid designs incorporating a mix of both. This section covers some practical aspects of private-line networking and concludes with a comparison of private-line and circuit-switched networks.

Private (Leased)–Line Characteristics

Private lines are the simplest form of point-to-point communications. Private lines, also called leased lines, are dedicated circuits between two user locations. Since a service provider dedicates bandwidth to a private line connecting ports on its network, the customer pays a fixed monthly fee dependent upon the distance traversed and the bit rate ordered. Usually, private-line tariffs also have a nonrecurring installation fee. In return, the service provider guarantees the private-line bandwidth effectively 24 hours a day, 7 days a week.

Leased lines come in several grades and speeds. The most basic traditional service available consists of either analog or digital leased lines. Carriers offer digital leased lines at speeds such as the 9600 bps, 19.2 Kbps, 56/64 Kbps, fractional T1, T1 (1.544 Mbps), and higher speeds of the TDM hierarchy defined in the next section. Analog lines require a modem for digital-to-analog conversion, while digital private lines require a DCE (commonly called a channel service unit/data service unit [CSU/DSU]) for line conditioning, framing, and formatting. Most local exchange, interexchange, and alternate access providers, as well as international carriers, offer digital private-line services.

Private-Line Networking

Figure 6-3 depicts a network of three users' DTEs connected via private lines. User A has a dedicated 56 Kbps circuit to user B, as well as a dedicated T1 (1.544 Mbps) circuit to user C. Users B and C have a dedicated 1.544 Mbps circuit between them. Users generally lease a private line when they require continuous access to the entire bandwidth between two sites. The user devices are voice private branch exchanges (PBXs), T1 multiplexers, routers, or other data communications networking equipment. The key advantage of private lines is that a customer has complete control over the allocation of bandwidth on the private-line circuits interconnecting these devices. This is also the primary disadvantage, in that the customer must purchase, maintain, and operate these devices in order to make efficient use of the private-line bandwidth. Up until the 1980s, most voice networks were private line based, primarily made up of dedicated trunks interconnecting PBXs. The situation changed once carriers introduced cost-effective intelligent voice network services. Now, most corporate voice networks use these carrier services. Data networking appears to be moving along a similar trend toward shared carrier–provided public data services. In the early 1990s, virtually every corporate data network was private line based. Now, many corporate users have moved from private lines to embrace frame relay, the Internet, and ATM for their data networking needs.

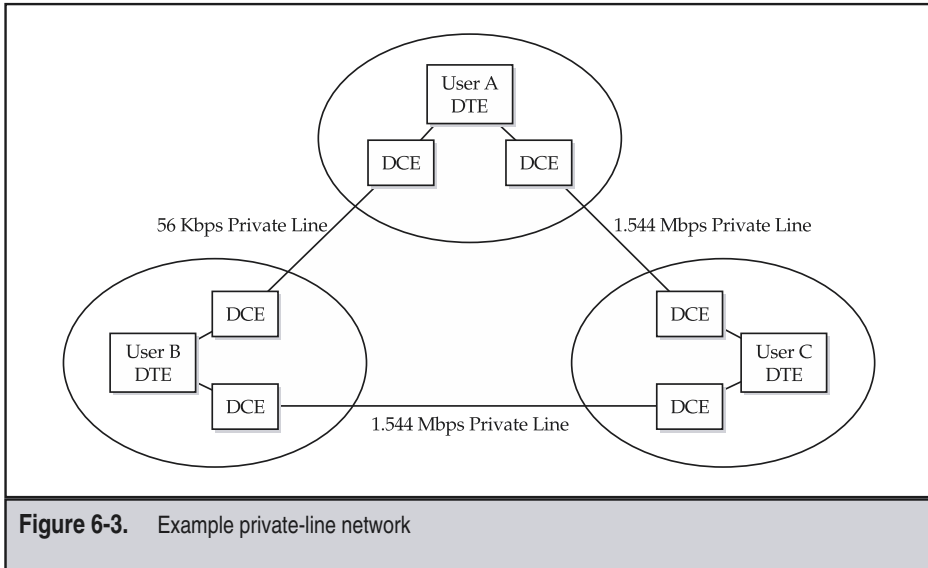
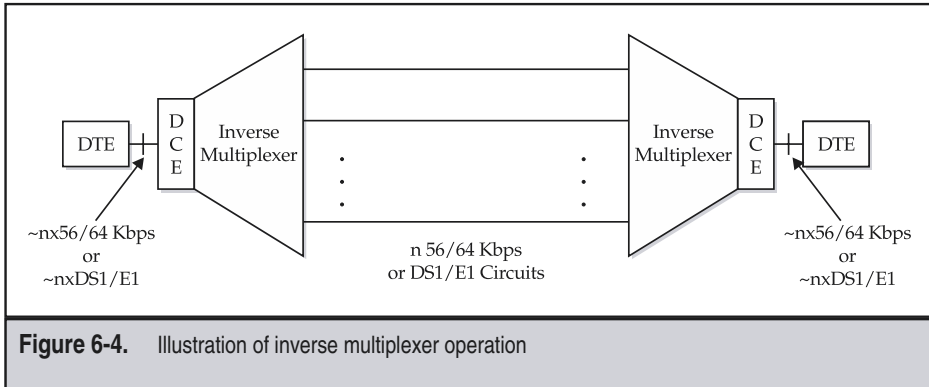


Figure 6-3. Example private-line network

While private lines provide dedicated bandwidth, carriers don't guarantee 100 percent availability, and sometimes a service provider statement of availability is related to the network and not an individual connection. Sometimes, a carrier provides for recovery of private-line failures via digital cross-connects, transmission protection switching, or SONET/SDH rings. However, in many cases, a private line comprises several segments across multiple carriers. For example, a long-haul private line typically has access circuits provided by local exchange carriers on each end and a long-distance segment in the middle provided by an interexchange carrier. If the private line or any of its associated transmission equipment fails (because of, e.g., a fiber cut), the end users cannot communicate unless the user DTEs have some method of routing, reconnecting, or dialing around the failure. Thus, the user must decide what level of availability is needed for communications between sites. Service providers usually provide a mean time to repair (MTTR) guarantee for a private-line user connection. This promises a user diligence in repairing a failed individual connection, usually within a time frame of two to five hours. There are two generic categories of restoration in TDM networks: linear and ring based. Linear restoration, commonly called protection switching, was implemented before SONET and SDH [Goralski 00]. It uses the concept of *working* and *protect* channels. Normally, a single protect channel protected n working channels, often indicated by the notation $1:n$ protection, pronounced as "1 for n " or "1 by n " or "1 to n " protection. When a working channel fails, the equipment at each end of a linear system quickly switches over the working

channel to the protect channel. If the protect channel is already in use (or already has failed), the working channel cannot be restored. For this reason, a commonly encountered deployment configuration for short transmission spans is 1:1 protection over diverse facilities; whereas, for longer spans, this 100 percent redundancy becomes expensive and is therefore often avoided. Also, if all $n + 1$ channels traverse the same physical route, then a single failure could disrupt all n working channels. Therefore, when protecting longer-distance systems, the working and protect channels should be on diverse physical facilities, although in reality only small values of n are practical because of a limited amount of diversity in the physical fiber plant. Before the advent of wavelength division multiplexing (WDM) systems, a $1 + 1$ transmission system required four fibers for operation (working and protect channels with respective transmit/receive pairs). The first step to better utilize the fiber plant was to multiplex transmit and receive signals onto one fiber, making it possible to double the capacity of the fiber. WDM systems now provide hundreds of wavelengths over a single fiber. However, WDM systems introduce another point of failure for SONET/SDH systems, and the ability to switch to redundant WDM systems becomes necessary. To provide 100 percent connection redundancy in such an environment would require $1 + 1$ types of SONET/SDH systems together with redundant WDM systems, a very expensive solution. More common deployments use diverse redundant WDM systems together with $1:n$ SONET/SDH systems (sometimes with diverse protection channels) or multinode ring systems.

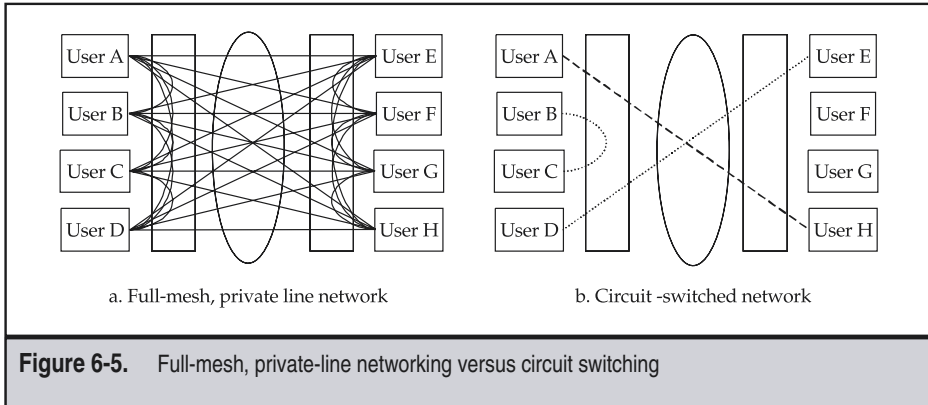
Protection switching was the precursor to ring switching, a subject we cover later in this chapter after introducing the SONET/SDH architecture and inverse multiplexers (I-Muxes). Early digital private-line networks offered service at only 56/64 Kbps or DS1/E1 (i.e., 1.5 and 2.0 Mbps) rates. The gap in speed and price between these speeds created a market for inverse multiplexers, commonly called I-Muxes, that provided intermediate-speed connectivity by combining multiple lower-speed circuits. As illustrated in Figure 6-4, an I-Mux provides a single high-speed DTE-DCE interface by combining n lower-speed circuits, typically private lines. Inverse multiplexers come in two major categories: $nx56/64$ Kbps and $nxDS1/E1$. The inverse multiplexer provides a DCE interface to the DTE operating at a rate of approximately 56/64 Kbps or DS1/E1 times n , the number of circuits connecting the I-Muxes. These devices automatically change the DTE-DCE bit rate in response to circuit activations or deactivations. The I-Muxes also account for the differences in delay between the interconnecting circuits. The actual serial bit rate provided to the DTE is slightly less than the $nx56/64$ Kbps or $nxDS1/E1$ rate because of overhead used to synchronize the lower-speed circuits. Some I-Muxes also support circuit-switched data interconnections in addition to private-line connections. A bonding standard defines how $nxDS0$ I-Muxes interoperate. Higher-speed $nxDS1/E1$ I-Muxes utilize a proprietary protocol and are hence incompatible among vendors. Chapters 7 and 11 describe frame relay and ATM standards for inverse multiplexing.



Permanent Versus Switched Circuits

We come back to the basic concepts of permanent circuits and switched circuits repeatedly in this text, so we begin with a simple example here as an introduction to the basic tradeoffs involved. Figure 6-5 shows a simplified comparison of two communications networks connecting eight users, labeled A through H, which could be LANs, MANs, PBXs, or hosts. Network (a) shows dedicated private-line circuits connecting each and every user to every other user, while network (b) shows circuit-switched access to a common, shared network with only a dedicated access line to the network for each user. In network (a), each user has seven access lines into the network for dedicated circuits connecting to a distant access line for each possible destination. Circuit switching transfers data or voice information at the physical layer. In other words, circuit switching is transparent to higher-layer protocols, which means that the network does not process the information content. The example in the circuit-switched network (b) shows user A talking to user H, and user D talking with user E. Any user can communicate with any other user, although not simultaneously, just as in the telephone network.

The signaling protocols employed by narrowband ISDN, X.25, frame relay, ATM, and MPLS all use the same basic circuit switching paradigm described in the previous section, except that the end device is usually some kind of computer or router instead of a telephone. The X.25, frame relay, ATM, and MPLS protocols all have a form of switched virtual circuit (SVC) capability similar to the telephone call described previously. Furthermore, they also have a permanent virtual circuit (PVC) capability analogous to a (virtually) dedicated pair of wires between each pair of end users wishing to communicate.



DIGITAL TIME DIVISION MULTIPLEXING (TDM)

Public network carriers first developed plesiochronous digital transmission for economical, high-quality transmission of voice signals. Later on, the carriers used these same transmission systems to offer private-line data services. In the 1990s, North American carriers began deployment of the Synchronous Optical Network (SONET), while the rest of the world deployed Synchronous Digital Hierarchy (SDH)-based transmission systems. The SONET and SDH systems provide higher speeds, standardized interfaces, automatic restoration, and superior transmission quality compared with the plesiochronous systems that preceded them. This section reviews some basics of these important TDM technologies.

Plesiochronous Digital Hierarchy (PDH)

The Plesiochronous (which means nearly synchronous) Digital Hierarchy (PDH) was developed in the 1950s by Bell Labs to carry digitized voice over twisted wire more efficiently in major urban areas. This evolved first as the North American Digital Hierarchy, depicted in Table 6-1. The convention assigns a level to each digital signal (DS) format in the hierarchy.

Plesiochronous transmission systems multiplex several lower-numbered digital streams into the higher-numbered digital streams within a certain frequency tolerance. No fixed relationship exists between the data between levels of the hierarchy—except at the lowest level, called a DS0, at a rate of 64 Kbps. Figure 6-6 illustrates a convention commonly used in North America to label multiplexing between the various levels of the hierarchy depicted in Table 6-1. For example, an M1C multiplexer converts 2 DS1s into a DS1c signal. An M12 multiplexer takes 2 DS1c signals and multiplexes these into a DS2 signal. Finally, an M13 multiplexer takes 7 DS2 signals and combines these into a single DS3 signal. Hence, an M13 multiplexer converts 28 DS1s into a DS3 signal but uses the M1C and M12 intermediate multiplexing stages to do so.

Signal Name	Rate	Structure	Number of DS0s
DS0	64 Kbps	Individual time slot	1
DS1	1.544 Mbps	24 × DS0	24
DS1c	3.152 Mbps	2 × DS1	48
DS2	6.312 Mbps	2 × DS1c	96
DS3	44.736 Mbps	7 × DS2	672

Table 6-1. North American Plesiochronous Digital Hierarchy

Bell Labs also defined a transmission repeater system over four-wire twisted pair and called it T1. Many trade press articles and even some technical books use the term “T1” to colloquially refer to a DS1 signal. There is actually no such thing as a “T3” signal, even though many people use this term colloquially when referencing a DS3 signal. The actual interfaces for DS1 and DS3 are called the DSX1 and DSX3 interfaces, respectively, in ANSI standards. The DSX1 is a four-wire interface, while the DSX3 interface is a dual coaxial cable interface.

Europe and Japan developed different Plesiochronous Digital Hierarchies as summarized in Table 6-2 [Kessler 85]. All of these hierarchies have the property that multiplexing is done in successive levels to move between successively higher speeds. Furthermore, the

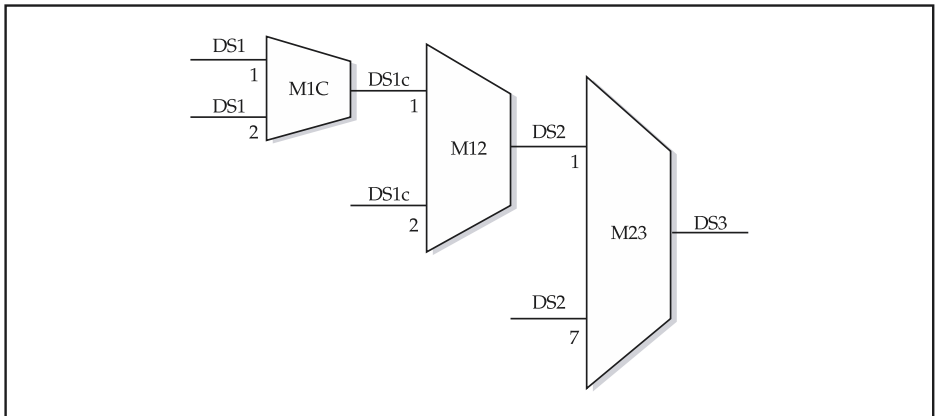


Figure 6-6. North American Plesiochronous Digital Hierarchy multiplexing plan

Digital Multiplexing Level	Number of Voice Channels	Bit Rate (Mbps)		
		North America	Europe	Japan
0	1	0.064	0.064	0.064
1	24	1.544		1.544
	30		2.048	
	48	3.152		
2	96	6.312		6.312
	120		8.448	7.876
3	480		34.368	32.064
	672	44.376		
	1344	91.053		
4	1440			97.728
	1920		139.268	
	4032	274.176		
	5760			397.200
5	7680		565.148	

Table 6-2. Summary of International Plesiochronous Digital Hierarchies

speed of each level is asynchronous with respect to the others within a certain frequency tolerance.

An important consequence of these digital hierarchies on data communications is that only a discrete set of fixed rates is available, namely $n \times \text{DS0}$ (where $1 \leq n \leq 24$ in North America and Japan and $1 \leq n \leq 30$ in Europe), and then the next levels in the respective multiplex hierarchies. The next section, on N-ISDN, defines the details of the DS0-to-DS1 and E1 mappings. Indeed, one of the early ATM proposals [Turner 86] emphasized the capability to provide a wide range of very granular speeds as a key advantage of ATM over TDM.

SONET and the Synchronous Digital Hierarchy (SDH)

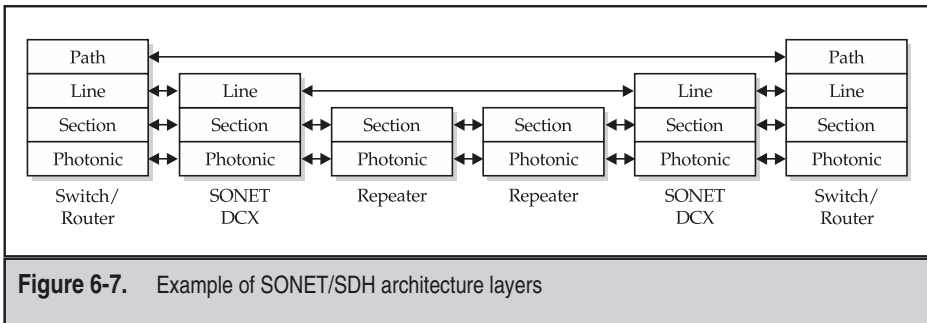
The Bellcore-driven North American standards defined a Synchronous Optical Network (SONET), while the CCITT/ITU developed a closely related international Synchronous

Digital Hierarchy (SDH) in the late 1980s. These standards were the next step in the evolution of time division multiplexing (TDM). SONET/SDH have two key benefits over PDH: rates of higher speeds are defined, and direct multiplexing is possible without intermediate multiplexing stages. Direct multiplexing employs pointers in the TDM overhead that directly identify the position of the payload. Furthermore, the fiber optic transmission signal transfers a very accurate clock rate along the transmission paths all the way to end systems, synchronizing the entire transmission network to a single, highly accurate clock frequency source.

Another key advance of SONET and SDH was the definition of a layered architecture (illustrated in Figure 6-7) that defines three levels of transmission spans. This model allowed transmission system manufacturers to develop interoperable products with compatible functions. The SONET/SDH framing structure defines overhead operating at each of these levels to estimate error rates, communicate alarm conditions, and provide maintenance support. Devices at the same SONET/SDH level communicate this overhead information as indicated by the arrows in Figure 6-7. The path layer covers end-to-end transmission, where ATM switches or MPLS label-switching routers (LSRs) operate as indicated in the figure. This text refers to a transmission path using this definition from SONET/SDH. Next comes the maintenance span, or line layer, which comprises a series of regenerators (or repeaters). An example of a line-layer device is a SONET/SDH cross-connect. The section regenerator operates between repeaters. Finally, the photonic layer involves sending binary data via optical pulses generated by lasers or light emitting diodes (LEDs).

SONET standards designate signal formats as *synchronous transfer signals (STSs)*; they are represented at N times the basic STS-1 (51.84 Mbps) building block rate by the term STS- N . SONET designates signals at speeds less than the STS-1 rate as virtual tributaries (VTs). The optical characteristics of the signal that carries SONET payloads is called the *optical carrier (OC- N)*. An STS- N signal can be carried on any OC- M , as long as $M \geq N$. The standard SONET STS and VT rates are summarized in the text that follows.

The CCITT/ITU developed a similar synchronous multiplex hierarchy with the same advantages using a basic building block called the synchronous transfer module (STM-1)



with a rate of 155.52 Mbps, which is exactly equal to SONET's STS-3 rate to promote interoperability between the different standards. The SDH standards also define a set of lower-speed signals, called virtual containers (VCs). Therefore, a direct mapping between the SONET STS-3N rates and the CCITT/ITU STM-N rates exists. An STM-1 frame is equivalent to an STS-3c frame in structure. The pointer processing and overhead byte definitions differ between SONET and SDH, so direct interconnection is currently not possible. Different vendor transmission equipment does not interwork easily even within SONET and SDH implementations, since proprietary management systems often utilize part of the protocol overhead for maintenance and operational functionality. Table 6-3 shows the SONET speed hierarchy by OC level and STS level as it aligns with the international SDH STM levels and the bit rates of each.

Table 6-4 illustrates a similar mapping to that of Table 6-3, comparing the mapping of the North American and CCITT/ITU PDH rates to the corresponding SONET virtual tributary (VT) and SDH virtual container (VC) rates and terminology. Note that the common 1.5, 2, 6, and 44 Mbps rates can be mapped consistently by using AU-3-based mapping, as shown later in Figure 6-11. SDH provides some alternative multiplexing paths for PDH signals, whereas SONET provides one way for each. SDH supports all PDH signals except for DS1C (3.152 Mbps). Also note that SONET does not support the popular International E3 rate. The other common rates are 155 and 622 Mbps and above. The rates indicated in the table include the actual payload plus multiplexing overhead, including path overhead. The table includes ATM-carried payload rates for commonly used ATM mappings over SONET/SDH for comparison purposes.

SONET and SDH standards evolved the state of the art in restoration through the concept of ring switching [Goralski 00]. A ring-switching architecture could protect either a line or path segment and make traffic flow in either one direction or both directions around the ring. SONET line switching operates on an entire OC-N bundle, while path switching can operate on a tributary. A unidirectional ring means that normal routing of

SONET Level	SDH Level	Bit Rate (Mbps)
STS-1	-	51.84
STS-3, OC-3	STM-1	155.52
STS-12, OC-12	STM-4	622.08
STS-24	-	1,244.16
STS-48, OC-48	STM-16	2,488.32
STS-192, OC-192	STM-64	9,953.28

Table 6-3. SONET STS-N/OC-N and SDH STM-M Speed Hierarchy

North American SONET Container (SPE)	SONET		SDH CCITT/ITU Container	SDH	
	SONET Payload Carried (Mbps)	Payload + Overhead (Mbps)		SDH Payload Carried (Mbps)	Payload + Overhead (Mbps)
VT1.5	1.544 (DS-1)	1.664	VC11	1.544 (DS-1)	1.728
			VC12	1.544 (DS-1)	2.304
VT2.0	2.048 (E1)	2.240	VC12	2.048 (E1)	2.304
VT3.0	3.152	3.392		N/A	N/A
VT6.0	6.312	6.848	VC2	6.312	6.912
STS-1	44.736 (DS-3)	50.112	VC3	44.736 (DS-3)	48.960
			VC3	34.368 (E3)	48.960
STS-3c	139.264 (OC-3)	150.336	VC4	139.264 (E4)	150.336
ATM on STS-1	49.536	50.112			
ATM on STS-3c	149.760	150.336	ATM on STM-1	149.760	150.336
ATM on STS-12c	599.040	601.344	ATM on STM-4	599.040	601.344

Table 6-4. SONET/SDH Digital Hierarchy Payload and Overhead Rates

both directions of working traffic flows in the same direction around the ring. Conversely, in a bidirectional ring, normal routing of the different directions of working traffic flow in the opposite direction around the ring. This results in the following ring-switching restoration schemes:

- ▼ Bidirectional line-switched ring (BLSR)
- Bidirectional path-switched ring (BPSR)
- Unidirectional line-switched ring (ULSR)
- ▲ Unidirectional path-switched ring (UPSR)

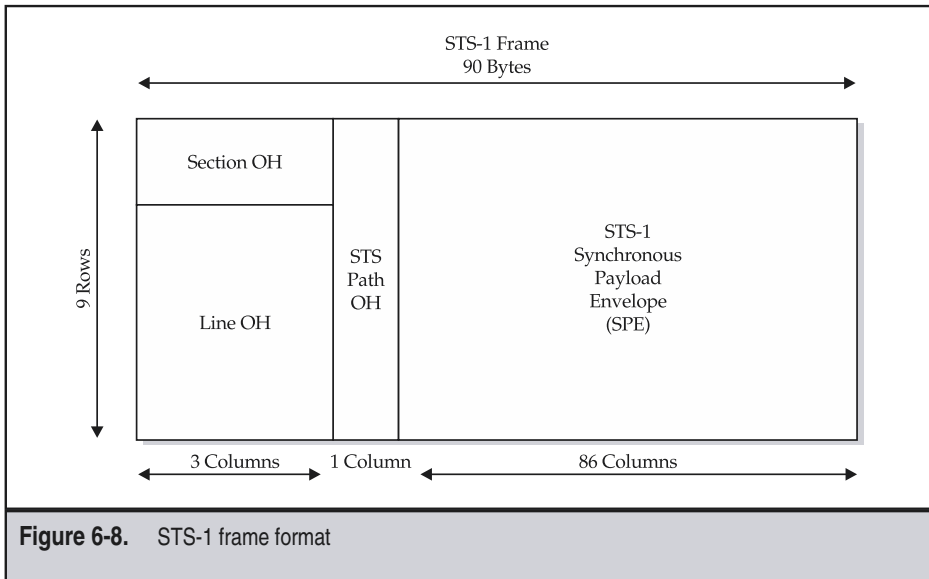
Any of these schemes can operate using either two or four fibers. This results in a potential for eight combinations; however, commonly fielded configurations are two-fiber UPSR, two-fiber BLSR, and four-fiber BLSR. A number of tradeoffs exist in deciding which ring technology is best. The directions of traffic flow in unidirectional rings have asymmetric delays, while bidirectional rings have symmetric delay. Two-fiber rings are often deployed in metropolitan areas, while four-fiber rings are more efficient and can protect against some scenarios involving multiple failures in regional deployments.

Two-fiber rings are less expensive to deploy initially but are less efficient than four-fiber rings when fully loaded.

The large-scale deployment of SONET/SDH rings by carriers and ISPs is important to performance, since the switchover time from the occurrence of a failure to a completely restored circuit line or path segment is 50 ms or less. The fact that SONET/SDH rings restore traffic in quite literally the blink of an eye has significant consequences. It means that a click might be heard on a voice call, a digitally transmitted video might lose a few frames, or packet transfer would be momentarily disrupted. We will see later that ATM and MPLS technology is striving to meet the impressive restoration time performance of SONET/SDH rings. It is interesting to note that dense wavelength division multiplexing (DWDM) ring implementations also leverage the basic concepts of these SONET/SDH ring protection schemes.

Basic SONET Frame Format

Figure 6-8 illustrates the SONET STS-1 frame format. Notice that the frame is made up of multiple overhead elements (section, line, and path) and a synchronous payload envelope (SPE). The frame size for an STS-1 SPE is 9 rows \times 90 columns (1 byte per column) for a total of 810 bytes, composing a 783-byte frame (excluding the 27 byte section and line overhead). The total STS-1 frame of 810 bytes transmitted each 125 μ s results in the basic STS-1 rate of 51.84 Mbps. The STS-N SPE format essentially replicates the STS-1 format N times.



The basic transmission rate for SDH is the 155.520 Mbps STM-1. The STM-1 frame consists of 2430 bytes, corresponding to the frame duration of 125μs. The basic SDH structure is a 9 row × 270 columns frame, where the overhead consist of the first 9 columns (81 bytes pointer and section overhead). The STM-1 forms the basis for higher rates by replicating this format just as in SONET. Since all STM- and STS-level signals are synchronous, multiplexing to higher rates is achieved by simple byte interleaving. Note that in order to keep the signal structure intact, in forming an STM-M higher-rate signal, the STM-N signals must be M/N byte interleaved. For example, to form an STM-4, four STM-1 signals are 4-byte interleaved. STS-level signals are byte interleaved the same way.

Figure 6-9 illustrates the mapping of VT1.5s into an STS-1 SONET synchronous payload envelope (SPE). Each VT1.5 uses 27 bytes to carry 24 bytes of a DS1 payload. The first column of 9 bytes is the STS-1 path overhead. The next 28 columns are bytes 1 through 9 of the (28) VT1.5 payloads, followed by a column of stuff bytes. Similarly, columns 31 through 58 are bytes 10 through 18 of the (28) VT1.5 payloads, followed by a column of stuff bytes. The last 28 columns are bytes 19 through 27 of the VT1.5 payloads.

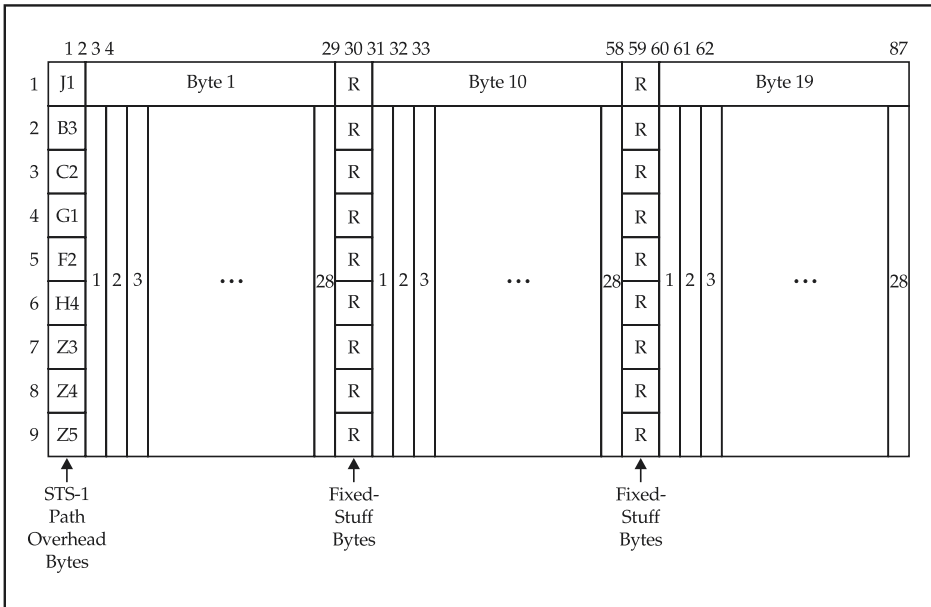


Figure 6-9. VT1.5 mapping within an STS-1 frame

Figure 6-10 shows the format of an individual VT1.5. Note that 27 bytes are transferred every 125 μ s in the SPE as defined previously, but that only 24 bytes are necessary to carry the user data. User data byte 25 is included to be able to carry the DS1 framing bit transparently. The other 2 bytes provide a pointer so that the VT can “float” within its allocated bytes and thus allow for timing to be transferred, as well as provide for VT-level path overhead. The SONET overhead results in a mapping that is less than 100 percent efficient. In fact, VT1.5 multiplexing for support of individual DS0s is approximately 85.8 percent efficient (i.e., $24 \times 28/87/9$) in terms of SPE bandwidth utilization. As shown in Chapter 16, ATM support for transport of DS0s using structured-mode circuit emulation is approximately 87.4 percent efficient over SONET.

In SDH, plesiochronous signals are mapped into the payload area of the STM-1 frame by adding path overhead to form a virtual container (VC) of an appropriate size (e.g., a T1 in a VC-11, a E1 in a VC-12, or a DS3 or E3 into a VC-3), as illustrated in Figure 6-11. SDH also defines two types of path overhead, depending on whether the mapping performed is VC-2/VC-1 or VC-4/VC-3. A VC is positioned in either a tributary unit (TU) or an administrative unit (AU). VC-1 and VC-2 are positioned in TUs, whereas the VC-4 is always positioned in an AU-4. VC-3s are positioned in a TU-3, but the SONET option indicated in Figure 6-11 will always use an AU-3. TUs and AUs are each bundled into tributary unit groups (TUGs) and administrative unit groups (AUGs), respectively. TUGs can be multiplexed into higher-order VCs, which, in turn, are positioned in AUs with a pointer indicating the start of the VC relative to the AU. This is the same AU pointer used to indicate the position of the AU in relation to the STM-1 frame [ITU G.707].

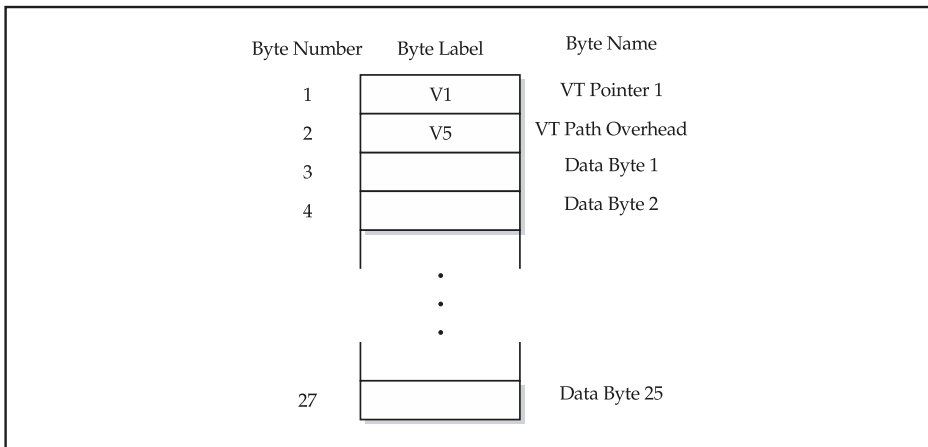
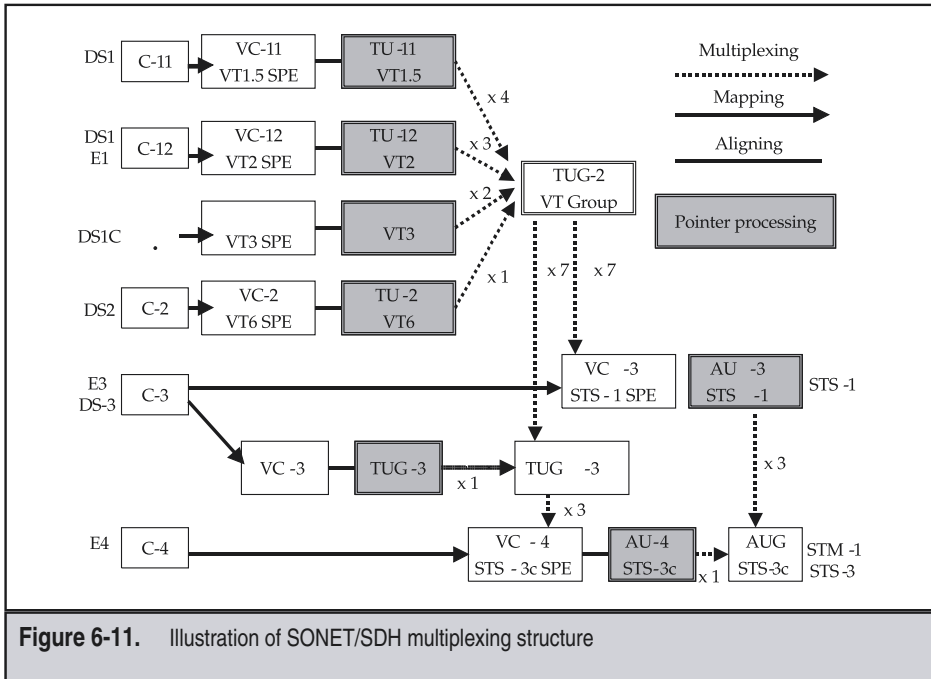


Figure 6-10. Illustration of SONET VT1.5 format



BASICS AND HISTORY OF NARROWBAND ISDN (N-ISDN)

N-ISDN is the phoenix of the late 1990s. Rising out of the ashes of its slowly smoldering adoption in the preceding decade, it takes its place as a switched alternative to the dedicated 56 Kbps analog local loop; an alternative dial access protocol to services like the Internet; and other corporate uses, such as an on-demand backup facility. We now turn our attention to the original Integrated Services Digital Network (ISDN) standards, where the frame relay and ATM protocols discussed in this book have their roots. First, we summarize the N-ISDN Basic Rate Interface (BRI) and Primary Rate Interface (PRI) configurations. Next, the text covers basic N-ISDN protocol and framing structure. In order to differentiate the ATM-based Broadband ISDN (B-ISDN) from the earlier narrowband ISDN, the standards refer to these protocols as Narrowband ISDN (N-ISDN), a term we use consistently in this book to avoid confusion. See Reference 6 for more N-ISDN information.

Narrowband ISDN Basics

N-ISDN builds upon the TDM hierarchy developed for digital telephony. Although most N-ISDN standardization is complete, the CCITT/ITU-T continues to define new standards for the N-ISDN. Two standards exist for the physical interface to N-ISDN: the Basic

Rate Interface (BRI), or basic access, as defined in ITU-T Recommendation I.430, and the Primary Rate Interface (PRI), as defined in ITU-T Recommendation I.431.

Figure 6-12 illustrates the N-ISDN functional groupings and reference points (as defined in ITU-T Recommendation I.411). Both the BRI and PRI standards define the electrical characteristics, signaling, coding, and frame formats of N-ISDN communications across the user access interface (S/T) reference point. The physical layer provides transmission capability, activation and deactivation of terminal equipment (TE) and network termination (NT) devices, data (D) channel access for TEs, bearer (B) channels for TEs, maintenance functions, and channel status indications. ITU-T Recommendation I.412 defines the basic infrastructure for these physical implementations, as well as the detailed definition for the S and T reference points, TEs, and NTs. The TA manufacturer defines the R reference point to non-ISDN terminal equipment. The CCITT/ITU-T defines two possible network interface points at the S and T reference points where a carrier always places equipment on the customer's premises. In the United States, no formal network boundary exists; however, ANSI standards define this as the U reference point, as indicated in Figure 6-12.

The BRI and PRI N-ISDN interfaces provide a set of bearer B-channels and a D-channel, as described previously. As illustrated in Figure 6-13, the B-channels provide a layer 1 service to the N-ISDN terminal equipment, while the D-channel supports a layer 3 signaling protocol for control of the B-channels. Optionally, end-user N-ISDN equipment may transfer limited amounts of packet data over the D-channel. Note that the NT1 device operates at only layer 1 for the D-channel. As we shall see, B-ISDN utilizes a similar concept when labeling the signaling protocol the control plane and the bearer capabilities the user plane.

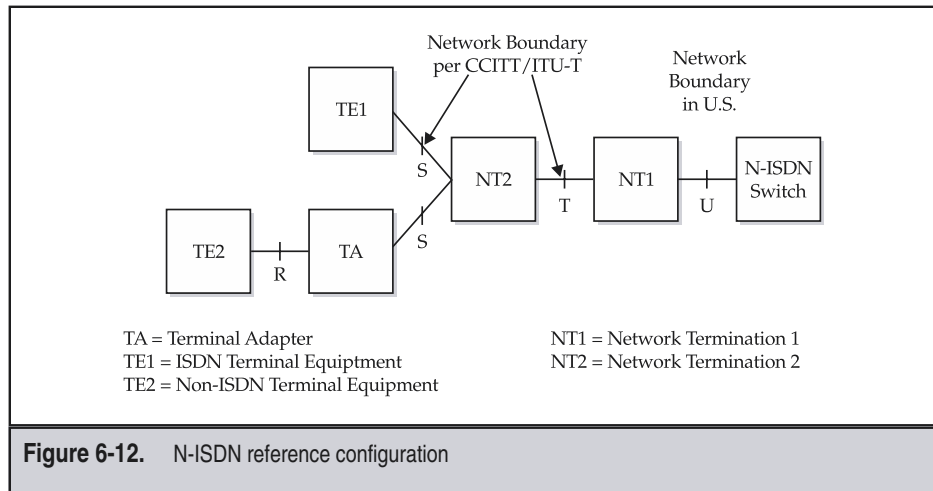


Figure 6-12. N-ISDN reference configuration

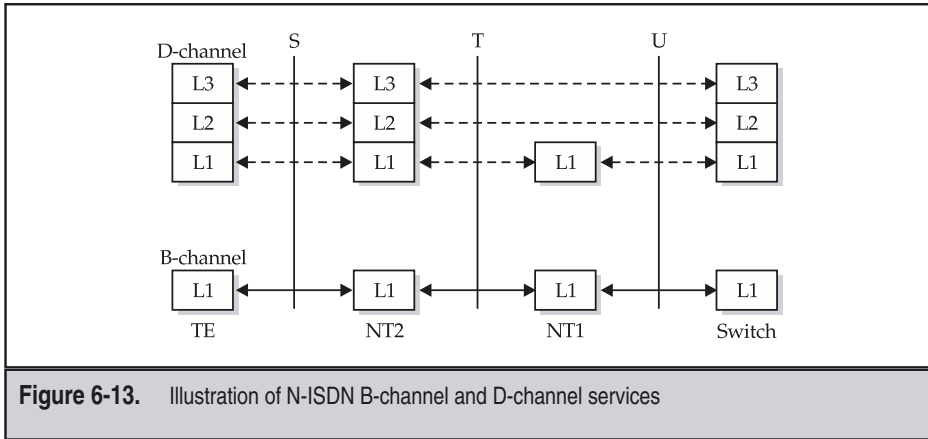


Figure 6-13. Illustration of N-ISDN B-channel and D-channel services

BRI and PRI Service and Protocol Structures

The N-ISDN Basic Rate Interface (BRI) and Primary Rate Interface (PRI) service configurations are defined as follows:

- ▼ **Basic Rate Interface (BRI)** Provides two 64 Kbps bearer (B) channels for the carriage of user data and one 16 Kbps control, messaging, and network management D-channel. Documentation commonly refers to the BRI as a 2B+D interface. The BRI was intended for customer access devices such as N-ISDN voice, data, and videophone. Many Internet service providers now use the BRI through the local telephone company to provide high-performance access to the Internet at speeds up to 128 Kbps.
- ▲ **Primary Rate Interface (PRI)** In North America, provides twenty-three 64 Kbps B-channels and one 64 Kbps signaling D-channel, commonly referred to as a 23B+D interface. Internationally, 30 B-channels are provided in a 30B+D configuration. The PRI was intended for use by higher-bandwidth or shared-customer devices such as the Private Branch Exchange (PBX), routers, or T1 multiplexers.

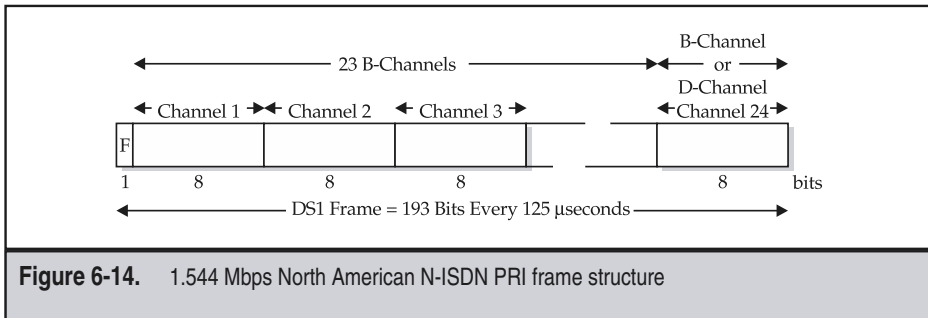
The N-ISDN BRI operates over the same twisted pairs used for telephone communication at a physical data rate of 192 Kbps full duplex, of which 144 Kbps is for user data (i.e., 2B+D = $2 \times 64 + 16 = 144$). BRI may operate in either a point-to-point or point-to-multipoint mode. CCITT Recommendation I.430 details the BRI layer 1 protocol. BRI devices are significantly more complex than telephony devices. The line coding used by BRI was considered sophisticated in the late 1980s. However, it now pales in comparison with modern digital subscriber line (DSL) technology that operates over the same twisted pair telephone lines at speeds that are an order of magnitude larger as described in Chapter 11.

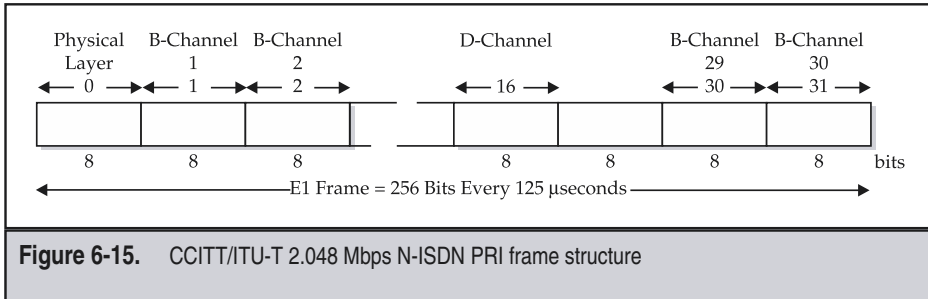
The N-ISDN PRI provides a single 1.544 Mbps DS1 or a 2.048 Mbps E1 data rate channel over a full-duplex synchronous point-to-point channel using the standard TDM hierarchy introduced earlier in this chapter. CCITT Recommendations G.703 and G.704 define the electrical characteristics and frame formats of the PRI interface, respectively. The 1.544 Mbps rate is accomplished by sending 8000 frames per second with each frame containing 193 bits. Twenty-four DS0 channels of 64 Kbps each compose the DS1 stream. Figure 6-14 shows the format of the DS1 PRI interface. Note that the 193rd framing bit is defined by DS1 standards for error rate estimation and maintenance signaling. A DS1 PRI contains at least 24 B-channels. The 24th DS0 time slot contains either the signaling D-channel, or a 24th B-channel if another D-channel controls this DS1.

The CCITT/ITU-T E1-based PRI interface differs somewhat from the DS1 interface, as shown in Figure 6-15. The E1 PRI has 30 B-channels; one 64 Kbps D-channel in time slot 16; and a channel reserved for physical layer signaling, framing, and synchronization in time slot 0. A primary attribute distinguishing N-ISDN service from telephony is the concept of common channel signaling, or out-of-band signaling using the D-channel. The D-channel and B-channels may share the same physical interface as indicated in the previous illustrations, or the D-channel on one interface may control the B-channels on several physical interfaces.

Since PRI's run at higher speeds, they support additional bearer capabilities called *H-channels*. Two types are defined: H_0 -channel signals that have a bit rate of 384 Kbps, and H_1 -channels defined for DS1 and E1 PRI's. H_{11} -channels have a bit rate of 1536 Kbps in the United States and Japanese DS1 PRI's, while H_{12} -channels operate at 1920 Kbps on E1 PRI's in Europe and other parts of the world. The H_0 -channel uses any six time slots in the same DS1 or E1; that is, the time slots need not be contiguous. The H_{11} -channel uses all 24 time slots in a DS1, which means that the D signaling channel must be on a separate physical interface.

Standards also define a capability to establish an $n \times 64$ Kbps bearer service, where n ranges from 1 to 24 (or 30 at the European channel rate) via N-ISDN signaling. The $n \times DS0$

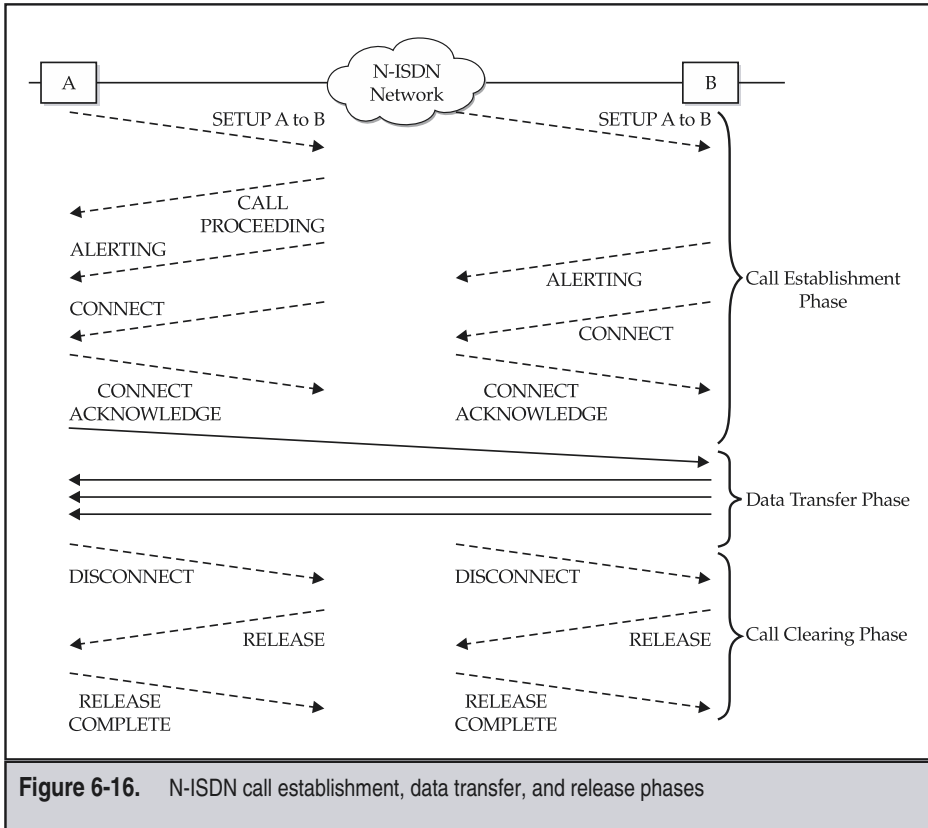




service uses n contiguous time slots or a bit map—specified set of DS0 time slots in the DS1 or E1 frame. Standards call this capability the Multirate Circuit-Mode Bearer Service (MRCMBS). Also, N-ISDN signaling can establish a frame-mode bearer service (FMBS) or a switched X.25 connection, as discussed in the next chapter.

ISDN D-Channel Signaling

CCITT Recommendation Q.931 defines a set of message types containing a number of *information elements* used to establish and release N-ISDN bearer connections. Figure 6-16 illustrates an example sequence of such messages involved during the establishment, data transfer, and release phases of an N-ISDN call. Starting in the upper left-hand corner of the figure, terminal equipment A places a call to B via the N-ISDN network using the SETUP message. The network responds with a CALL PROCEEDING message and relays the call request to the destination, issuing a SETUP message to B. In the example, B alerts the called user and informs the network of this action via the ALERTING message, which the network relays to A. Eventually, B answers the call with the CONNECT message, which the network acknowledges using the CONNECT ACKNOWLEDGE message. The N-ISDN network also relays the connect indication back to calling party A, indicating the response using the CONNECT message. Usually, the CONNECT message identifies the time slot(s) assigned to the bearer connection. Once A confirms the completed call using the CONNECT ACKNOWLEDGE message, A and B may transfer data over the bearer channel established by this signaling protocol for an indefinite period of time. Either party may release the connection by sending the DISCONNECT message to the network. In this example, A initiates call clearing, which the network propagates to user B in the form of the DISCONNECT message. The normal response to a DISCONNECT message is the RELEASE message followed by a RELEASE COMPLETE message, as indicated in the figure. The information element content of the N-ISDN signaling messages (as well as some additional messages not included in this simple example) emulate all of the traditional functions available in telephony, along with support for a number of additional advanced features.



Upon receiving a SETUP message for an N-ISDN call, network switches establish an end-to-end ISDN B- or H-channel between the source and the indicated end destination. The service provider's network, however, uses a different signaling system. Signaling between the remote ISDN devices and the public voice and data network switches occurs using D-channel protocols such as Q.931, which, in turn, are converted into Signaling System No. 7 (SS7) signals within the carrier's digital voice and data networks. With SS7, carriers are able to maintain clear channel N-ISDN connections by communicating signaling information in a distinct separate channel. The switch at the destination side of the network then communicates with the remote ISDN device using its D-channel protocol.

REVIEW

This chapter introduced the basic concepts of circuit switching and private line-based networking, providing design recommendations throughout. Study of these subjects is important because the circuit-switching paradigm described here appears again in X.25, frame relay, ATM, and MPLS. Next, the text described the digital time division multiplexing (TDM) hierarchy and its evolution to the North American Synchronous Optical Network (SONET) and the international Synchronous Digital Hierarchy (SDH). Originally designed to provide more cost-effective telephone calls, TDM now provides the foundation for the high-performance digital data communications central to IP, MPLS, and ATM. Upon this foundation, the standards bodies constructed the Narrowband Integrated Services Digital Network (N-ISDN) protocol model. As studied in Part 3, the ATM-based Broadband ISDN (B-ISDN) protocol adopts the concepts of separate user, control, and management protocols from N-ISDN. Analogous terminology is used within the industry to distinguish between forwarding and control protocols within the parlance of IP and MPLS.



CHAPTER 7



Connection-Oriented Protocols—X.25 and Frame Relay

This chapter presents an overview of X.25 packet-switching and Frame Relay–forwarding protocols used in major public and private connection-oriented data services. The text describes aspects of each protocol, beginning with the origins of the protocol, followed by an overview of the packet formats and protocol functions. The coverage includes operation of the protocol through an illustrated example. The traffic and congestion control aspects of the Frame Relay and X.25 protocols are also surveyed. We touch on the aspects of the protocol that are supported in public services. Since ATM and MPLS inherit many concepts from X.25 and Frame Relay, we cover the relevant subjects in sufficient detail to make this book a stand-alone reference for the reader without background knowledge of these important protocols. This chapter provides an example of the separation of the user and control planes central to the treatment of Frame Relay.

PACKET SWITCHING

The CCITT standardized X.25 as the earliest public data network protocol in 1974, continued with refinements and corrections over the years, and most recently updated the standard in 1996 [ITU X.25]. X.25 packet switching provides the network environment needed to handle intermittent terminal-to-host data traffic. *Packet switching* refers to protocols in which the information is broken up into smaller packets before being sent. Each packet is transmitted individually, and various packets may even follow different routes to the destination. Thus, each packet has a header containing information about how to reach the destination. At the destination, the packets are reassembled into the original data. Most modern wide area network (WAN) protocols, such as TCP/IP, X.25, and Frame Relay, are based on packet-switching technologies. The typical packet-switching application involves a user inputting keyboard data ranging from a few characters to a few lines and then forwarding the information to a host. Typically, the host then responds with a set of data ranging from many lines to a full-screen display. An interval of user “think time” separates these interchanges, resulting in traffic that has a much higher peak transmission rate than the average transmission rate. The data communications industry uses the term *burstiness* to describe the ratio of peak-to-average transmission rates, derived from the experience with X.25 and SNA networking.

Human nature changes slower than technology. As evidence of this fact, note that this same concept of bursty communication applies today in the Web browser user–network interaction of the modern Internet. The basic paradigm for Web surfing involves a user inputting a set of data ranging from a single mouse click to an entire form and submitting it to the Web server. The server then responds by transmitting an updated Web page. Sometimes the user input kicks off the playback of an audio or video clip, or initiates a file transfer. What has changed from the days of X.25 to the Web-fueled content of the Internet today is the user’s power to unleash bandwidth-hungry and QoS-aware applications.

Origins of X.25

In the beginning, there were proprietary protocols; then the CCITT standardized upon the first international physical, link, and network layer protocol—X.25. The CCITT developed

the X.25 packet-switching standard, along with a number of other X-series standards, to provide a reliable means of data transport for computer communications over the noisy, unreliable analog-grade transmission medium prevalent in the 1970s. By the 1980s, X.25 networks connected the entire planet. X.25 packet switching still serves user communities in public and private networks.

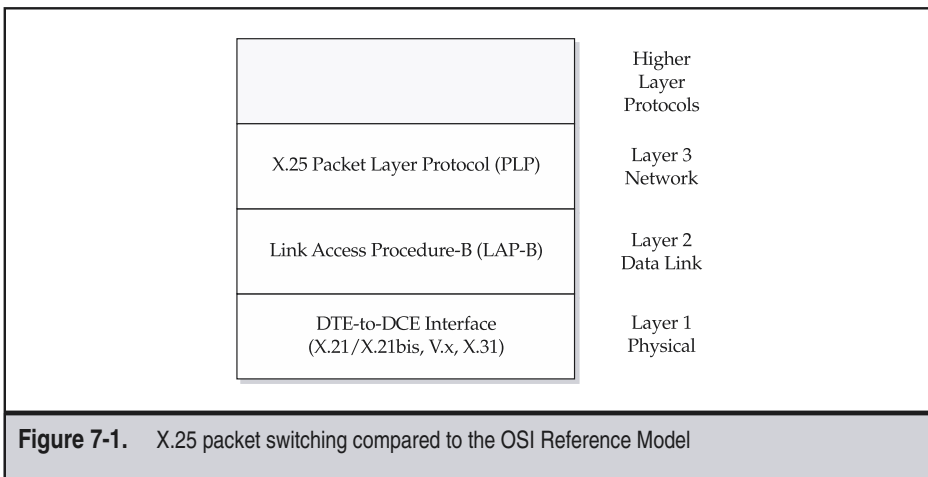
Protocol Structure

The CCITT set of X-series standards for the physical, link, and packet layer protocols shown in Figure 7-1 are known collectively as X.25 and were adopted as part of the OSI Reference Model (OSIRM). These standards define the protocol, services, facilities, packet-switching options, and user interfaces for public packet-switched networks.

The physical layer is defined by the X.21 and X.21bis standards. X.21 specifies an interface between data terminal equipment (DTE) and data communications equipment (DCE). X.21 also specifies a simple circuit-switching protocol that operates at the physical layer implemented in the Nordic countries.

The data link layer standard is based upon the High-Level Data Link Control (HDLC) ISO standard [ISO 13239]. X.25 modified this and initially called it a Link Access Procedure (LAP), subsequently revising it again to align with changes in HDLC, resulting in the Link Access Procedure Balanced (LAP-B).

The packet layer standard is called the X.25 Packet Layer Protocol (PLP). The packet layer defines permanent virtual circuit (PVC) and switched virtual call (VC) message formats and protocols. As we study later, the concept of PVCs and switched VCs from X.25 is also used in Frame Relay and ATM.

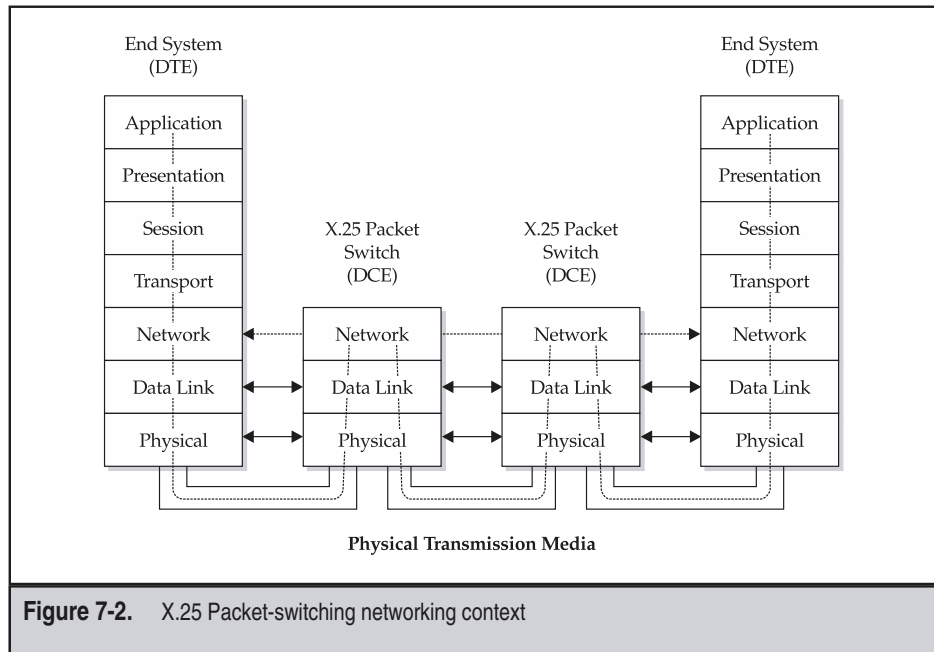


The suite of X-series standards covers the entire OSIRM protocol stack, including X.400 (e-mail) messaging and X.500 directory services [Black 95]. The standards related to the network, data link, and physical layers include X.121, X.21, X.25, X.28, X.29, X.3, and X.32. Recommendation X.121 defines the network layer numbering plan for X.25.

CCITT Recommendations X.3, X.28, and X.29 define the method for asynchronous DTEs to interface with X.25 networks via a packet assembler/disassembler (PAD) function. A PAD takes strings of asynchronous characters from the DTE and assembles these into an X.25 packet. The PAD takes data from X.25 packets and delivers asynchronous characters to the DTE. Recommendation X.3 defines PAD parameters, such as terminal characteristics, line length, break key actions, and speed. Recommendation X.28 defines the terminal-to-PAD interface, while Recommendation X.29 defines the procedures governing communications between a PAD and a remote packet mode DTE on another PAD. Recommendation X.21 defines a dedicated physical interface, and Recommendation X.32 defines a synchronous dial-up capability. X.25 also supports the V-series of modem physical interfaces, as well as Recommendations X.31 and X.32 for semipermanent ISDN connections.

Networking Context

Figure 7-2 shows how the X.25 protocol layers operate in a network context interconnecting two end systems—for example, a terminal and a host. In the example, two interconnected intermediate X.25 switches transfer packets between a terminal and a host. The X.25 link



layer and network layer protocols define procedures for the establishment of multiple virtual circuits over a single physical interface circuit interconnecting terminals and hosts to an X.25 network. Once an X.25 virtual circuit is established, it usually traverses the same physical path between end systems. Each node operates at the physical, link, and network layers, as shown in Figure 7-2. X.25 packet switches store each packet and then forward it to the next node using a link layer protocol. The transmitting switch deletes the packet from memory only after its link-level peer acknowledges receipt.

Some aspects of the operation of the network layer occur only on an end-to-end basis (e.g., packet layer flow control), as indicated by the dashed arrow connecting the end systems in Figure 7-2. Of course, X.25 switches use the packet layer address to determine the forwarding path, as indicated by the dashed line traversing the layers in the figure. Also, note that the internal interface between X.25 packet switches could be some other protocol, such as Frame Relay. End systems (e.g., terminals and hosts) also operate at layers 4 through 7 (i.e., transport through application) using either OSI-compatible protocols or other protocol suites, such as SNA or TCP/IP. Now we take a more detailed look at the X.25 protocol involved in layer 2, the link layer, and then layer 3, the network layer.

SDLC, HDLC, and X.25's Link Layer Protocol

This section covers the origins and details of X.25's link layer protocol. We begin with IBM's SDLC protocol and move on to the ISO's enhancements, resulting in HDLC. This section concludes with an overview of Link Access Procedures (LAPs) defined by the ITU-T for X.25, ISO's enhancements, resulting in HDLC. This section concludes with an overview of the Link Access Procedures (LAPs) defined by the ITU-T for X.25, ISDN, and Frame Relay.

Synchronous Data Link Control (SDLC)

In 1973, IBM produced the first bit-oriented data communications protocol, called Synchronous Data Link Control (SDLC). Previous protocols (e.g., IBM's BSC) were all character oriented. SDLC, as well as subsequent bit-oriented protocols, allowed computers to transfer arbitrary binary sequences commonly encountered in programs and databases. Also, messages no longer needed to be precisely aligned on an eight-bit character boundary. The International Organization for Standardization (ISO) adopted this de facto standard and extended it into the widely used High-Level Data Link Control (HDLC) protocol. The present version of IBM's SDLC primarily uses the unbalanced normal response mode of HDLC together with a few proprietary commands and responses for support of polling in loop or ring topologies. SDLC operates independently on each communications link and can operate in multipoint or point-to-point, switched or dedicated circuit, and full- or half-duplex operation.

SDLC replaced the BSC protocol described in Chapter 4. Some improvements of SDLC over BSC include the ability to send acknowledgments; addressing, block checking, and polling within every frame rather than in a separate sequence; the capability to handle long propagation delays; no restrictions to half-duplex; absence of susceptibility to missed or duplicated blocks; topology independence; and character code transparency.

High-Level Data Link Control (HDLC) Terminology

The HDLC protocol is not only the most popular protocol for data link control implementations at layer 2, but it also forms the basis for ISDN, Frame Relay, PPP, and the Packet over SONET (POS) version of MPLS. HDLC is an international standard, defined in the document jointly produced by the ISO and the International Electrotechnical Commission (IEC) as ISO/IEC 13239. HDLC is a bit-oriented synchronous protocol passing variable length frames over either a point-to-point or multipoint network topology. HDLC operates over either dedicated or switched facilities. HDLC operates in simplex, half-duplex, or full-duplex modes. X.25 uses the HDLC protocol for both PVCs and switched virtual calls.

Two types of HDLC control links operate over point-to-point circuits for the HDLC connection mode: balanced and unbalanced. Both control link types work on either switched or nonswitched (i.e., dedicated) facilities. In a *balanced link*, each station is responsible for organization and transmission of the information flow, as well as the use of an acknowledgment for error recovery, as illustrated in Figure 7-3. The HDLC standards name the stations as combined (primary/secondary), as indicated in the figure.

For connection-oriented HDLC, *unbalanced links* involve a primary station and a secondary station, as shown in Figure 7-4. In the unbalanced link, the primary/control station polls the secondary/tributary station, which responds with information frames. The primary station then sends an acknowledgment for receipt of frames from the secondary station. Information flow from the primary to the secondary station occurs within the polled information flow. The unbalanced link emulates the IBM SDLC protocol. The unbalanced link also defines procedures for operation over multipoint lines.

HDLC defines three operational modes for data transfer. The *asynchronous balanced mode (ABM)* applies to the balanced link configuration described previously. The other two types apply only to the unbalanced link configuration: the *normal response mode (NRM)*, which requires that the secondary station wait for a poll command from the primary station prior to transferring data, and also the *asynchronous response mode (ARM)*, which allows a secondary station to transmit data to the primary station if it detects an idle channel.

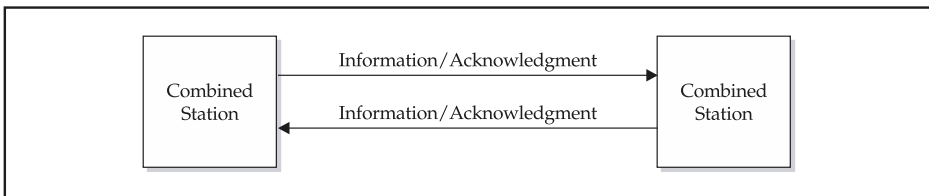
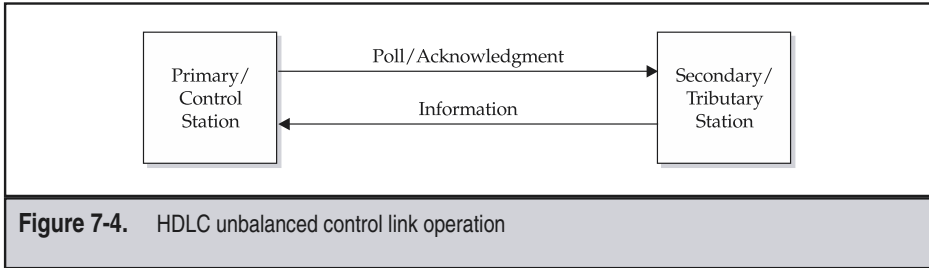


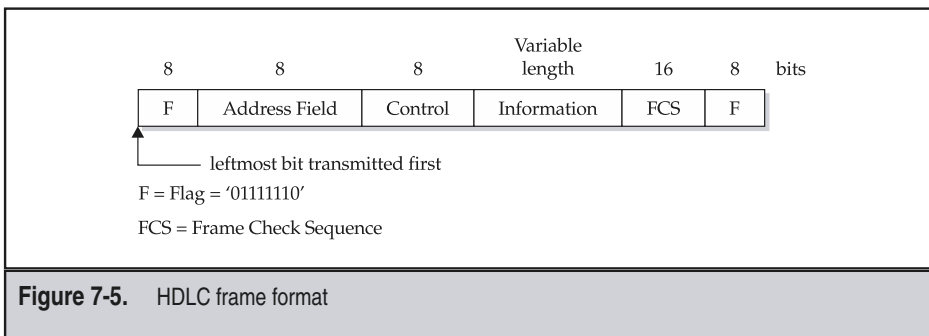
Figure 7-3. HDLC balanced control link operation



HDLC Frame Formats

The basic one-octet control field HDLC frame format shown in Figure 7-5 is used for both information exchange and link-level control. Two flag fields always encapsulate a frame; however, the closing flag for one frame may be reused as the opening flag for the subsequent frame. The HDLC frame format supports several control field formats. An address field provides the address of the secondary station (but is not needed for point-to-point configurations). The information field contains the data being transmitted, and the frame check sequence (FCS) performs error detection for the entire frame. Also included in this frame is a *control field* to identify one of three types of frames available.

The Flag (F) sequence is a zero followed by six ones and another zero. Flags delimit the beginning and end of an HDLC frame. A key function of the data link layer is to encode the occurrence of the flag sequence within user data as a different sequence using *bit stuffing* as follows: If the link layer detects a sequence of five consecutive ones in the user data, then it inserts a zero immediately after the fifth one in the transmitted bit stream. The receiving link layer removes these inserted zeros by looking for sequences of five ones followed by a “stuffed” zero bit. Thus, if an HDLC flag bit pattern, 01111110, is present in the user data; the link layer transmits this as 011111010. Unfortunately, HDLC’s bit-stuffing mechanism can be fooled by bit errors on the physical medium, as



we shall see later in Chapter 23. Therefore, many higher-layer protocols also keep a length count to detect errors caused by bit errors corrupting the HDLC bit-stuffing procedure. To determine the boundaries of HDLC frames, the receiver need only check the incoming bit stream for a zero followed by six ones.

The address field of the LAP-B frame is primarily used on multidrop lines. The address field also indicates the direction of transmission and differentiates between commands and responses, as we detail in the next section.

The sender computes the two-octet frame check sequence (FCS), and the receiver uses the FCS to check the received HDLC frame to determine if any bit errors occurred during transmission. The following generator polynomial specifies the FCS:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

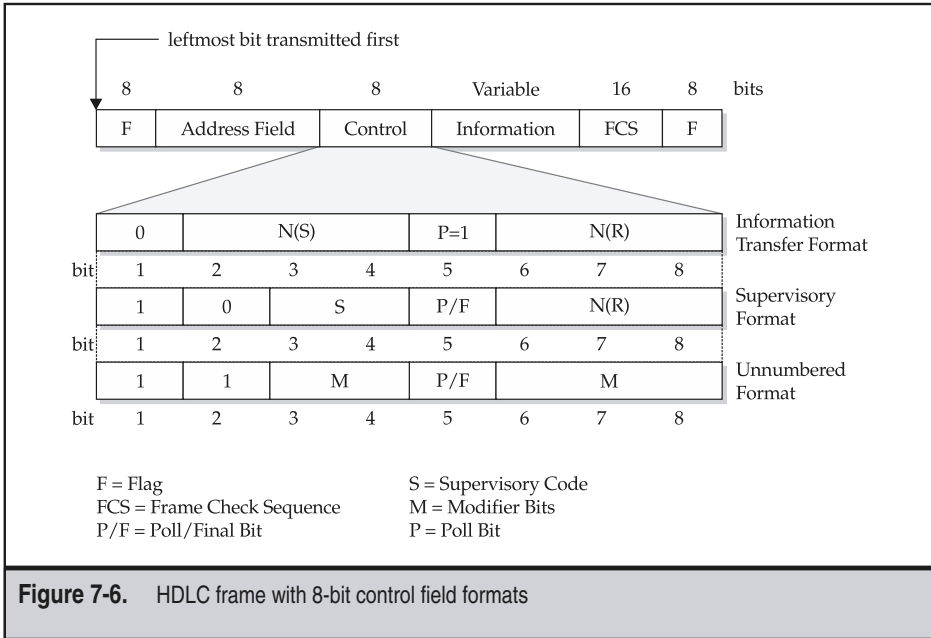
See Chapter 23 for a more detailed description of how generator polynomials are used to generate a cyclical redundancy check (CRC) error-detection field. The FCS of HDLC is capable of detecting up to three random bit errors or a burst of sixteen bit errors.

The HDLC standard supports control field frame formats of length equal to 8-, 16-, 32-, or 64-bit lengths negotiated at link establishment time. The control field of the X.25 LAP-B frame is identical to the corresponding HDLC frame with the one-octet length called *basic mode* or the optional modes for a two-octet length called *extended mode*, and the four-octet length called *super mode*. Figure 7-6 shows the 8-bit versions of the three HDLC control field formats: information, supervisory, and unnumbered frames. The unnumbered frame control field format is only 8 bits long for all control field formats. Note how the first two bits of the control field uniquely identify the type of frame: *information*, *supervisory*, or *unnumbered*.

The *information* frame transports user data between DTE and DCE. Within this frame, the N(S) and N(R) fields designate the sequence number of the last frame sent and the expected sequence number of the next frame received, respectively. HDLC also defines 16-bit versions of the information, supervisory, and unnumbered frames, the difference being in the size of the sequence number fields. Information frames always code the Poll (P) bit to a value of 1, as indicated in Figure 7-6. In supervisory and unnumbered formats, the Poll/Final (P/F) bit indicates commands and responses. For example, the DTE (or DCE) sets the P/F bit to 1 to solicit (i.e., poll) a response from the DCE (or DTE). When the DCE (or DTE) responds, it sets the P/F bit to zero to indicate that its response is complete (i.e., final).

The *supervisory* frame uses the Supervisory (S) code bits to acknowledge the receipt of frames, request retransmission, or request temporary suspension of information frame transfer. It performs these functions using the P/F bit in the following command and response pairs: Receive Ready (RR), Receive Not Ready (RNR), REject (REJ), and Selective REject (SREJ).

The *unnumbered* frame uses the Modifier (M) bits of the unnumbered format to provide the means for the DTE and DCE to set up and acknowledge the HDLC mode, and to terminate the data link layer connection. The HDLC standard defines a variety of control



messages to set up the HDLC mode discussed in the preceding section (e.g., NRM, ARM, and ABM). LAP-B uses only the asynchronous balanced mode (ABM).

The basic difference between the one-, two-, four-, and eight-octet control field formats is the length of the send and receive sequence number fields, N(S) and N(R)⁹ respectively. The HDLC standard defines the modulus as the maximum decimal value of these sequence number fields, as given in Table 7-1. In other words, HDLC stations increment the sequence number modulo the modulus value given in the table. For example, for a one-octet control field, stations increment the sequence numbers modulo 8; specifically, the stations generate the following pattern of sequence numbers: 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, and so on. The 16-bit control field initially targeted use over long-delay satellite links to increase application throughput. A larger sequence number improves performance because the sender can transmit up to the modulus of the sequence number without receipt of an acknowledgment. The 32- and 64-bit versions of the control field were developed for similar reasons as the bandwidth-delay product increased with higher-speed transmission links, such as those used in modern local area and wide area networks.

Point-to-point physical X.25 network access supports either a single link or multiple links. The LAP-B Single Link Procedure (SLP) supports data interchange over a single physical circuit between a DTE with address "A" and a DCE with address "B." The coding for the one-octet address field for the address "A" is a binary 1100 0000 and the coding for address "B" is a binary 1000 0000. The address field identifies a frame as

Control Field Length (Octets)	Sequence Number Length (Bits)	Sequence Number Modulus
1	3	8
2	7	128
4	15	32,768
8	31	2,147,483,648

Table 7-1. HDLC Control Field Lengths and Sequence Number Modulus

either a command or a response, since command frames contain the address of the other end, while response frames contain the address of the sender. Information frames are always coded as commands in the address field.

The optional Multilink Procedure (MLP) exists as an upper sublayer in the data link layer. Multilink operation uses the single-link procedures independently over each physical circuit, with the multilink procedure providing the appearance of a single data flow over two or more parallel LAP-B data links. MLP has several applications in real-world networks. It allows multiple links to be combined to yield a higher-speed connection; it provides for graceful degradation if any single link should fail; and, finally, it allows a network designer to gracefully increase or decrease capacity without interrupting service. The X.25 MLP design philosophy appears in inverse multiplexing in the TDM world, the Frame Relay (FRF.15 and FRF.16.1) and PPP multilink standards, as well as Inverse Multiplexing over ATM (IMA), as described in Chapter 12.

Comparison of Link Access Procedure (LAP) Protocols

The ITU-T defines three types of Link Access Procedure (LAP) protocols. LAP, the first ISDN protocol, was based on the HDLC Set Asynchronous Response Mode (SARM) command used in “unbalanced” connections. This mode formed the basis for Link Access Procedure Balanced (LAP-B), an HDLC implementation that uses balanced asynchronous mode with error recovery to form the basis of the X.25 packet-switching protocol. The LAP-B protocol is identical in format to an 8-, 16-, or 32-bit control field HDLC frame as described in the next section. The next extension of HDLC and LAP was Link Access Protocol over D-channel (LAP-D) standardized by the ITU-T in Recommendations Q.920 and Q.921 as the Digital Subscriber Signaling System number 1 (DSS1) data link layer. This implementation of HDLC uses either the basic or extended asynchronous “balanced” mode configuration and provides the basis for both ISDN and Frame Relay services. In the late 1980s, the ITU-T removed the sequence numbering, windowing, and retransmission functions for a Frame Relay service in the Q.922 standard, resulting in a protocol dubbed LAPF that combined the address and control fields. Note that the

removal of the control field in Frame Relay eliminated the sequence numbers used to implement retransmission of lost frames in LAP-B and LAP-D. Instead, higher-level protocols, for example TCP, must perform error detection and retransmission when operating over Frame Relay. As we shall see, a close family relationship exists between other link access procedures employed by X.25 (LAP-B), ISDN (LAP-D), and Frame Relay (LAP-F), as illustrated in Figure 7-7.

Packet Layer Format and Protocol

The X.25 Packet Layer Protocol (PLP) is at layer 3 of the OSI architecture and is primarily concerned with network routing functions in public and private packet networks. The protocol provides a standard layer 3 networking interface between a subscriber or logical DTE, and the network entry point called either the data switching exchange (DSE) or logical DCE. Each X.25 packet transferred across the DTE/DCE interface exists within a basic LAP-B frame, as shown in Figure 7-8. Note that the X.25 layer 3 packet, including packet header and packet data, forms the user data (or information) field of the layer 2 LAP-B frame.

An X.25 packet has a header and a user data field, as shown in Figure 7-8. The Qualifier (Q) bit allows a transport layer protocol to separate control data from user data. The D bit is used in delivery confirmation during X.25 switched virtual call setup. The next two bits indicate the packet type, with 01 indicating a data packet with a three-octet header. A four-octet header is also standardized. The X.25 packet layer address has a 4-bit group number and an 8-bit logical channel number, together forming a 12-bit logical channel number (LCN). Channel zero is reserved, and therefore there can be up to 2^{16} minus 1, or 4095, logical channels on a physical circuit carrying the X.25 protocol.

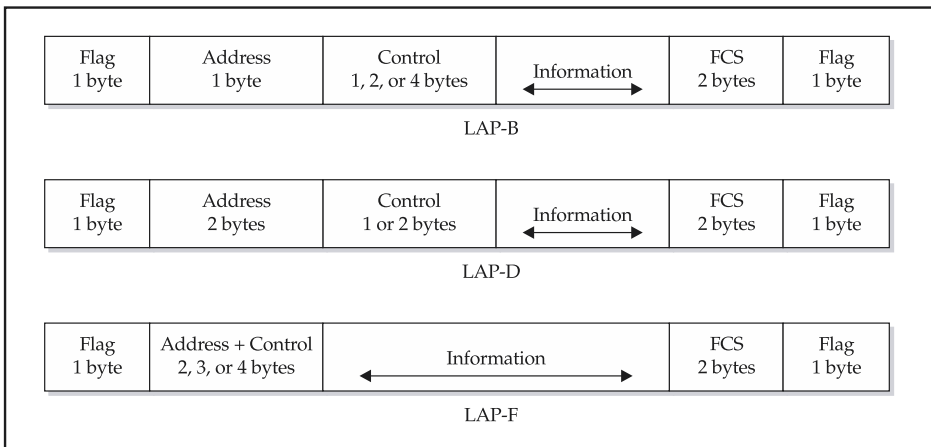
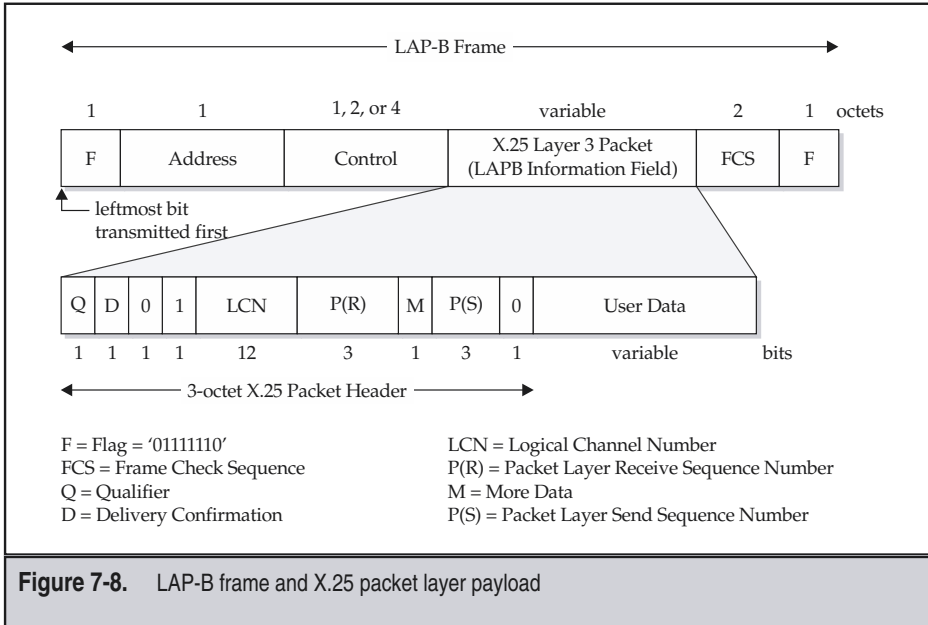


Figure 7-7. Comparison of X.25, ISDN, and Frame Relay information-frame formats



By convention, logical channel numbers (LCNs) are assigned to each of the incoming and outgoing virtual calls for each DCE and DTE, respectively, as well as to all PVCs. Out of the 4095 logical channel numbers available per physical circuit, PVCs are assigned the lowest numbers, followed by one-way incoming virtual calls, then two-way incoming and outgoing calls, with the highest numbers reserved for one-way outgoing virtual calls. Note that LCNs hold only local significance to a specific physical port but must be mapped to a remote LCN for each virtual call. X.25 packet networks use search algorithms to resolve collisions and assign LCNs to each virtual call.

The packet layer uses the receive and send sequence numbers (P(R) and P(S)) to support a packet layer flow control protocol described later in this section. The More (M) bit supports segmentation and reassembly by identifying the first and intermediate packet segments with a value of 1, with the last segment having a value of zero.

Two types of services are defined in the X.25 standard: virtual circuit and datagram. Virtual circuits assure sequence integrity in the delivery of user data, established either administratively as a PVC, or as a switched virtual call (VC) through call control procedures. PVCs are permanently established between a source and destination DTE pair. Datagrams do not require call control procedures and either are sent in a best-effort mode or request explicit receipt notification.

Control Functions

Call control packets defined in X.25 support switched virtual calls (VCs). X.25 VCs use either the X.121 or the E.164 addressing format. VCs act much like telephone calls where a source must first connect to the destination node before transferring data. Therefore, one source can connect to different destinations at different times, as compared with a PVC that is always connected to the same destination. Figure 7-9 shows a typical control packet sequence for the establishment of an X.25 VC, followed by the call clearing procedure. Note that the data transfer stage occurs only after successful call establishment. An issue with VCs is that the X.25 network may become so busy that it blocks connection attempts. Applications that cannot tolerate occasional periods of call blocking should use dedicated PVCs.

Example of X.25 Operation

The LAP-B protocol uses a store-and-forward approach to ensure reliable delivery of packets across noisy, error-prone transmission links. The example of Figure 7-10 illustrates the store-and-forward approach for recovering from errors between two packet-switching

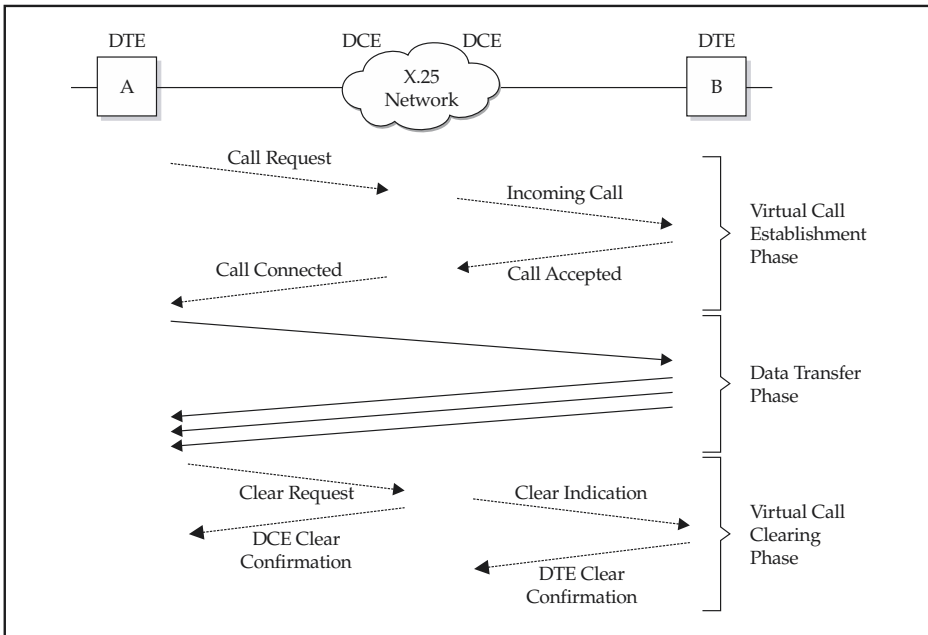
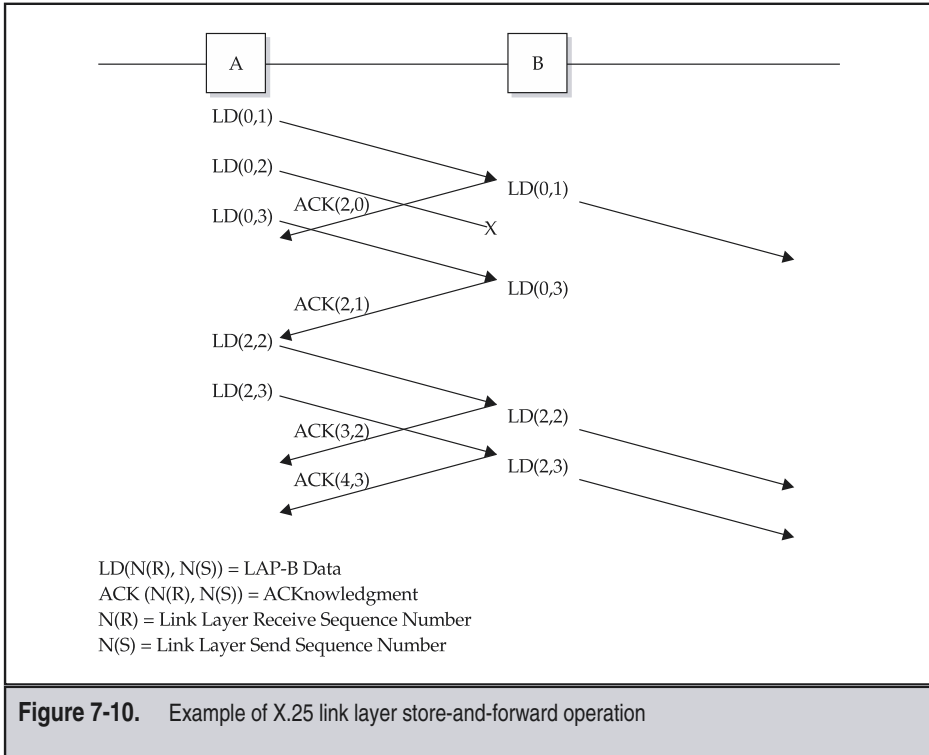


Figure 7-9. X.25 virtual call establishment, packet transfer, and clearing example



nodes, labeled A and B, using the X.25 link layer protocol. Two types of packets are exchanged between the nodes: LAP-B Data (LD) and LAPB acknowledgments (ACK). In the example, separate link layer control packets (e.g., Receiver Ready (RR)) carry the link layer acknowledgments; however, in actual operation, switches often piggyback acknowledgments onto packets heading in the opposite direction. Each LAP-B frame has a pair of link layer sequence numbers: N(R), the receive sequence number, and N(S), the send sequence number. N(S) indicates the sequence number of the transmitted packet. N(R) indicates the value that this receiver expects to see in the send sequence number (N(S)) of its peer. Therefore, the receive sequence number, N(R), acts as a cumulative acknowledgment of all link layer frames with sequence numbers up to the value N(R) minus 1.

The example begins with node A sending user data with N(S) = 1, which is successfully transferred to node B, which acknowledges its receipt with an ACK containing N(R) = 2, indicating that the next expected value of N(S) is 2. Node B now stores packet 1 and attempts to forward this packet to the next node. Meanwhile, node A sent the next packet with N(S) = 2; however, node B detected errors and discarded it as indicated by the X in Figure 7-10. B then receives the third packet sent by A with N(S) = 3, but it is out of

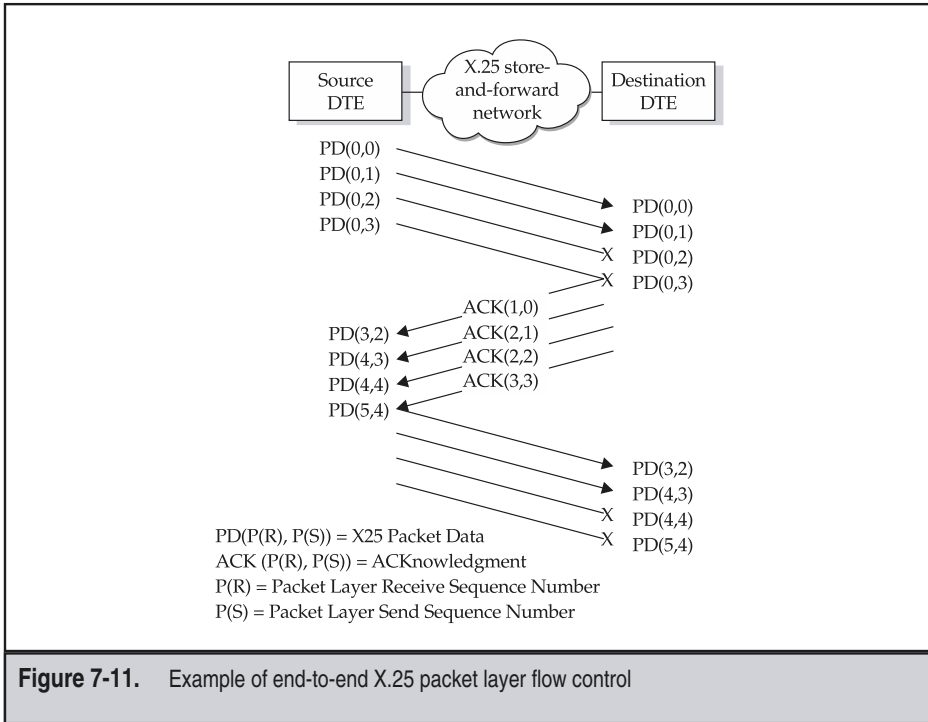
order. B acknowledges this with a link layer RR control packet containing $N(R) = 2$, indicating that it expects the next packet to have a send sequence number of $N(S) = 2$. Node A responds by resending packets with $N(S)$ equal to 2 and 3, which are successfully received and acknowledged by node B. Node B can now attempt to forward these packets to the next node using the same process, but probably with different link layer sequence numbers. The example also illustrates the link layer acknowledgment of the control packets received from B by A.

This simple example illustrates the complexity involved in store-and-forward packet switching. Note that not only do acknowledgments occur on every link as shown in this example, but also they occur again at the end-to-end network layer, as described later in this section. This processing was worthwhile when a non-negligible fraction of the packets experienced transmission errors on the noisy facilities, a situation prevalent in the analog and radio transmission facilities of 1970s and 1980s wide area networks. Creative engineers developed even more sophisticated schemes in which only the errored packets were retransmitted using the Selective Reject (SREJ) control protocol. SREJ significantly improves performance over networks such as ATM with long bandwidth-delay products, as analyzed in Chapter 25. These schemes became the basis of protocols used in B-ISDN that we study in the next part. However, since the ubiquitous deployment of fiber, many of these error-correcting capabilities can now be done at the end points only, instead of at every intermediate node—as is done in Frame Relay, for example. Furthermore, as we shall see later, other higher-layer protocols also perform error detection and retransmission, which obviates the need to repeat the functions at a lower layer. For example, as described in Chapter 8, the Transmission Control Protocol (TCP) commonly used on the Internet also performs error detection and retransmission.

Traffic and Congestion Control Aspects of X.25

The send and receive sequence numbers in the X.25 packet layer provide flow control between the packet layer source and destination end systems (or DTEs). Figure 7-11 illustrates a simple example of this end-to-end X.25 packet layer flow control between source and destination DTEs connected by an X.25 store-and-forward network. The packet layer send sequence number, $P(S)$, is a sequential number for the current packet incremented modulo the packet header sequence number modulus. X.25 defines packet layer sequence number modulus values of 8, 128, and 32,768. The destination uses the packet layer receive sequence number, $P(R)$, in the ACKnowledgment to indicate the packet layer send sequence number expected in the next packet from the other end point of that virtual circuit. Therefore, the packet layer receive sequence number, $P(R)$, acts as an acknowledgment for all packets up to $P(R)$ minus 1. The reader should note that these packet layer sequence numbers operate on an end-to-end basis, as opposed to a link-by-link basis for LAP-B frames.

X.25 defines a separate value of the window size at the transmitter that controls the maximum number of packets it can send without receiving a packet layer acknowledgment from the destination. Similarly, X.25 defines a window size at the receiver that controls how many packets the DTE will accept prior to generating an acknowledgment. Of



course, the window size must always be smaller than the packet layer sequence number modulus. The example of Figure 7-11 employs a window size of 2 at the receiver and a window size of 4 at the transmitter.

As shown in the example in Figure 7-11, the source transmits four packets and then waits for the packet-level acknowledgment (for example, in a packet-level RR packet) before sending additional packets. The destination is slow to acknowledge these packets; and since its window size is only 2, it only acknowledges the first two packets received. The destination indicates that it still expects to receive $P(S) = 2$ for the third and fourth packets. The source retransmits the third and fourth packets, along with the fifth and sixth as the process repeats. Normally, the source and destination would coordinate transmit and receive window sizes; however, they don't need to do so. The example in Figure 7-11 also illustrates the packet layer acknowledgment of the control packets between the source and the destination.

The generic name for this procedure is a *sliding window* flow control protocol. This procedure allows the transmitter or the receiver to control the maximum rate of transmission over a virtual circuit, and is therefore a form of traffic control. This is still an essential

function for a slow receiver (such as a printer) to control a fast transmitter (a computer) in many data communications applications today. The receive sequence number acknowledgment can be “piggybacked” in the packet header for a packet headed in the opposite direction on a virtual circuit, or it may be sent in a separate layer 3 control packet.

Service Aspects of X.25

X.25 packet switching serves many user communities—especially in Europe, where it still constitutes a significant portion of public and private data services. X.25 traffic levels have remained relatively flat after the introduction of Frame Relay services absorbed much of the existing public packet-switching communications market growth and now are entering a period of decline. Packet switching remains a popular technology, however, and will likely continue to be used globally well into the twenty-first century to reach remote areas of the world or act as a backup to other networks.

Recently, due to the ever-increasing WAN bandwidth requirements of computing, X25 networking speeds have also increased. Trunk speeds have increased beyond the 300 to 1200 bps access and 56 Kbps trunks of the early X.25 networks. Now, many packet switches provide access at 56 Kbps with trunks at DS1/E1 speeds.

FRAME RELAY—OVERVIEW AND USER PLANE

Frame Relay led the way in a minimalist trend in data communications, essentially being X.25 on a diet. Proof of Frame Relay’s profound importance in data networking is the fact that public Frame Relay services have displaced many private line-based data networks. Today, almost every WAN manufacturer and data communications service provider supports the Frame Relay protocol. This section presents an overview of Frame Relay, highlighting the control and user plane concepts. Frame Relay operates as an interface and as a network service in the user plane, as described in this section. The next section describes the control plane operation of Frame Relay, which signals status for permanent connections and establishes switched connections.

Origins of Frame Relay

X.25 packet switching, proprietary private networks built upon private lines, and legacy networks running HDLC and SDLC dominated the data communications marketplace from 1980 through the early part of the 1990s. In order to keep pace with the increased bandwidth and connectivity requirements of today’s applications, users needed a new data communications technology to provide higher throughput at a lower cost. Frame Relay responded to this need, beginning in the early 1990s to provide higher-bandwidth and more cost-effective transfer of packet data. Frame Relay did this by eliminating the overhead of the network layer present in X.25, as well as reducing the complexity of the link layer protocol. Frame Relay is a simplified form of X.25 packet switching and is generally considered its replacement.

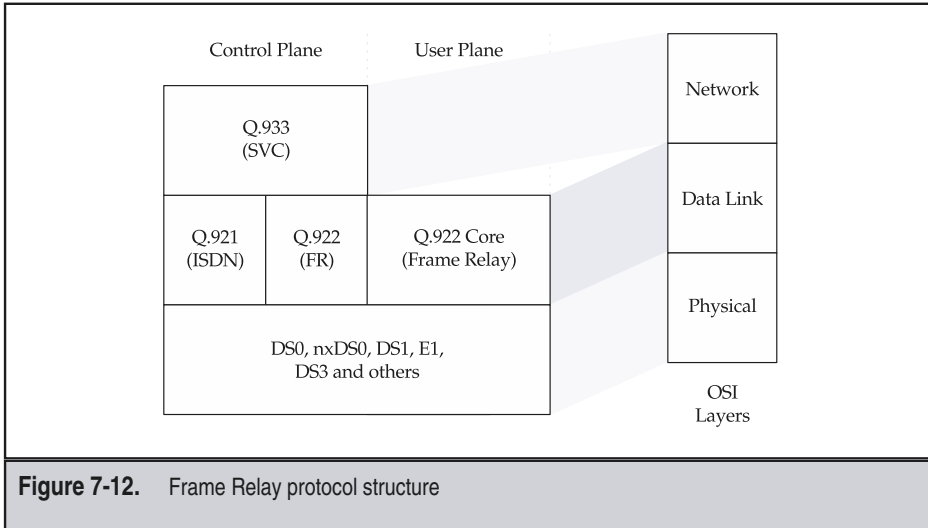
The biggest difference between X.25 and Frame Relay is that X.25 does error detection and retransmission and flow control on a link-by-link as well as end-to-end level, while Frame Relay does none of this. As such, X.25 requires much more processing power and increases delay when compared with Frame Relay. The elimination of retransmission and flow control allows Frame Relay implementations to operate cost effectively at much higher speeds. The nearly ubiquitous digital fiber optic transmission facilities made lost packets relatively rare—hence applications networking across a slimmed-down Frame Relay protocol could afford to occasionally retransmit lost packets. While Frame Relay has its origins in HDLC and ISDN link layer protocols, its streamlined protocol eliminated the overhead involved in error-correction and flow-control overhead. Higher-layer protocols (e.g., TCP) recover lost or corrupted data when operating over Frame Relay networks.

Not only does Frame Relay offer the security of private lines; it also provides greater granularity of bandwidth allocation. Since Frame Relay is a lighter-weight protocol, it runs at higher speeds than X.25. Indeed, for software-based implementations that run both Frame Relay and X.25, the Frame Relay protocol runs much faster. Frame Relay provides an upgrade to packet-switch technology by supporting speeds ranging from 64 Kbps through nxDS1/nxE1, and all the way up to SONET/SDH speeds. The physical layer interface support is now all compiled together into FRF.14. For lower rates of SONET/SDH, the signal must use the virtual tributary (VT) mapping of the signals up to T3/E3 rates. Frame Relay fills a technology gap between X.25 and ATM, and at the same time also can provide a smooth transition to ATM. Frame Relay standards for interworking with ATM are now well established and will be discussed in Chapter 17. The standards bodies and vendors are working on providing Frame Relay interworking with MPLS networks as well.

Frame Relay standards derived from the ISDN Link Access Procedure for the D-channel (LAP-D or, as later modifications were called, LAP-F), as described in ITU-T/CCITT Recommendation Q.921, which led to I.122, I.233, I.370, and Q.922. The signaling procedures in the ISDN control plane specified in ITU-T Recommendation Q.931 led to the Frame Relay signaling standards defined in Recommendation Q.933, as described later. The corresponding ANSI standards are T1.617 and T1.618, which standardized the service description, protocol, and status signaling for Frame Relay. The many advantages offered by Frame Relay also caused the formation of a separate industry group, called the Frame Relay Forum (FRF), whose charter is to develop Implementation Agreements (IAs) to facilitate interoperability between different vendor products and services. Recognizing that Frame Relay is primarily used today as an access protocol, rather than as a bearer service within ISDN, the ITU-T recently created two new standards. Recommendation X.36 defines the Frame Relay User-Network Interface (UNI), and Recommendation X.76 defines the Network-to-Network Interface (NNI). All future ITU-T standards work will be based upon these recommendations instead of the I-series and Q-series recommendations cited previously.

Frame Relay Protocol Structure

The Frame Relay protocol has two logically separate components: the control plane (C-plane) and the user plane (U-plane). The concept of control and user planes is a practice begun by the ITU-T for ISDN, as described in Chapter 6. Figure 7-12 illustrates the

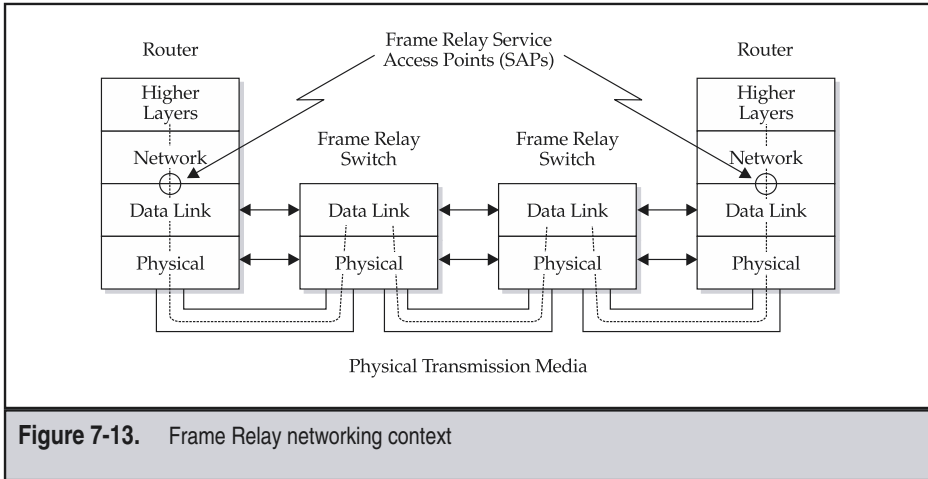


structure of the protocols that support Frame Relay. The user plane of Frame Relay implements a subset of the OSIRM data link layer functions as specified in ITU-T Recommendation Q.922. Frame Relay also has a control plane involved with reporting on the status of PVCs and the establishment of SVCs. Recommendation Q.933 defines the status signaling for PVCs and the call control procedures for SVCs. The Frame Relay Q.933 signaling protocol may either operate over the Frame Relay protocol (Q.922) directly or be signaled separately via the ISDN protocol (Q.921) on a separate TDM channel. The ITU-T signaling standards for ISDN (Q.931), Frame Relay (Q.933), and B-ISDN (Q.2931) have a common philosophy, message structure, and approach.

Frame Relay Networking Context

Figure 7-13 shows how the user plane Frame Relay protocol layers operate in a network context interconnecting two end systems, for example, two routers of a virtual private network connected by a public Frame Relay service.

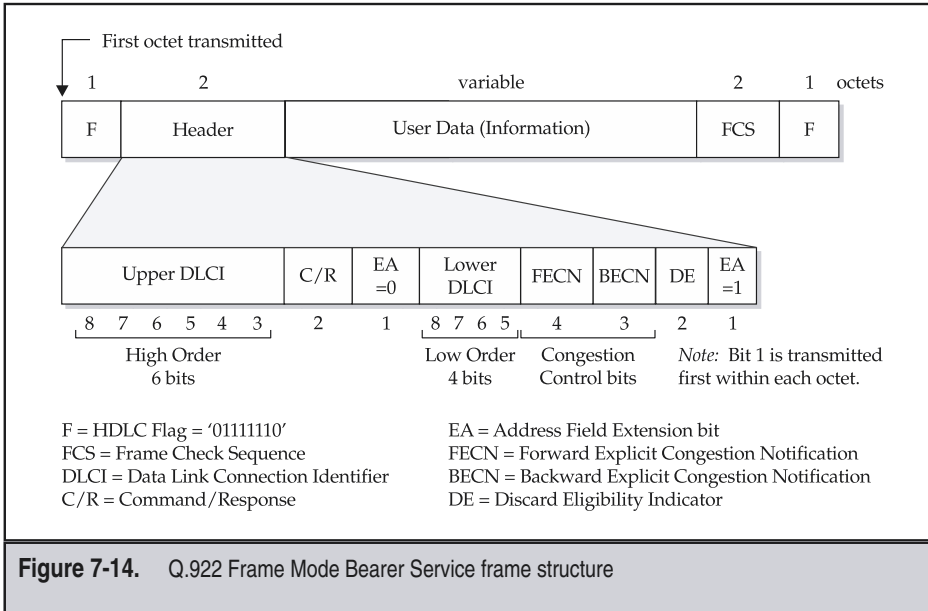
In the example, two interconnected intermediate switches relay frames between service access points (SAPs) in the end systems. Multiple permanent or switched virtual connections (PVCs or SVCs) may exist on a single physical access circuit. As shown in the figure, Frame Relay operates at the link layer only, which reduces complexity and usually allows for higher-speed operation, but at a certain price. Frame Relay service usually provides in-sequence delivery of frames. Since Frame Relay has no sequencing and no retransmission to detect misordered or lost frames, higher-layer protocols, such as TCP, must perform retransmission and resequencing.



ITU-T Recommendation Q.922 defines the terms and basic concepts of a frame mode bearer service, called Link Access Procedure F (LAPF). The Service Access Point (SAP) is the logical-level data link interface between the data link layer and the network layer, as indicated in Figure 7-13. Each SAP corresponds to a data link connection end point, uniquely determined via data link connection identifiers (DLCIs) in the frame header. The Frame Relay protocol also operates in the control plane, which actually runs separate protocols between Frame Relay users and networks, as well as between networks, as described later.

Frame Format

The Link Access Procedure for Frame Relaying (LAP-F) format used by Frame Relay services is a derivative of the ISDN Link Access Protocol D-channel (LAP-D) framing structure. Flags indicate the beginning and end of the frame. Three primary components make up the Frame Relay frame: the header and address area, the user-data portion, and the frame check sequence (FCS) field. Figure 7-14 shows the basic Frame Relay frame structure with the two-octet header and address field from ITU-T Recommendation Q.922, with 10 bits representing the data link connection identifier (DLCI), and 6 bits of fields related to other functions, detailed in the text that follows. The first and last one-octet flag fields, labeled F, are HDLC flags. HDLC bit stuffing, identical to that described in the previous section on X.25, is performed to avoid mistaking user data for an HDLC flag. Although HDLC supports a user data field of up to 8188 octets, Frame Relay standards specify smaller frame sizes. For example, FRF.1.2 states that all implementations must



support a maximum frame size of at least 1600 octets for transport of encapsulated Ethernet traffic. The frame check sequence (FCS) field is two octets long as defined in the HDLC standard.

The reader should note that the X.25 and FR standards use a different notation for the order of bit transmission. In the X.25 standards, bit transmission order was from left to right, as indicated by the arrows in the figures and the bit position numbering in the previous section. In Frame Relay, the bits are grouped into octets, which are transmitted in ascending (i.e., left to right) order. For each octet, bit 1, which is the least significant bit, is transmitted first; and bit 8, which is the most significant bit, is transmitted last. There is no right or wrong way to indicate bit order transmission, just as different countries around the world select different sides of the road to drive on.

As shown in Figure 7-14, the Frame Relay header contains upper and lower Data Link Connection Identifier (DLCI) fields, together forming a 10-bit DLCI that identifies up to 1024 virtual circuits per interface in the two-octet header format. The DLCI has only *local significance* on a physical interface. This means that the DLCIs may differ on the interfaces at each end of the point-to-point Frame Relay VC. On any interface, each user CPE device has a separate DLCI for each destination. This limits the size of a fully meshed Frame Relay network to approximately 1000 nodes. Larger Frame Relay networks require the three- or four-octet header fields, or else a hierarchical, partial-mesh topology.

The standards require that networks transparently convey the Command/Response (C/R) between Frame Relay users. Hence, the C/R bit can be employed by user applications.

The Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion Notification (BECN) bits indicate to the receiver and sender, respectively, the presence of congestion in the network. Specifically, the network sets the FECN indicator in frames traversing the network from sender to receiver that encounter congestion. Receiver-based flow control protocols, such as DECnet, use FECN to initiate congestion avoidance procedures. The network sets the BECN indicator in frames traversing the network from receiver to sender for congestion occurring in the sender-to-receiver direction. That is, the network sets the BECN indicator in frames traveling in the opposite direction on the same VC to those in which it sets the FECN indicator. Therefore, BECN aids senders, who can then dynamically change their source transmission rate.

The Discard Eligibility (DE) bit, when set to 1, indicates that during congested conditions, the network should discard this frame in preference to other frames with a higher priority—for example, those with the DE bit set at 0. Note that networks are not constrained to discard only frames with DE set to 1 during periods of congestion. Either the user or the network may set the DE bit. The network sets the DE bit when the received frame rate exceeds the committed information rate (CIR) specified for a particular VC. Users rarely set the DE bit, since there is rarely an advantage to doing so.

The address field extension (EA) bit is the first bit transmitted in each octet of the Frame Relay address field. When set to 0, it indicates that another octet of the address field follows. When set to 1, it indicates that the current octet is the last octet of the address field. As an example, Figure 7-14 illustrates the basic two-octet frame format where the first EA bit is set to 0, and the second EA bit is set to 1. Recommendation Q.922 defines how the EA bits extend the DLCI addressing range to three- and four-octet formats. The Frame Relay Service Specific Convergence Sublayer (FR-SSCS) defined in ITU-T Recommendation I.365.1 is identical to the Frame Relay frame without FCS, flags, and HDLC zero insertion. Chapter 17 summarizes FR-SSCS and presents the three- and four-octet Frame Relay header formats.

Frame Relay Functions

A primary use of Frame Relay is as a user interface to a public data service, or as an interface between networks. Frame Relay supports permanent virtual connections (PVCs) and switched virtual connections (SVCs). Frame Relay virtual connections (VCs) are either point-to-point or multipoint-to-multipoint (also called multicast), as defined in Chapter 4. SVCs use a call establishment protocol similar to that employed by X.25, ISDN, and ATM. PVCs are managed by a status signaling protocol.

Example of Frame Relay Operation

Figure 7-15 illustrates a key aspect of the simplification that Frame Relay provides. End system A on the left-hand side of the figure transmits frames to a Frame Relay network, which relays the frames to a destination end system B. There are no sequence numbers in the frames. The numbers in the example are for illustrative purposes only. The Frame Relay network simply relays frames in the same order on the interface to destination B. If a frame is corrupted while in transit, the Frame Relay network simply discards or drops it. A Frame Relay network may also discard frames during intervals of congestion. In our example, the Frame Relay network never delivers the fourth frame because of either errors or discard due to congestion. This simplified relaying-only protocol allows vendors to build simple, fast switches; however, it requires more intelligence in the higher-layer protocols—such as TCP residing in the end systems—to recover lost frames.

Frames can be lost due to transmission errors or congestion. When Frame Relay operates over modern transmission media, such as fiber optic systems, frames lost due to errors become a rare occurrence. Loss then occurs primarily due to congestion on the trunks within the Frame Relay network or on the access line to the destination. Since loss occurs infrequently in well-managed Frame Relay networks, the end systems seldom need to invoke error recovery procedures.

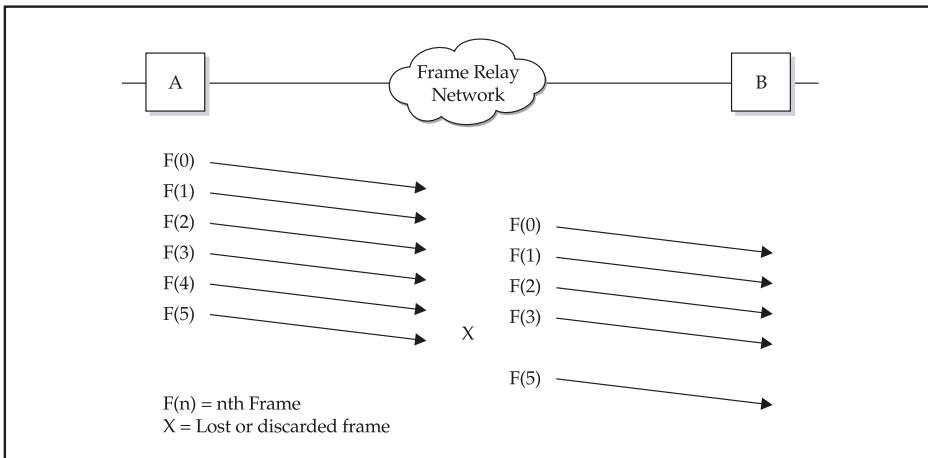


Figure 7-15. Frame Relay interface, switching, and network service

Traffic and Congestion Control Aspects of Frame Relay

This section covers traffic and congestion control functions implemented in Frame Relay networks. These functions are simpler than their ATM and MPLS counterparts, and hence provide a good introduction and background to the important concepts of policing, congestion indication, and reaction to congestion, as detailed in Part 5.

Frame Relay Traffic Control

Frame Relay defines a committed information rate (CIR), which is the information transfer rate the network commits to transfer under normal conditions [ITU I.370]. However, the term "committed" is somewhat ambiguous in that it does not specifically state the frame loss or delay objective. Transmission errors can cause frames to be lost, and finite buffer space in networks that implement statistical multiplexing can result in lost frames due to momentary overloads. The Frame Relay network sets a CIR for each PVC or SVC.

We describe the Frame Relay traffic control procedures with reference to the diagram depicted in Figure 7-16, which plots the total number of bits transmitted on the vertical

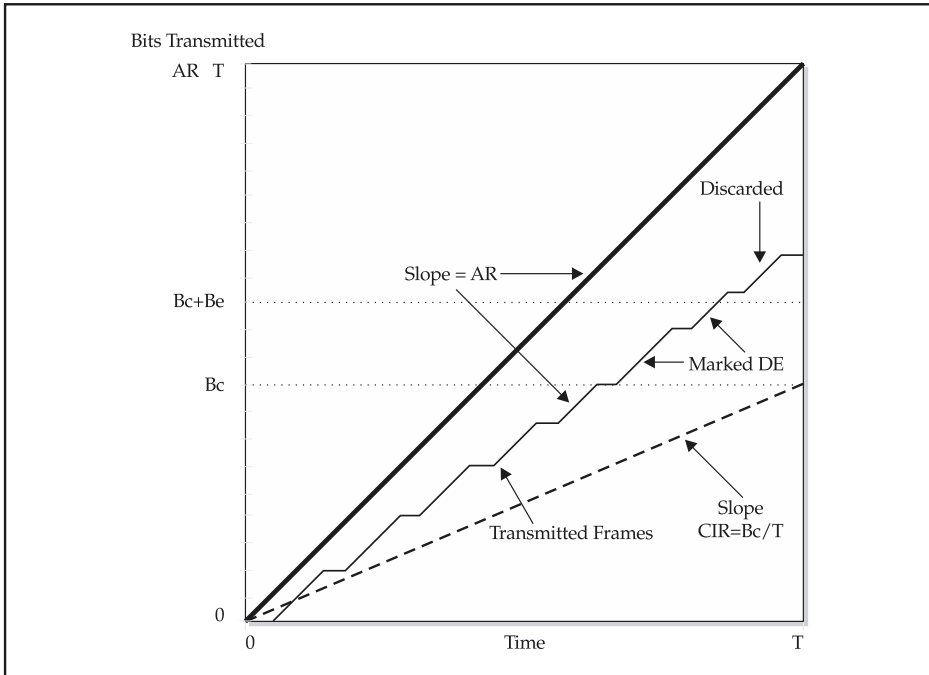


Figure 7-16. Example of Frame Relay traffic control operation

axis versus time on the horizontal axis. The maximum number of bits transmitted per unit time is, of course, limited by the access line rate, AR, as shown by the thick line in Figure 17-16. The CIR is the number of bits in a committed burst size, Bc, that can arrive during a measurement interval T such that $CIR = Bc / T$. The dashed line in the figure has a slope equal to CIR. Frames are sent at the access line rate, indicated by the trace labeled “transmitted frames” in the figure. If the number of bits that arrive during the interval T exceeds Bc but is less than an excess threshold, Bc + Be, then the frames associated with those bits are marked as Discard Eligible (DE). The bits that arrive during the interval T in excess of Bc + Be are discarded by the access node. To avoid discards at the ingress node for a given access rate, AR, the value of Be must be greater than or equal to $AR \times T - Bc$.

At present, there is no uniform method for setting the measurement interval T. If T is set too small, such that Bc is less than the length of a single frame, then every frame will be marked DE. If T is set too large, the buffer capacity in the FR network required to guarantee CIR may not be practical. Setting T to a value on the order of the maximum round-trip delay is a good guideline to achieve good TCP/IP throughput over Frame Relay. Coast-to-coast round-trip delays across the continental United States are typically on the order of one hundred milliseconds across most Frame Relay networks.

Network providers offer different interpretations of the committed information rate (CIR) and discard eligible (DE) functions according to different pricing plans. For example, CIR may be available only in certain increments, or DE traffic may be billed at a substantial discount. Customer network management, performance reporting, and usage-based billing are value-added services offered by some providers. It pays to shop around carefully. To address this general issue of differing interpretations and use of traffic variables, the Frame Relay Forum issued a service-level definition Implementation Agreement at the end of 1998 to help service providers, vendors, and end users better describe and evaluate the Frame Relay offerings. Even though this was a step in the right direction, there were still no standard procedures that could be used to verify that the claims for a specific service really delivered what was expected in terms of the set of traffic parameters' theoretical behavior. Standard verification procedures have finally been developed and are described in FRF.19. We describe some details of this agreement later. Because of the relative implementation complexity of these agreements to verify traffic contracts, it is likely that it will take Frame Relay service providers some time to deploy these systems.

There are two different philosophies in the setting of the CIR. We call the first *full-rate allocation*, where the sum of the CIRs associated with the PVCs on an interface is no more than the interface speed, and the second is called *oversubscription* (or *overbooking*), where the sum of CIRs exceeds the interface rate. The interface may be between the Frame Relay user and the network or an internal trunk interface. In full-rate allocation, the fact that the sum of the Frame Relay connection's CIR values across an interface does not exceed the actual bandwidth guarantees predictable, deterministic performance at the expense of lower utilization because all users rarely utilize their entire CIR.

In the oversubscription case, loss performance becomes statistical; however, individual PVCs may transmit more in excess of the CIR rate than in the regular booking case, since the sum of the PVC CIRs exceeds the interface rate. The oversubscription paradigm

relies on the averaging of a larger number of smaller connections in a technique called *statistical multiplexing*, as described in Chapter 24. Typically, oversubscription achieves higher utilization at the expense of a higher loss rate. For properly engineered networks, the frame loss level can be controlled through proper parameter settings. Some Frame Relay networks that use closed loop congestion control algorithms automatically control parameters to reduce loss by allowing less data into the network during congested intervals. However, the setting of the congestion control algorithm parameters is important in achieving acceptable loss and fairness.

Frame Relay Congestion Control

The Discard Eligible (DE) bit may be set by either the customer or the network. DE is rarely set by the customer; rather, the network provider ingress switches set DE for frames exceeding the CIR. If a network node becomes congested, it should first discard the frames with the DE bit set. Beware, however, of the danger of discarding frames marked with DE during long periods of congestion: the applications may react by retransmitting lost frames, intensifying congestion, and possibly resulting in a phenomenon called *congestion collapse*, as described in Chapter 23.

Congestion notification is provided in the Frame Relay header field by the FECN and BECN bits. Nodes in the Frame Relay network set the FECN bit when they become congested. The FECN bit informs receiver flow controlled protocols of the congestion situation or allows the receiver or the egress node to set the BECN bit. The BECN bit is set in FR frames headed in the upstream direction to inform transmitter flow controlled protocols of the congestion situation. An increase in the frequency of FECN and BECN bits received is a good indication of network congestion. At present, little use is being made of this technique by higher-layer protocols in end and intermediate systems. One concern is that by the time the FECN/BECN arrives at the controlling end, the congested state may no longer exist in the FR node that sent the FECN/BECN notice. The main concern is that the FECN/BECN notice is delivered to the CPE router, which is not the primary source of flow control. Currently, no technique exists for the CPE router to convey the FECN/BECN message to TCP or the application that could provide flow control.

Figure 7-17 depicts a network of Frame Relay switches in Houston, Atlanta, Chicago, and Raleigh connecting a host device in Dallas with a LAN in Charleston via a pair of routers. In this example, the host in Dallas accesses the router via Frame Relay and is capable of dynamically controlling its transmission rate. In the process of downloading a large volume of files from the host in Dallas to the users on a local area network in Charleston via a PVC (shown in Figure 7-17 as a dashed line), congestion occurs in the output buffers on the Atlanta switch on the physical link to the Raleigh switch, as indicated in the figure. The Atlanta switch sets the FECN and BECN bits on all DLCIs traversing the Atlanta-to-Raleigh trunk to notify the senders and receivers of the congestion condition. The Atlanta node sets the FECN bit to 1 and notifies the router in Charleston using the PVC receiving traffic from Dallas of impending congestion. The Atlanta node also sets the BECN bit to 1 on frames destined from Charleston to Dallas, informing the router in Dallas using the PVC of the same congestion condition. Either the Dallas user could throttle back, or the Charleston

Relay service provider network interconnections exist today because the interconnection point is sometimes a single point of failure.

Service providers could implement a Frame Relay switched PVC (SPVC) to provide a standard resilient network-to-network interconnection, as defined in ITU-T Recommendation X.76 and FRF.10.1. This concept is very similar to the Soft PVC concept used in ATM's Private Network-to-Network Interface (PNNI), as discussed in Chapter 15, but the dynamic nature of such implementations, together with the exposure of the interconnected networks, presents issues that limit its use primarily to a single service provider. Service providers are offering vendor-specific Frame Relay services that provide additional features, such as prioritization, automatic restoration, closed-loop congestion control, and usage-based billing. Several carriers provide international Frame Relay services in a number of areas in Europe, Asia, and Australia. Frame Relay is still a very popular and mature service, and there is also a great deal of active standards work providing continued enhanced Frame Relay service and network functionality. In addition to producing numerous new Frame Relay forum (FRF) interoperability agreements, practically all previously released work has been revised and updated to keep pace with emerging technology and customer demands.

Frame Relay provides some basic public network security in that data originating and terminating through an access line is limited to connectivity established by the PVCs for that access line. Indeed, virtual private networks (VPNs) are a key application for Frame Relay networking. Figure 7-18 illustrates a Frame Relay network configuration common to

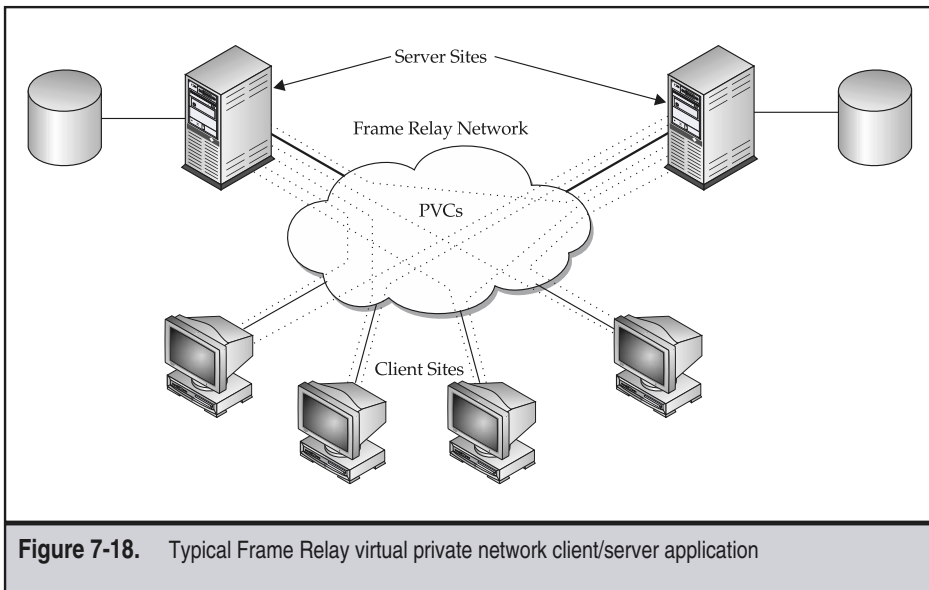


Figure 7-18. Typical Frame Relay virtual private network client/server application

many VPNs. At the top of the figure, two larger server sites retain a database for an enterprise with many smaller client sites. The server sites are interconnected via PVCs, and each client site is connected to each server via a pair of PVCs. Thus, any single failure of a server, access line, or network switch leaves the network intact. If the access line to a client site were to fail, then only that site would be disconnected. When used with a public Frame Relay service that has a per-PVC charge, this type of design is often quite economical when the underlying enterprise paradigm is client/server based. A limited privacy agreement that we will discuss later in this section does provide for authentication, encryption, and key exchange operations, but does not provide protection from active security attacks.

FRAME RELAY—CONTROL PLANE

The Frame Relay control plane functions as a *signaling protocol*. The benefits of using Frame Relay's signaling capabilities are

- ▼ Status signaling for permanent virtual connections (PVCs)
- ▲ Dynamic call setup for switched virtual calls (SVCs)

We first cover the relationship of the control plane to the user plane, and then cover each of these applications by describing the formats and protocols along with an illustrative example of each.

Frame Relay Control Protocol Networking Context

Figure 7-19 illustrates the context for the Frame Relay control plane (C-plane) operating in conjunction with the user plane (U-plane) over a *single* physical interface. Starting in the upper left-hand corner on the CPE side of the User-to-Network Interface (UNI), note that the higher layers interface with both the control and user planes simultaneously. This model corresponds to the real-world application where a higher-layer protocol, such as IP first issues commands to the control plane to establish a Frame Relay connection through the serving FR network, prior to transferring data over the user plane connection. Next, note that on the right-hand side of the figure the network operates at not only the core of the data link layer for the user plane as described in the previous section, but also at the layer 3 control signaling in the control plane.

The shaded portions in Figure 7-19 indicate the closely related user and network protocols detailed in the Frame Relay standards discussed later. These standard user-side and network-side protocols must be present in compatible implementations in each end-user device attached to a service provider's Frame Relay network. Because of these clearly written standards, interoperability is commonplace in mission-critical, real-world applications around the world today. The data link layer provides a core set of services that constitute the actual Frame Relay service. On top of this same layer 2 service, the control data link layer adds a reliable, in-sequence delivery service in support of Frame Relay's layer 3 control signaling protocol used for SVCs. Signaling messages

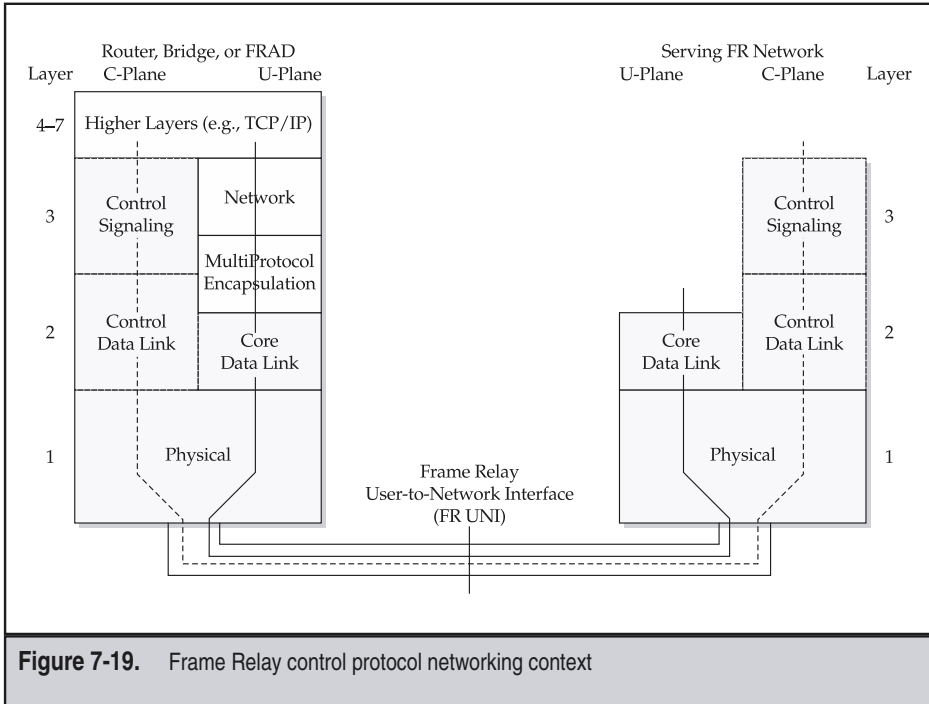


Figure 7-19. Frame Relay control protocol networking context

exchanged between a user and a network set up calls in a similar manner to those performed in ISDN and X.25 switched connections as studied earlier.

Frame Relay Standards and Specifications

A number of standards and specifications define the Frame Relay protocol at the User-to-Network Interface (UNI) and the Network-to-Network Interface (NNI). Furthermore, these specifications cover the user plane and the control plane. We start with the Frame Relay Forum (FRF) Implementation Agreements (IAs), since these reference other standards defined by the ITU-T and ANSI. Study of these Frame Relay standards is important to gain a full understanding of ATM, since the ITU-T signaling standards for ISDN (Q.931), Frame Relay (Q.933), and B-ISDN (Q.2931) all embrace a common philosophy, message structure, and approach. This is the secret to understanding the evolution of these technologies.

Frame Relay Forum Implementation Agreements

Related protocols in Frame Relay's control plane establish and release SVCs, as well as report on the status of PVCs. Figure 7-20 illustrates the various Frame Relay signaling protocols and their context. A Frame Relay connection can be either a PVC or an SVC as

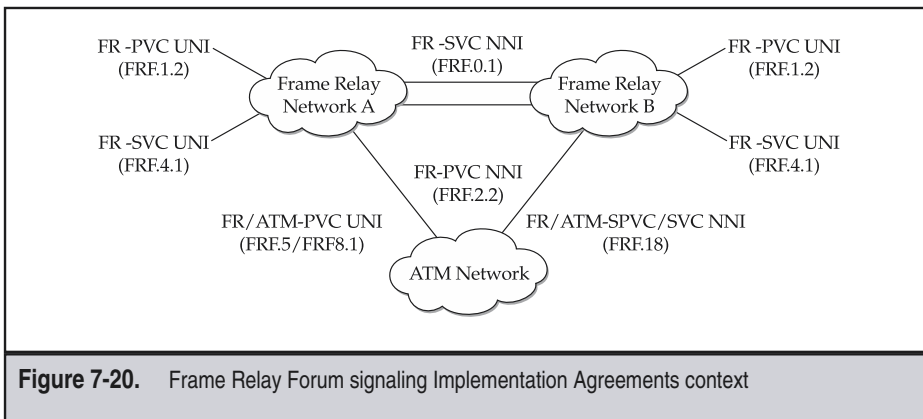


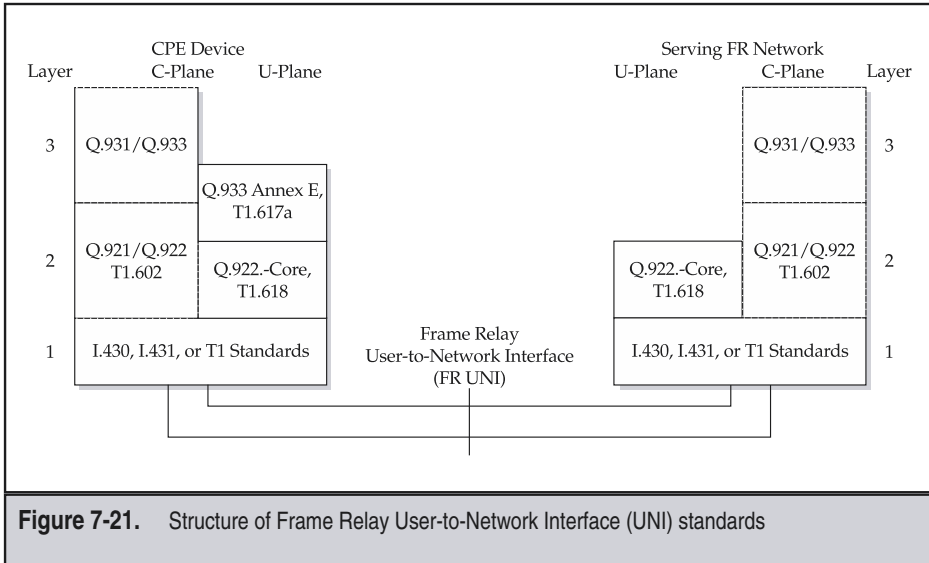
Figure 7-20. Frame Relay Forum signaling Implementation Agreements context

specified in Frame Relay Forum (FRF) implementation agreements [FRF.1.2] and [FRF.4.1], respectively. PVC signaling consists of conveying the edge-to-edge status of each Frame Relay DLCI, while SVCs offer connectivity on demand according to the network address and the Frame Relay traffic parameters. A different protocol interconnects Frame Relay networks at the NNI level for PVCs, as shown in Figure 7-20. The FRF.2.2 implementation agreement defines the NNI for PVCs, while FRF.10.1 defines the NNI for SVCs, and FRF.18 defines Frame Relay switched PVC/SVC to ATM Soft PVC/SVC service interworking across the NNI.

Frame Relay Forum (FRF) Implementation Agreements (IAs) are the reference used for interoperability agreements. Usually rather brief, the FRF IAs make extensive references to ITU-T international and ANSI national standards. These documents subset the standards, making modifications or clarifications of the standards as necessary. The next section provides a guide and references for more details on these concepts and important standards.

ITU-T and ANSI Frame Relay Standards

Figure 7-21 shows the major standards from the ITU-T (denoted as Q.xxx) and ANSI (denoted as T1.xxx) for the Frame Relay UNI C-plane and U-plane mapped out according to the protocol functions identified in the shaded area of Figure 7-19. The standards foundation for Frame Relay resides in ITU-T Recommendation Q.922 at the link layer (layer 2) and Recommendation Q.933 at the network layer (layer 3). The Q.922 standard defines the core frame relaying bearer service (also defined in ANSI T1.618), as well as the reliable data link layer for the Q.933 SVC signaling protocol. Recommendation Q.933 defines the formats, protocols, and procedures to dynamically set up and release switched virtual calls (SVCs) for the frame relaying bearer service. Note that standards for SVCs as well as PVCs often appear in the same document. For example, Annex A of Recommendation Q.933 defines status signaling for PVCs, while much of the



rest of the document focuses on the call control protocol for FR SVCs. Frame Relay standards split the data link layer of the user plane into two areas: core services and user-specified services, such as multiprotocol encapsulation specified in Q.933 Annex E, IETF RFC 2427 (which superseded RFC 1490), FRF.3.2, and T1.617a. Multiprotocol encapsulation provides a flexible method for carrying (multiplex and demultiplex) multiple protocols on a single Frame Relay connection.

Frame Relay has two modes of operation: an end user permanently connected to a Frame Relay switch using the Q.922 link layer, or such a user connected via circuit-switched access via Q.931/Q.921 ISDN signaling to a remote Frame Relay switch. Although not widely used, the second mode of operation is useful for on-demand or backup access to a Frame Relay network. See ANSI T1.617 or References [Black 94] or [Spohn 96] for more details on switched access to a remote frame switch. Some Frame Relay customers connect routers or bridges to Frame Relay switches with ISDN or switched 56 Kbps circuit-switched access for backup purposes or as a means to augment access capacity on demand.

Frame Relay PVC Status Signaling

Figure 7-22 illustrates the basic concept of Frame Relay status signaling for a CPE device connected to a Frame Relay switch on the edge of an FR network.

Status signaling allows the end user as well as the network to detect failures between the end points of PVCs. Generally, each physical access line connecting an end user to the serving Frame Relay switch carries multiple DLCIs. The FR standards define a status signaling protocol running on a particular DLCI (e.g., zero) that reports on the status for all

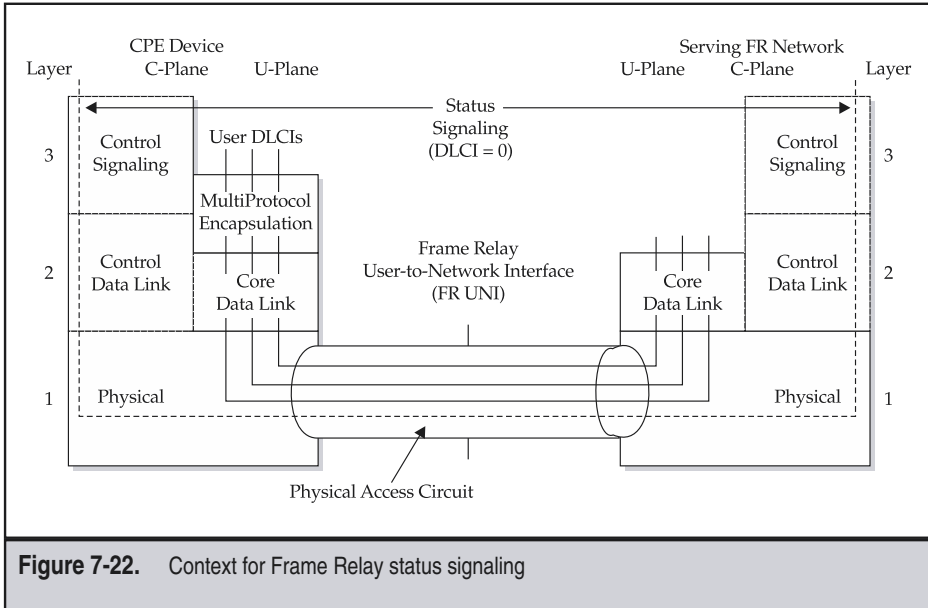


Figure 7-22. Context for Frame Relay status signaling

DLCIs carried by the physical FR UNI access line. The periodic exchange of status signaling messages also acts as a heartbeat and fault-detection mechanism for the physical access line connecting the user to the network. The status message also contains a field indicating that the network has provisioned a new PVC.

Originally, a group of vendors defined a local management interface (LMI) standard for Frame Relay. The LMI with extensions defined a protocol to provide a keep-alive signal between the FR CPE and the FR network access port. Also, the LMI simplified initial Frame Relay configuration by providing automatic notification of changes in PVC connectivity, as well as notification of the provisioning of new PVCs. Over time, incompatible standards arose between the LMI, ITU-T, and ANSI standards. For example, the LMI extension uses DLCI 1023 for status reporting, while T1.617 and ITU-T Q.933 Annex A employ DLCI 0, as shown in Figure 7-22. Fortunately, the ITU-T Q.933 Annex A standard is now closely aligned with ANSI T1.617, and the original proprietary LMI specification is falling out of use. Because of basic differences between the various local management interface standards, CPE and networks can automatically detect the status signaling protocol employed by the end user and react appropriately. This removes one configurable item, and thus reduces the complexity of turning up a Frame Relay PVC for service.

ITU-T Recommendation Q.933 Annex A and ANSI Standard T1.617 Annexes B and D define the modern Frame Relay status signaling message formats and procedures covered in this chapter. ANSI Standard T1.617 Annex B defines additional status signaling procedures for interfaces carrying both PVCs and SVCs. These specifications define three main

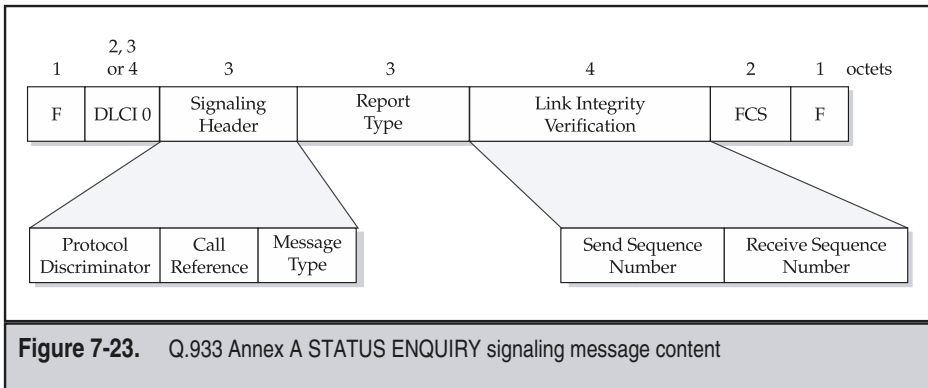
areas of PVC management: PVC status signaling, DLCI verification, and the physical interface keep-alive heartbeat.

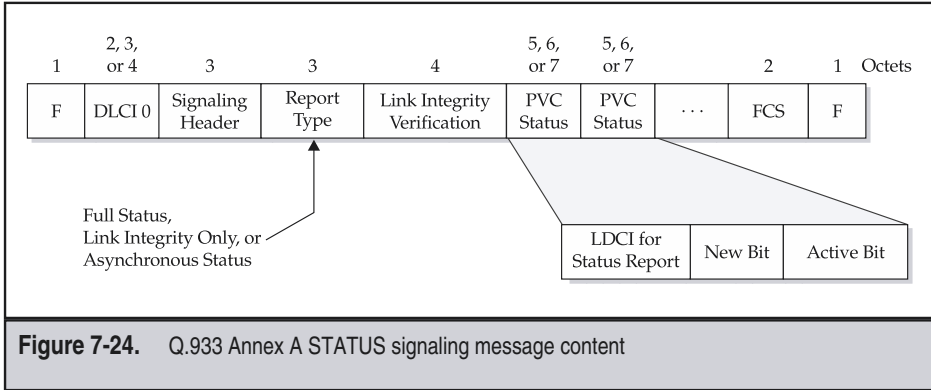
The PVC status signaling procedures utilize two messages: STATUS ENQUIRY and STATUS. Figure 7-23 shows the Q.933 STATUS ENQUIRY signaling message format and content. The message contains a signaling header common to all signaling messages, consisting of a protocol discriminator field, a call reference field, and the message type field (i.e., a string of bits indicating STATUS ENQUIRY, in this case). The report type information element indicates a choice of one of the following three options: a full status (FS) report on every provisioned PVC, link integrity verification only, or asynchronous reporting on sets of PVCs whose status has changed.

Figure 7-24 shows the format and content of a Frame Relay STATUS message from Q.933 Annex A. The STATUS message has the same format as the STATUS ENQUIRY message with the addition of one or more optional PVC status information elements. Of course, the value of the message type indicates that this is a STATUS message in response to a STATUS ENQUIRY. One or more optional PVC status information elements contain the DLCI of the connection that the status applies to, as defined by two bits: new and active. The new bit is set to 1 for a new PVC; otherwise, it is set to 0. The active bit indicates whether the PVC is operational (1) or not (0). The primary benefits of PVC status signaling derive largely from these two simple bits of information, as expanded on in the example in the next section.

T1.617 Annex D differs slightly from the Q.933 Annex A formats, as it adds a one-octet locking shift to the codeset 5 information element after the common signaling header, to indicate that the link integrity verification and PVC status information elements are for national use (e.g., the United States for ANSI). Status messages contain a link integrity verification field useful for checking continuity as well as acknowledging and detecting lost signaling messages, using the send and receive sequence numbers in a manner analogous to the sliding window protocol of X.25.

The maximum number of usable DLCIs on a physical interface is limited by the maximum frame size defined for that interface. This occurs because the maximum frame size





dictates how many PVC status information elements fit within a frame used to signal the Full Status (FS) message, and thus limits the maximum number of DLCIs on that interface. With enhancements to the PVC management procedures in FRF.1.2, this limitation, inherited from Q.933 Annex A, is removed. When the number of PVCs exceeds the number supported with the maximum frame size, these procedures add a status continued report type to the Report type Information element carried in the STATUS message in order to segment the full status message. Upon receiving a STATUS message indicating a “full status continued” type, the user device responds by sending another STATUS ENQUIRY message containing a “full status continued” type; this repeats until all status information is received, as indicated by the final STATUS message not indicating a status continued report type.

FRF.1.2 also enhances the LMI information included in the Full Status message to also report the link layer core parameters and the Priority and Service class parameters to the network or user. This will indicate the maximum frame size, the committed information rate, the committed burst size, the excess burst size, the transfer priority, and the service classes. In Q.933, the Full Status message is limited to reporting the status of the PVC, available or not. The reporting of availability is also enhanced in FRF.1.2 so that any change in the link layer core parameters or Service class parameters will initiate a STATUS message. Recognizing the fact that an NNI will likely need to support more DLCIs than a UNI, the Frame Relay Forum adopted an event-driven procedure documented in the FRF 2.2 implementation agreement. This procedure uses asynchronous status reporting to communicate only changed PVC status along with periodic updates for unchanged PVCs. This procedure overcomes the limit imposed by full status reporting on the maximum number of PVCs supported by an interface.

Frame Relay PVC Status Signaling Example

Figure 7-25 shows an example of Frame Relay status signaling illustrating access link failure detection and end-to-end PVC failure reporting.

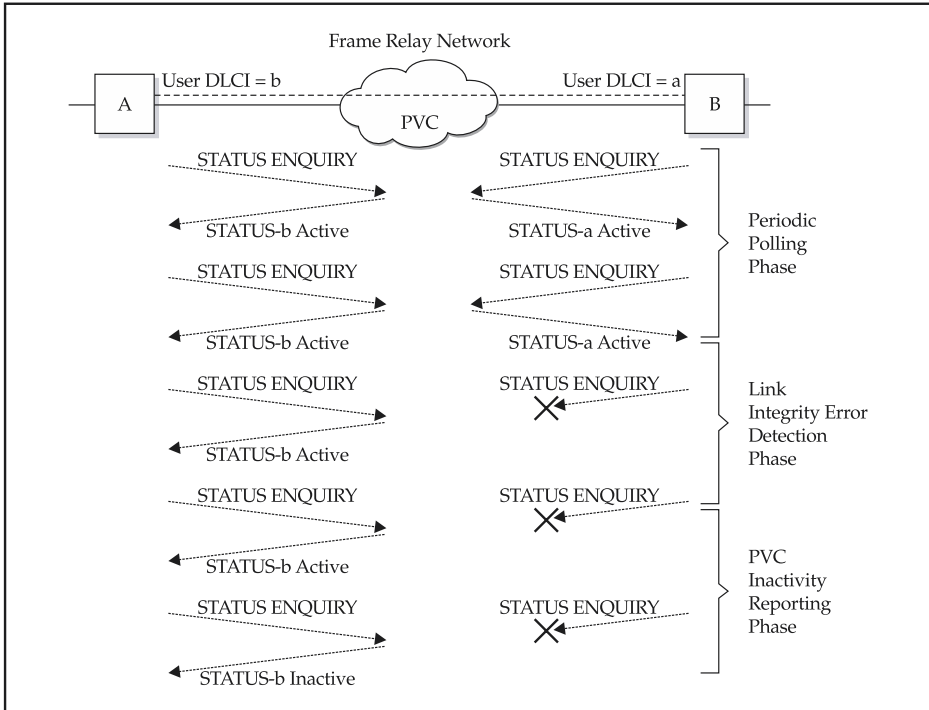


Figure 7-25. Example of Frame Relay status signaling operation

In the example in Figure 7-25, the Frame Relay network provides a PVC between CPE devices A and B operating on DLCIs b and a, respectively, as shown at the top of the figure. Time runs from top to bottom in the figure, with the message exchanges between devices A and B and the network shown as labeled arrows. As noticed during the periodic polling phase of messages in the figure, the CPE device periodically sends a STATUS ENQUIRY message to the network requesting a particular type of status report. The default polling interval is once every ten seconds. In Q.933 Annex A, the default is to poll only for full status, once every six message exchanges: the other message exchanges verify link integrity using the send and receive sequence numbers. For brevity, our example assumes that the requested report type is full status for each STATUS ENQUIRY message. The network responds with a STATUS message reporting that the PVC from A to B is active, as well as the other PVC information discussed previously. Although not shown in Figure 7-25, note that this same status message also reports on all other PVCs terminating on the UNI connected to A or B. In general, these status reports differ on A's and B's user-network interfaces, since at least one PVC will have a different destination, and hence will also have a different DLCI.

Continuing our example to the next phase, after two cycles of polling status, the physical access line, connecting user B to the network, fails. In the standards, if the user or network devices miss three consecutive STATUS messages, then they declare the link inactive. For brevity, we assume that this occurs after two missed messages at the expected times. The Frame Relay network now knows that the end-to-end PVC is down, and it reports this fact in the next STATUS message to user A. The network should also report changes in PVC active status for internal failures. Recommendation Q.933 states that the user equipment should cease sending frames if the network reports the PVC as inactive. Once the access line failure clears, the user and network reestablish the status enquiry-response status signaling protocol, synchronizing the sequence numbers in the link integrity verification information element, and stand ready to detect subsequent PVC status and parameter changes.

Multilink Frame Relay

Frame Relay networks typically operate with user connections up to a DS1/E1 rate, but there is a growing demand to support increased capacity because of more intensive Internet use, large file transfers, and a general increase of traffic. The most common higher-speed Frame Relay user interface in use today is a DS3/E3, and there is standards support for OC3/STM-1 rates, as well. However, most users who currently have a DS1/E1 do not need (nor want to pay for) a DS3/E3, but would prefer to increase their access capacity by a more modest amount in a more cost-effective manner. Adding additional DS1/E1 ports to the access device and splitting the traffic between them can be quite complex using IP; so in order to provide Frame Relay efficiently at intermediate speeds, Multilink Frame Relay was conceived. The basic approach to Multilink Frame Relay is to bundle multiple DS1/E1 links to act together and look like *one big pipe* with bandwidth approximately equal to the sum of the links. This concept, of course, is not new, and many vendor-proprietary implementations do just this, using what is normally called an inverse multiplexer (I-Mux), as described in Chapter 6 for TDM. With an I-Mux, a user most often will connect with a High Speed Serial Interface (HSSI) and the traffic stream would split across several DS1/E1 physical links.

The Frame Relay Forum provides two different standard Multilink Frame Relay (MLFR) implementation agreements: FRF.15 and FRF.16.1. The FRF.15 agreement covers an end-to-end mode, operating between two DTE peers. The CPE using the MLFR protocol provides an aggregated virtual circuit (AVC) allowing multiple virtual circuits to transport one single stream of frames, using a standard UNI on one or more DS1/E1 lines into a service provider. The service provider needs no knowledge of the MLFR; but since this kind of MLFR is only end-to-end, this means that each MLFR bundle can talk only to one endpoint. The ability to aggregate one frame stream over several virtual circuits is extremely useful to provide resilience by allocating the frame stream across two redundant UNIs, as shown in Figure 7-26.

Since frames in a stream may lose their original sequence as they move in an AVC over several independent virtual circuits, called *constituent virtual circuits (CVC)*, every packet is marked with a sequence number that is used for resequencing at the destination. Because

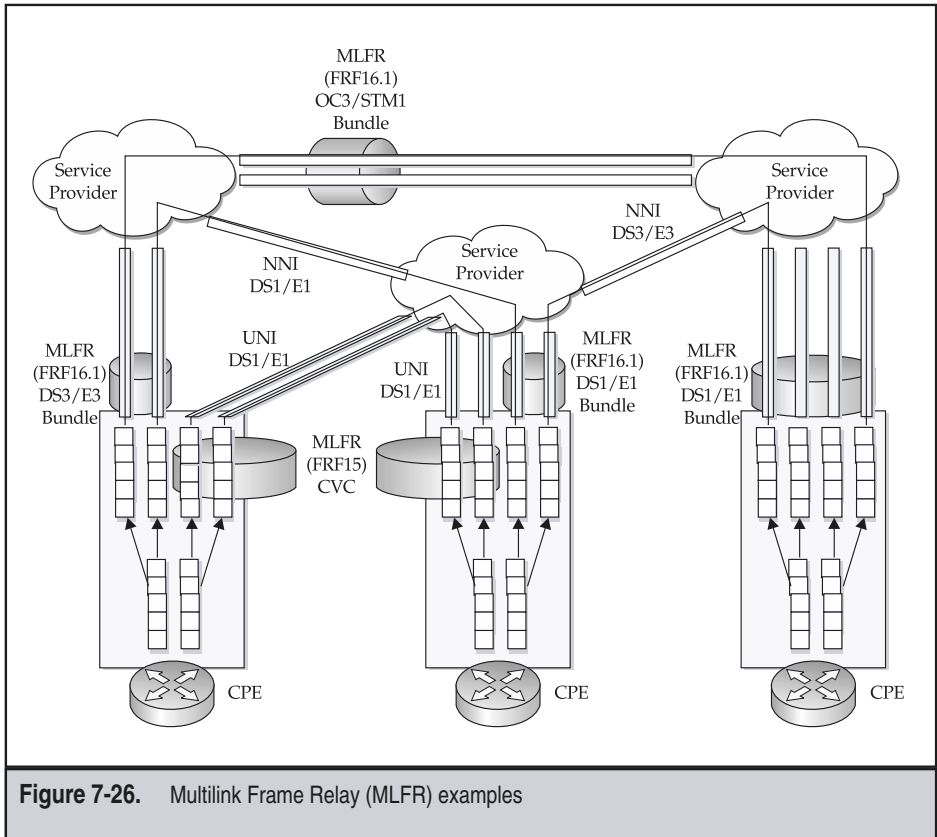


Figure 7-26. Multilink Frame Relay (MLFR) examples

resequencing of frames may be necessary, a frame-loss detection mechanism is also needed so that frames are not buffered indefinitely. Use of a minimum interval timer to declare a frame lost when a reasonable time required to transmit a frame is exceeded, or checking all the CVCs to determine if the last received sequence numbers are higher than the one missing, are both suggested in FRF.15.

Fragmentation (FRF.12), discussed later in this section, can also be used with MLFR, where different fragments can use different CVCs, and therefore the receiving end needs to be able to receive individual fragments to reassemble the frame from any CVCs. The need for sequencing, resequencing, fragmentation capabilities, and frame-loss detection may cause implementations to introduce additional latency with MLFR; so when using

applications that may be sensitive to additional latency, ask vendors for their specific implementation methods and performance. Status reporting for the AVC is determined by the underlying CVCs independently, using existing standard Frame Relay signaling; but at layer 3, the AVC status is aggregated into one AVC status and can be based on one of three different modes mutually agreed upon between the communicating peer DTE:

- ▼ **Mode 1** The AVC is in active status if any CVC is in active status.
- **Mode 2** The AVC is in active status only if all CVC are in active status.
- ▲ **Mode 3** The AVC may define an active operational state, based on a minimum number of active CVCs, the available CIR on each CVC, or the total aggregated CIR available on the AVC.

FRF.16.1 differs from FRF.15 in very significant ways. In FRF.16.1, multiple UNI and NNI physical links are aggregated into one logical link with MLFR, called a “bundle,” which then still supports all Frame Relay services based on the UNI and NNI standards. Therefore, MLFR is implemented between logically adjacent devices—and not only between two DTEs, as shown in the examples of Figure 7-26. Once in the service provider network, the traffic can be routed over other MLFR bundles. Because the bundle terminates at each network interface, different endpoints via multiple PVCs on each bundle can be supported. FRF.16.1 also provides standard provisioning and signaling methods in order to dynamically add, remove, activate, and deactivate links from a MLFR bundle; but in essence, the functionality described previously for virtual circuits remains similar, providing for frame order and fragmentation procedures. Importantly, the standard allows any link within a bundle to fail without bringing down the rest of the links within the bundle. The bundle operates in conjunction with the Q.922 data link layer, emulating the functions of the physical layer, and multiplexes all the different data link layer connections within the bundle, including the signaling links.

Frame Relay Service Level Agreements (SLAs)

We now come back to the problem of different service provider interpretations and use of traffic variables mentioned earlier. With Service Level Definition, FRF.13, and Operations, Administration, and Maintenance (OA&M) Protocol and Procedures, FRF.19, the road toward meaningful service level agreements (SLAs) and verification procedures for an end user is laid out. These two agreements can be used by communication managers to diagnose potential issues with Frame Relay VCs, as well as monitor service level agreements. We describe these two agreements in this section.

FRF.13 utilizes some of the principles and reference models defined in ITU Recommendation X.144, and the focus is on defining the characteristics of the following parameters: Frame Transfer Delay (FTD), Frame Delivery Ratio (FDR), Data Delivery Ratio (DDR), and Service Availability. The parameters are defined for point-to-point connections only, and a

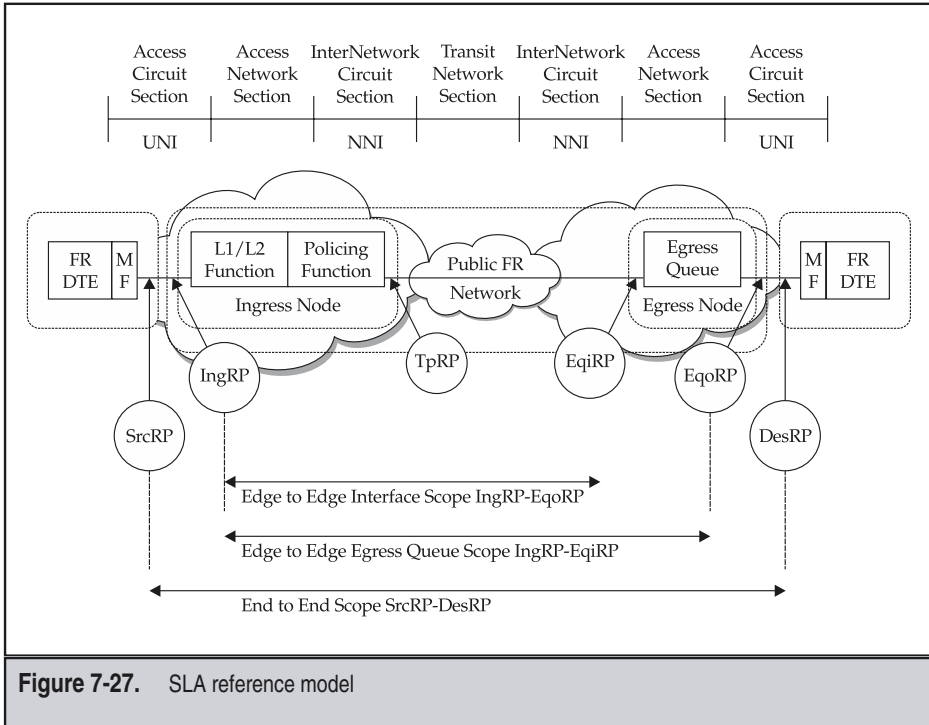


Figure 7-27. SLA reference model

reference model, shown in Figure 7-27, representing the structure of any end-to-end connection is used to describe the boundary domains for the appropriate measurement points.

Figure 7-27 shows a source Frame Relay DTE where traffic originates. A measurement function either is located in an external probe or may be integrated within the Frame Relay DTE. The public Frame Relay network is an abstraction of a public-offered service and could be any number of intermediate systems using NNIs. The ingress and egress nodes represent the first and last of the network equipment in the path of the connection. The L1/L2 function refers to the OSIRM layer 1 and layer 2 processing applied to the incoming frame; if errors, such as FCS errors or invalid addresses, are encountered, then frames are discarded. The traffic policing function and the egress queue function represent the application of traffic contracts and the queue associated with the destination physical interface. Finally, the destination Frame Relay DTE is the receiving end of the connection.

The reference points for the SLA measurements are indicated in Figure 7-27 as well, and three connection segments are defined: edge to edge (between the Ingress Reference Point, IngRP, and the Egress Output Queue Reference Point, EgoRP), edge to egress

queue (between the Ingress Reference Point, IngRP, and the Egress Input Queue Reference Point, EqiRP), and end to end (between the Source Reference Point, SrcRP, and the Destination Reference Point, DesRP). The FTD measurement reflects the time required to transport Frame Relay data through the network between the reference points for the three measurement domains. Note that no jitter (delay variation) SLA parameter is defined, and no details on what measurement methodology to use are specified beyond a very general description. The measurement methodology for FTD and DDT are also not defined in FRF.13.

Within FRF.19, these methods are better specified, and we will discuss the SLA definitions further within this context. FRF.19 does not specify appropriate measurements for service availability, reflecting the general difficulty in tracking the information needed to calculate an accurate value. Two types of outages are differentiated: fault outages and excluded outages. Fault outages result from faults in the network for various reasons and normally would be indicated by a VC status change, some alarm condition, or simply a trouble ticket that needs to be resolved. Excluded outages are defined to be scheduled maintenance times and outages resulting from events “out of the control of the network,” a very flexible yet ambiguous definition. A user should examine closely the methodology a service provider used to calculate and claim availability values.

Frame Relay Operations, Administration, and Maintenance

Even though the framework and parameters for an SLA are defined, you also need to have a standard way to measure these parameters so that real comparisons and verifications can be made. FRF.19 provides the capability to test, diagnose, and measure the quality of a Frame Relay service in a comparable and consistent fashion with Operations, Administration, and Maintenance (OA&M) information that can be passed within an OA&M Frame Relay message type. Used to define a reference model for typical Frame Relay network configurations, and indicating monitoring points for VCs that span different network sections and administrative boundaries, this information can consistently measure conformance to an SLA that may be in place for a connection. Figure 7-28 shows the reference model for the monitoring points that could be used to verify the three different types of SLAs defined in FRF.13, along with additional network diagnostic information.

FRF.19 is designed to either supplement or replace the ITU I.620 Frame Relay operations and maintenance principles and functions; it does not claim interoperability with these functions of the ITU I.610 specification. Note that the indicated monitoring points marked “MP” in Figure 7-28 are supported with FRF.19 and are defined as Frame Relay OA&M monitoring points (FROMP), while the monitoring point marked “AMP” is an ATM end-point device included to illustrate an FR/ATM interworking scenario. However, OA&M monitoring of the Frame Relay DLCI while performing service interworking to an ATM endpoint is not yet standardized.

The complete network monitoring reference model is laid out by describing different virtual circuits that cross the reference network, three individual circuit models, and an

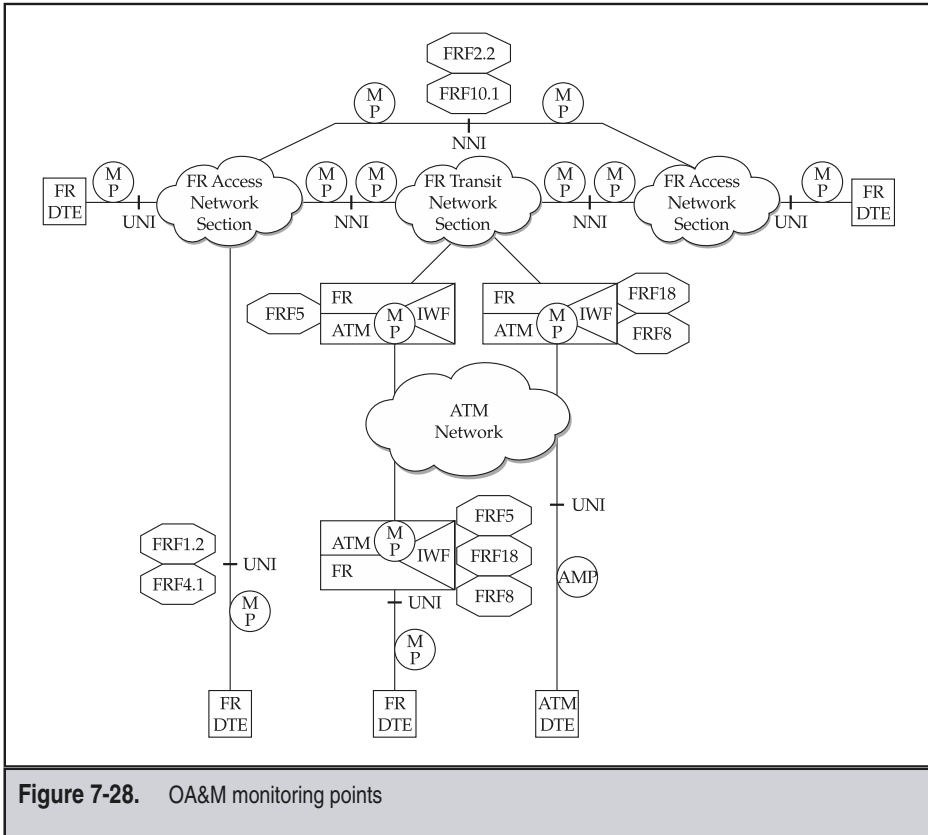


Figure 7-28. OA&M monitoring points

administrative domain reference model. Figure 7-28 indicates these three models. The reference network shows a Frame Relay network that also interoperates with an ATM network. There are several virtual connections that can be made across this network. A VC may cross a single Frame Relay access network section, two Frame Relay access network sections connected via an NNI, two Frame Relay access network sections connected via a Frame Relay transit network, two Frame Relay access network sections connected by an ATM transit network section using network interworking, or a single Frame Relay VC connection spanning a single access network section terminated as an ATM VC with service interworking. Each VC has different monitoring points in order to verify performance with the SLA type, either an end-to-end, edge-to-edge, or edge-to-edge queue type. A virtual connection may also cross administrative boundaries

where an administrative domain represents portions of a VC that are administered by the same organization(s).

The OA&M monitoring and reporting functionality is set up automatically with signaling procedures for both PVCs and SVCs. A Frame Relay MP may initiate an MP discovery process in order to engage devices that are advertising their presence and are willing to participate in OA&M. This process is initiated with a HELLO message that each MP sends out periodically. Upon receipt in return of a valid HELLO message an MP peer, which indicates the OA&M capabilities that are available for the duration of the Frame Relay connection, other OA&M messages (other than the HELLO) can be transmitted toward the MP closest to the VC endpoints. HELLO messages apply only within a single domain; and when a VC crosses administrative boundaries, the MP may generate separate HELLO messages for every administrative domain to which it belongs.

A Frame Relay MP may belong to multiple domains and may be allowed to generate separate HELLO messages for each of these domains, and may advertise different OA&M capabilities as well. An MP at an administrative boundary can also not be allowed to participate in exchanging information with other domains. The administrative domains provide for well-defined zones of OA&M processing. All OA&M messages contain a domain identification that is used to identify the intended administrative domain. A Frame Relay MP at a domain boundary will not forward its domain's HELLO messages beyond the domain boundary, it also will, by comparing the origin location of the MP message, detect and discard counterfeit messages destined toward points inside the MP domain that enter from outside its domain. Messages destined for other domains will be passed through in either direction without interpretation or discard.

Three service measurements are supported by the OA&M service verification message: frame transfer delay (FTD), frame delivery ratio (FDR), and data delivery ratio (DDR). The measurements are independent of each other and use individual information fields, but they could be combined in a single message.

The FTD is measured as a round-trip delay, and this is divided in half to obtain the one-way FTD value, as defined in FRF.13. The OA&M message initiates the measurement by sending an "initiator" TX time stamp representing the time the opening bit of the frame begins transmission. The responding Frame Relay MP copies this value and then adds a "responder" RX timestamp, representing the arrival time of the closing bit of the frame, as well as the opening bit of the frame that will begin transmission. A pad field is used to assure that the received and sent messages are exactly the same length. Once the initiator receives this message back and records the timestamp of the arrival time of the closing bit of the received frame, the FTD is determined by

$$\text{FTD} = ((\text{Initiator_RX} - \text{Initiator_TX}) - (\text{Responder_TX} - \text{Responder_RX})) / 2$$

The FDR is measured by completing several exchanges between the initiator and responder MPs. The beginning of a measurement requires a synchronization of the respective MP frame counters. Attempted frame transmissions are referred to as Frames Offered. Successfully delivered frames are referred to as Frames Delivered. These

measures are further categorized as being within the committed information rate (Frames Offered_C or Frames Delivered_C) or as burst excess (Frames Offered_e or Frames Delivered_e), as follows:

$$\begin{aligned} \text{Frames Offered}_C &= \text{Frames Offered}_{C2} - \text{Frames Offered}_{C1} \\ \text{Frames Offered}_e &= \text{Frames Offered}_{e2} - \text{Frames Offered}_{e1} \\ \text{Frames Delivered}_C &= \text{Frames Received}_{C2} - \text{Frames Received}_{C1} \\ \text{Frames Delivered}_e &= \text{Frames Received}_{e2} - \text{Frames Received}_{e1} \\ \text{Frames Lost}_C &= \text{Frames Offered}_C - \text{Frames Delivered}_C \\ \text{Frames Lost}_e &= \text{Frames Offered}_e - \text{Frames Delivered}_e \\ \text{FDR}_C &= \text{Frames Delivered}_C / \text{Frames Offered}_C \\ \text{FDR}_e &= \text{Frames Delivered}_e / \text{Frames Offered}_e \end{aligned}$$

$$\text{FDR} = \frac{\text{Frames Delivered}_C + \text{Frames Delivered}_e}{\text{Frames Offered}_C + \text{Frames Offered}_e}$$

The DDR measurement requires synchronization, recording, and storage of the counters for data offered and data received in order to establish the ratio of octets delivered and octets offered in the same way as the FDR. The values for DDR are calculated by using the same methodology as the FDR, substituting data octets for frames to arrive at DDR_C, DDR_e, and DDR:

$$\begin{aligned} \text{DDR}_C &= \text{Data Delivered}_C (\text{committed}) / \text{Data Offered}_C (\text{committed}) \\ \text{DDR}_e &= \text{Data Delivered}_e (\text{excess}) / \text{Data Offered}_e (\text{excess}) \end{aligned}$$

$$\text{DDR} = \frac{\text{Data Delivered}_C + \text{Data Delivered}_e}{\text{Data Offered}_C + \text{Data Offered}_e}$$

Frame Relay diagnostics may also be performed on the VC between two MP points within the same domain. Two diagnostics are supported: latching and nonlatching loopback. The latching form is a service maintenance action that removes the VC from service. The nonlatching one is used to echo an individual OA&M frame without taking the VC out of service so that frames are still forwarded. Latching causes all the arriving frames of a specific VC to be looped back toward the transmitter, while other VCs passing through the same device will not be affected. While in latching loopback, only OA&M cells directly addressed to the loopback MP points will be processed, thus enabling the FTD, FDR, and DDT to still be measured and additional diagnostic information to be processed. Diagnostic information may be indications of whether latching loopback is enabled or not, and whether the physical layer is up or down.

Frame Relay Fragmentation and Compression

Frame Relay supports encoding of user data into compressed data and decoding this back to uncompressed user data, using a Data Compression Protocol (DCP), as specified in FRF.9. With FRF.20, Frame Relay also supports a control protocol to negotiate the use of IP header compression over a Frame Relay VC. We discuss the details of both these

compression alternatives, but first we will examine how these compression protocols interact with the FRF.12 fragmentation procedures that are based on RFC 1990, as well as the FRF.11.1 Voice over Frame Relay, the FRF.19 OA&M, and the FRF.3.2 Multiprotocol Encapsulation (MPE) Implementation Agreements. Only certain combinations of features can be supported on a single Frame Relay connection, with specific combinations resulting in improved performance.

Table 7-2 shows the combination of capabilities that can coexist on a single Frame Relay VC. When transmitting data over Frame Relay using lower bandwidth links, to better utilize the available bandwidth, various techniques of compression could be used. When transmitting both voice and data packets, which may have both short and long frames, together with real-time traffic, compression of data and headers does not control the delay and delay variation required when, for example, voice is carried on the same link. Fragmentation of the data carrying VCs on such a link is designed to segment the expected longer frames into a sequence of smaller frames, which will be reassembled by the receiving peer DTE or DCE, resulting in better delay and delay variation characteristics on the link.

When compressing data frames using FRF.9, the frames are also multiprotocol encapsulated according to FRF.3.2. Compression can be performed on this PVC first, and then the compressed frame can also be fragmented. At the receiving end, the fragmentation is first reassembled and then uncompressed. A wrinkle of complexity can be encountered, as the FRF.11.1 agreement also allows fragmented data frames to be carried in the Voice over Frame Relay subframe data payload. This means voice and fragmented data frames can be combined on the same PVC, with the data optionally being compressed as well. FRF.20 allows for packets with compressed IP headers to be encapsulated and transported, but one caveat here is that the data compression (FRF.9) and IP header compression (FRF.20) schemes can be simultaneously performed on the same VC, but not the same packet.

Valid Configuration	End-to-end Fragmentation for single protocol encapsulation	UI Frame Multiprotocol Encapsulation (MPE)	Non-UI Frame MPE	Single Protocol X.25	FRF.11.1 VoFR	UI-Only MPE OAM	Non-UI OAM	SVC LLC Without OAM	SVC LLC IE with OAM
Full MPE		X	X			X		X	X
UI-Only MPE and VoFR		X			X	X		X	X
X.25 Data and VoFR	X			X	X		X	X	X
UI-Only MPE		X				X		X	X
X.25 Only	X			X			X	X	X
VoFR Only	X				X	X		X	X

Table 7-2. Frame Relay Capabilities on a Single DLCI

DCP is not supported in Frame Relay signaling, and therefore a connection with the appropriate expected bandwidth for the compressed data needs to be established before the DCP negotiation can be initiated. If negotiation for DCP fails for some reason, the connection is not necessarily released, and the bandwidth may not be sufficient to carry uncompressed information.

The DCP defines a general frame format for DCP frames used for data and control that are distinguished by the DCP header information. The control frames format is based on PPP (RFC 1661). The control protocol allows negotiation of the data compression function definition and its associated parameters. Interacting systems are expected to support several optional compression algorithms, and through negotiation accept an algorithm that both can support. The default compression function in FRF.9 is defined to be ANSI X3.241-1994, and in combination with the compression protocol is referred to as LZS-DCP. Any number of alternative and proprietary compression functions could be negotiated with DCP. When data compression is performed on a connection crossing an FR/ATM Service interworking function (FRF.8.1, FRF.18), the compression header is mapped to that specified in RFC 2427. If the ATM endpoint does not implement compression, then the frame will be discarded, thereby causing the originating device to time out and disable compression for the connection.

A Frame Relay IP Header Control Protocol (FRIHCP) is also defined that can negotiate the use of IP header compression in both directions on a VC. This agreement is based on the Link Control Protocol of PPP (RFC 1661). Algorithms used for compression are described in RFC 2507 and RFC 2508. The FRIHCP is a single protocol to negotiate header compression for both IPv4 and IPv6. A Frame Relay connection is assumed to already be established. A header compression negotiation can then begin, followed by transfer of the compressed headers and data. The negotiation is entirely done in the user plane, transparent to the Frame Relay network. This protocol will work with network interworking with ATM [FRF.5], but not over a VC that is using service interworking with ATM, as defined in [FRF.8.1] and [FRF.18].

Frame Relay Privacy

The FRF.17 Frame Relay Privacy Protocol (FRPP) is used on a per-VC basis, from end user to end user, over the Frame Relay user plane. All the FRPP frames follow the frame format of FRF.3.2, and the frame content is completely transparent to the Frame Relay network between the transmitting and receiving DTE. The VC must be established before the FRPP can be used by using standard PVC or SVC procedures before FRPP can be used. If the FRPP is initiated by the end user, it may include three stages: an authentication stage that is optional, followed by an encryption stage and a key exchange stage that must complete before the encrypted user data can be transferred. The protocol is based on work in the IETF, the PPP link control protocol (RFC 1661) and PPP encryption control protocols (RFC 1968 and 1969). Compression can also be used with this protocol as defined in FRF.9.

FRPP provides two modes of operation. Mode 1 uses a default algorithm, ANSI X3.92, that is also used in RFC 1968, which is the U.S. Data Encryption Standard (DES) with

Cipher Block Chaining (CBC). For more robust security, triple DES can be used as well. Mode 2 allows full negotiation of the encryption algorithm that a user may want to use, with all its associated parameters.

Since FRPP operates in the user plane, the only distinction that the Frame Relay network needs to provide consists of identifying the control frames that carry the information related to the authentication, encryption, and key exchange, as opposed to the frames that contain actual encrypted user data. The general frame structure in Figure 7-29 is used for both.

The distinction between the user and control frames is made by setting the Control/Data field (C/D) in the FRPP header to 1, indicating a control frame. The authentication (A) field is set to 1 if the frame is an authentication frame as opposed to an encryption frame. FRPP authentication supports any of the protocols defined for PPP. The encryption method is negotiated between the peer devices, and the mode and algorithm are selected independently for each direction of the VC. Encryption assumes that the two peers have a “shared secret” used in the encryption; how the peers distribute this secret in a secure manner is not specified.

A certain order must be followed when using this capability with other Frame Relay capabilities. In general, multiprotocol encapsulation is best done first, and then compression, encryption (although the encryption negotiation is recommended to be completed before

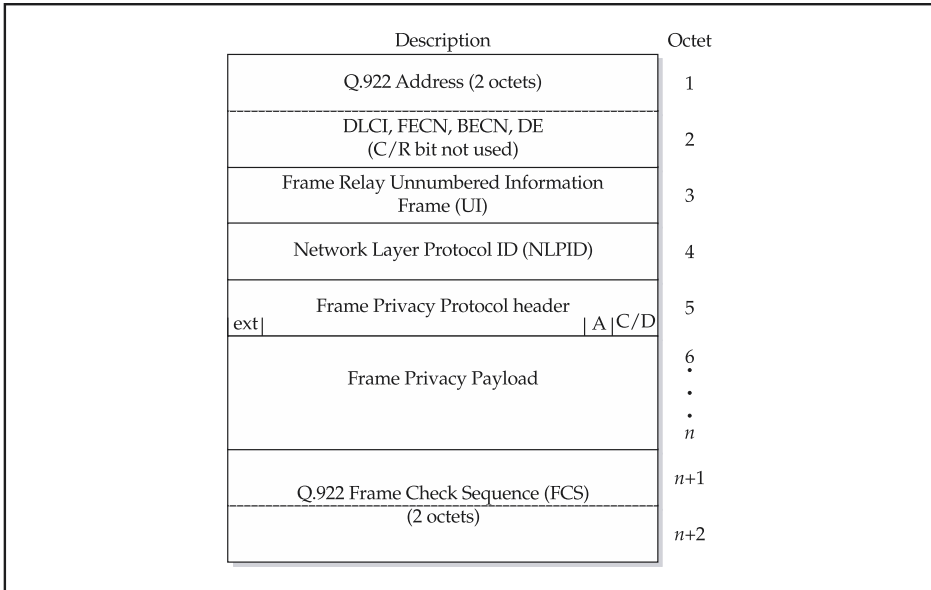


Figure 7-29. Frame Relay Privacy Protocol frame structure

the data compression negotiation), and fragmentation last (by either FRF.11.1 Annex J or FRF.12). FRPP is not intended to provide complete security, but instead aims to discourage easy observation of data transiting a Frame Relay network. More robust security protocols, such as IPsec, should be used for highly sensitive data.

Frame Relay Switched Virtual Connections (SVCs)

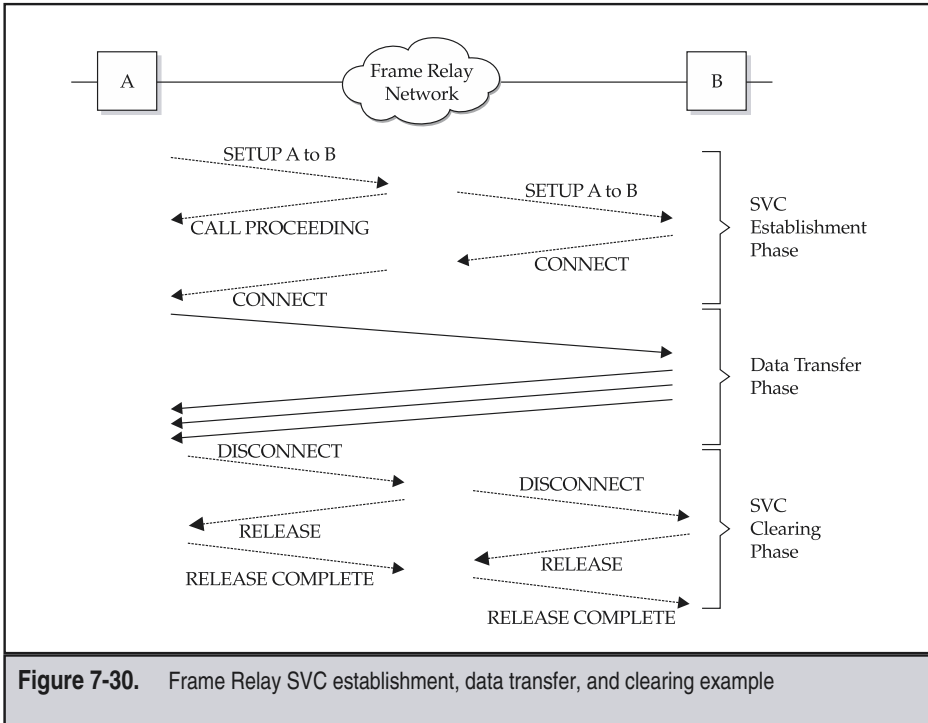
A CPE device can use the Frame Relay SVC signaling protocol (Q.933) to dynamically establish temporary connections across a Frame Relay network to another CPE device. Frame Relay SVCs support both the ITU-T (CCITT) E.164 and X.121 numbering plans for the purposes of identifying interfaces. X.121 supports interworking with X.25 networks, while E.164 is the long-term direction stated in standards. Each service provider administers its own numbering plans using, for example, a carrier prefix code in the E.164 numbering plan. The SVC protocol dynamically assigns each end of the connection a unique DLCI used for the duration of the call. SVC call control also provides a mechanism for parameter negotiation (e.g., maximum frame size, traffic parameters, or transit delay). The user exchanges SVC call control messages with the network over DLCI 0. SVC control messages require a reliable link layer protocol, as specified in the Frame Relay Forum's FRF.4.1 Implementation Agreement based upon Recommendation Q.922.

Even though the SVC standards are well documented, wide implementation of SVC services has been slow, primarily due to the complexity of SVC management within networks engineered primarily for predictable PVC traffic loads, alignment of standards, and the requirement for strict security and administration.

Example of Frame Relay SVC Operation

The Frame Relay signaling procedures for SVCs at the UNI are based upon ITU-T Recommendation Q.933, which uses Q.922 as a reliable link layer protocol. Q.933 signaling messages are sent on DLCI zero [FRF.4.1]. Figure 7-30 illustrates the sequence of messages involved in establishing and clearing a Frame Relay SVC. The messages are similar to those used in ATM in that they both have the same ancestor—narrowband ISDN. The next section covers the contents and the basic semantics of these messages.

Note that messages are sent in one of two contexts: from the user to the network, and from the network to the user. The rules of operation and allowed message information content differ somewhat in these two contexts as defined in the standards. In the SVC establishment phase, calling user A sends a SETUP message to the network, indicating B as the called party. If the network accepts the request, it responds with a CALL PROCEEDING message, finds a route to the called party, and generates a SETUP message toward B. If B accepts the incoming call request, it responds with a CONNECT message. Concurrently, the network propagates B's acceptance of the callback to calling user A via a CONNECT message. Now, A and B may exchange frames on the dynamically established DLCI, until either party clears the call by sending a DISCONNECT message. In this example, A initiates call clearing. A two-phase handshake, involving the RELEASE and



RELEASE COMPLETE messages, as shown in the figure, confirms that a call is indeed cleared. The multiple-level handshake in the SVC establishment and clearing phases is especially important if the network charges by call duration for SVC services. For example, failure to properly release a call could result in a very large charge to the user, since the network would record the event as a long call.

Frame Relay Signaling Message Information Elements

Similar to ISDN signaling messages, Frame Relay messages contain a number of information elements (IEs), which are either Mandatory (M) or Optional (O), as specified in ITU-T Recommendation Q.933. Table 7-3 depicts the population of the principal Frame Relay signaling message types with information elements. The Frame Relay Forum FRF.4.1 implementation agreement further subsets these requirements by not requiring the ALERTING, CONNECT ACKNOWLEDGE, or PROGRESS messages, as indicated in the table. Furthermore, the FRF document eliminates a majority of the information elements in interoperable implementations, as indicated by asterisks in Table 7-3.

	Alerting*	Call Proceeding	Connect	Connect Acknowledge*	Progress*	Setup	Disconnect	Release	Release Complete	Status Enquiry	Restart	Restart Acknowledge	OA&M
Protocol discriminator	M	M	M	M	M	M	M	M	M	M	M	M	M
Call reference	M	M	M	M	M	M	M	M	M	M	M	M	M
Message type	M	M	M	M	M	M	M	M	M	M	M	M	M
Bearer capability					M								
Data link connection identifier	O-1	O-1	O-1		O								
Link layer core parameters			O		O				O				
Calling party number					O								
Calling party subaddress					O								
Called party number					O								
Called party subaddress					O								
Transit network selection					O								
Cause (code)							M	O-2	O-2			M	M
Call state													
Channel identification*	O-1	O-1	O-1		O								

Table 7-3. Frame Relay Signaling Message Information Element Content

	Call		Connect		Progress*		Setup		Disconnect		Release		Status		Restart	
	Alerting*	Proceeding	Connect	Connect Acknowledge*	Progress*	Progress*	Setup	Setup	Disconnect	Disconnect	Release Complete	Release Complete	Status Enquiry	Status Enquiry	Restart Acknowledge	Restart Acknowledge
Progress indicator*	○	○	○	○	M	○	○	○	○	○	○	○	○	○	○	○
Network-specific facilities*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Display*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
End-to-end transit delay*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Packet layer binary parameters*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Link layer protocol parameters*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
X.213 priority*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Connected number*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Connected subaddress*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Repeat indicator*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Low-layer compatibility*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
High-layer compatibility*	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Priority and service class parameters	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Table 7-3. Frame Relay Signaling Message Information Element Content (continued)

	Call		Connect			Release		Status		Restart		OA&M
	Alerting*	Proceeding	Connect	Acknowledge*	Progress*	Setup	Disconnect	Release	Complete	Status Enquiry	Restart	
Capabilities**												M
Verification**												M
Non-latching loopback**												M
Latching loopback**												M
Diagnostic indication**												M
Called party SPVC												M
Calling party SPVC												M
CUG Interlock code												
Call identification												
User-user*	O					O					O	

1 = Mandatory if first message in response to SETUP
2 = Mandatory if first call clearing message
M = Mandatory
* Not applicable in FRF.4 implementation agreement
O = Optional
**Mandatory for the related OA&M type
O-x = Optional with note x

Table 7-3. Frame Relay Signaling Message Information Element Content (continued)

As seen in Table 7-3, the SETUP message contains by far the largest number of IEs, since it must convey all of the information about the connection so that the network or the called user can decide whether to accept the call. The principal information elements are the called party number and subaddress, transit network selection, and link layer core parameters (i.e., Bc, Be, and T). The calling party number and subaddress are optional. The DLCI information element enables dynamic assignment of the Frame Relay header values at call setup time. The bearer capability information element indicates transfer of unrestricted digital information using the frame mode service. The cause message indicates the reason for call release or the current status of an active SVC in response to a STATUS ENQUIRY message.

When to Use SVCs Instead of PVCs

A PVC is permanent, as the first letter of the acronym name states; while an SVC is switched, or established on demand. A user application can establish an SVC only when there is information to send, and then disconnect it once the information is transferred. This can be a considerable economic advantage if the FR SVC service provider charges for the duration of SVC calls. On the other hand, a PVC is always in an established mode, whether there is information to transfer or not. Although this means that there may be periods where unused capacity exists, PVCs avoid the possibility of call blocking that SVCs may have.

Frame Relay SVCs offer a more scalable solution than PVCs in networks with a large number of nodes that require only sporadic or ad hoc connectivity: for example, video conferencing, voice, data backup, or file transfer applications. Switched connections also make sense for remote sites that require access to other sites only on an intermittent basis. Furthermore, if the service provider implements address screening or closed user group interlock codes, then SVCs enable intranets and extranets.

In summary, PVCs are best when replacing a private line network with a virtual private network. A least-cost design for many of these PVC-based networks results in a single- or multiple-star topology, as described earlier in this chapter. A full-mesh topology for PVCs is, however, often uneconomical for all but the smallest Frame Relay networks. SVCs avoid the multiple hops through intermediate routers that occur in PVC-based networks because they dynamically establish a path from source to destination. SVCs are best suited for very large networks with either unknown or changing connectivity requirements. Furthermore, SVCs can be employed to better track traffic patterns and charge back actual network use to the actual application if the service provider supports usage-based billing. The major issue with SVCs is that as an on-demand service, there is a finite probability of blocking. By definition, PVC networks are nonblocking. If the network blocks an SVC call attempt, then the user cannot transfer any information. If your application cannot tolerate occasional call blocking, then you should design PVCs into the network to meet the need for guaranteed bandwidth that only a PVC can provide. As a final note, hybrid SVC/PVC networks are possible. A node initially connected via SVCs can be changed to a PVC connection if the performance, traffic patterns, and economics warrant such a change.

REVIEW

This chapter covered the predominant connection-oriented data services: X.25 and Frame Relay, in terms of the origins, protocol formats, functions, operation, traffic aspects, and commercial service aspects. The text first described the way X.25 performs store-and-forward packet switching and windowed flow control to account for the relatively poor private line network performance of the 1970s and early 1980s. Next, the coverage moved on to the successor of X.25 and private lines: Frame Relay. Essentially, Frame Relay is X.25 on a diet, and the lower feature content of Frame Relay has enabled a ten-fold increase in throughput over classical X.25 packet switching. Being part of the ITU-T's Integrated Services Digital Network extended family, Frame Relay defines concepts of traffic and congestion control similar to those used in ATM. Furthermore, the operation of SVCs in Frame Relay is similar to that of ATM. An appreciation of the common underlying paradigms flowing through the technologies from ISDN to MPLS can be gained. from studying technological evolution. This chapter provides the reader with a basic background in the connection-oriented data communication services and concepts that reappear within ATM and MPLS, as described throughout the remainder of the book.

CHAPTER 8

Connectionless Protocols—IP and SMDS

This chapter describes the dominant connectionless data service available in public and private data services during the 1990s—namely, the Internet Protocol (IP) suite. It also provides an overview of Switched Multimegabit Data Service (SMDS) as it pertains to ATM. We describe the historical origins and then offer an overview of packet formats and protocol functions. Next, the text illustrates the operation of the protocol through examples and then summarizes the traffic and congestion control aspects. Finally, we recount how public services utilize these protocols.

THE INTERNET PROTOCOL SUITE, TCP/IP

The origins of the Internet Protocol (IP) suite lie even earlier than X.25. Operating at layer 3, IP supports a number of protocols, most commonly the Transmission Control Protocol (TCP). TCP provides layer 4 transport-type services to applications such as the World Wide Web's Hypertext Transfer Protocol (HTTP). The emergence of the World Wide Web (WWW) as the predominant user interface to "cyberspace" established IP as the de facto standard for internetworking to the corporate desktop, as well as the home office and the residential user. It is an important set of protocols that MPLS was specifically designed to support, and its influence is seen within ATM as well.

Origins of TCP/IP

The U.S. Advanced Research Projects Agency (ARPA) began development of a packet-switched network as early as 1969, demonstrating the first packet-switching capability in 1972. A seminal paper by Vinton Cerf and Bob Kahn in 1974 [Cerf 74] laid the intellectual foundation for a network of networks, called an *internet*. Named the ARPANET, this network steadily grew as more universities, government agencies, and research organizations joined the network. Network designers introduced the Transmission Control Protocol/Internet Protocol (TCP/IP) in 1983, replacing the earlier Network Control Protocol (NCP) and Interface Message Processor (IMP) Protocol. This is essentially the same TCP/IP standard in use today. Also in 1983, administrators split the ARPANET into two parts—a military network and a nonmilitary research network that became the origin of today's Internet. In 1986, the National Science Foundation (NSF) constructed a 56 Kbps network connecting its research sites that became the origin of today's Internet. It included six new supercomputer centers, followed shortly thereafter with an upgrade from 56 Kbps speeds to 1.5 Mbps DS1 speeds in 1988. The Internet formed its own standards body in 1989, called the Internet Engineering Task Force (IETF), as discussed in Chapter 3. In 1990, the National Science Foundation (NSF) embarked upon a program to upgrade the Internet backbone to DS3 speeds (45 Mbps) for supercomputer interconnection. In 1995, the NSF split the Internet into a backbone for supercomputer communication running at OC-3 speeds (150 Mbps) over ATM and a set of network access points (NAPs) where major backbone Internet service providers (ISPs) could interconnect. A significant policy shift also occurred at this time: instead of funding a network for researchers, the NSF gave the funds to research institutions and allowed them to select their own ISPs.

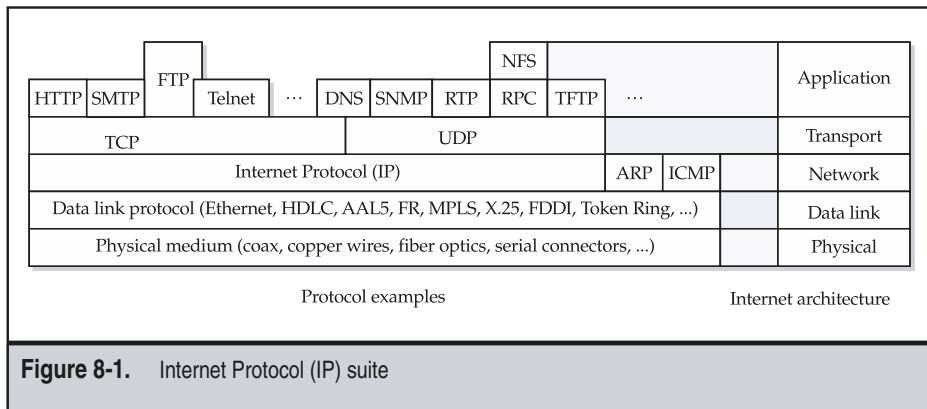
True to the spirit of capitalism, many larger ISPs directly interconnected, bypassing the more expensive government-funded NAPs. In the latter half of the 1990s, the Internet rapidly became a global phenomenon, with almost every nation having many ISPs providing access to enterprises and consumers.

The Internet became a household word beginning in 1995 with the introduction of a user-friendly, multimedia browser application accessing the World Wide Web and providing easy-to-use e-mail access for everyone. Having a Web page became like advertising—every business had to have it to compete. Now, banks, auto dealerships, retail stores, and even children’s television programs beckon users to look at their “Web page.” Online information is the new currency, and the WWW is the ATM machine that provides access to that currency. Gone are the good old days when the Internet was the haven of university researchers and government organizations. If you want to find information, to purchase goods, or just to play computer games against other players—do it on the Web. The Web offers the ultimate in interoperability: all you need is Internet access and a Web browser, and let the Web do all the rest. Let’s take an in-depth look at the foundations of this epitome of interoperability in data networking.

TCP/IP Protocol Structure

Figure 8-1 illustrates the layered Internet Protocol IP suite built atop IP. The User Datagram Protocol (UDP), the Internet Control Message Protocol (ICMP), routing control protocols, and the Transmission Control Protocol (TCP) interface directly with IP, composing the transport layer in the Internet architecture. For further details on TCP/IP, see the IETF RFCs referenced in the following sections, or look through one of the many good books that cover IP in much more depth, such as references Comer 91, Black 92, or Tannenbaum 96.

IP is a datagram protocol that is highly resilient to network failures but does not guarantee sequential delivery. Routers send error and control messages to other routers using the Internet Control Message Protocol (ICMP). ICMP also provides a function in which a



user can send a *ping* (echo packet) to verify reachability and round-trip delay of an IP-addressed host. The IP layer also supports the Internet Group Management Protocol (IGMP), which supports IP multicast.

The Address Resolution Protocol (ARP) directly interfaces to the data link layer, for example, Ethernet. The purpose of ARP is to map a physical address (e.g., an Ethernet MAC address) to an IP address. This is an important concept used in ATM. Chapter 9 covers routing protocols and ARP after this chapter introduces the IP packet format and addressing.

Both TCP and UDP provide the capability for the host to distinguish among multiple applications through port numbers. TCP provides a reliable, sequenced delivery of data to applications. TCP also provides adaptive flow control, segmentation, and reassembly, and prioritized data flows. UDP provides only an unacknowledged datagram capability. The Real Time Protocol (RTP) provides real-time capabilities in support of multimedia applications [RFC1889].

TCP works over IP to provide end-to-end reliable transmission of data across the network. TCP controls the amount of unacknowledged data in transit by dynamically *reducing* either the window size or the segment size. The reverse is also true in that *increased* window or segment size values achieve higher throughput if all intervening network elements have low error rates, support the larger packets, and have sufficient buffering to support larger window sizes.

A number of applications interface to TCP, as shown in Figure 8-1. The File Transfer Protocol (FTP) application provides for secure server logon, directory manipulation, and file transfers. This was an early form of downloading files from the Internet and is still used for large file transfers. Telnet provides a remote terminal logon capability, similar to the old command line interface of terminals to a mainframe. HTTP supports the popular World Wide Web. Most user access now employs this protocol, which runs over TCP as shown in Figure 8-1. Other applications operate over UDP. The Simple Network Management Protocol (SNMP) supports configuration setting, data retrieval, and alarm reporting, and is the most commonly used protocol for collecting management data from IP networked devices. The Trivial FTP (TFTP) protocol provides a simplified version of FTP, which is intended to reduce implementation complexity. The Remote Procedure Call (RPC) and Network File Server (NFS) capabilities allow applications to dynamically interact over IP networks. The Domain Name Service (DNS) provides a distributed or hierarchical name service running over UDP or TCP, which translates user-friendly names, like Web site Uniform Resource Locators (URLs) (e.g., www.mysite.com) and e-mail address domains (e.g., the name after the @ sign in myname@myISP.com) into IP addresses.

TCP/IP Networking Context

Figure 8-2 shows how the Internet Protocol layers operate in a network context interconnecting two end systems—for example, a computer workstation and a server. In the example, three interconnected IP routers interface to two end systems. The data link layer and internetwork layer protocols define procedures for dynamically discovering the “best” route between the end systems in a process called *routing*, a topic covered in some

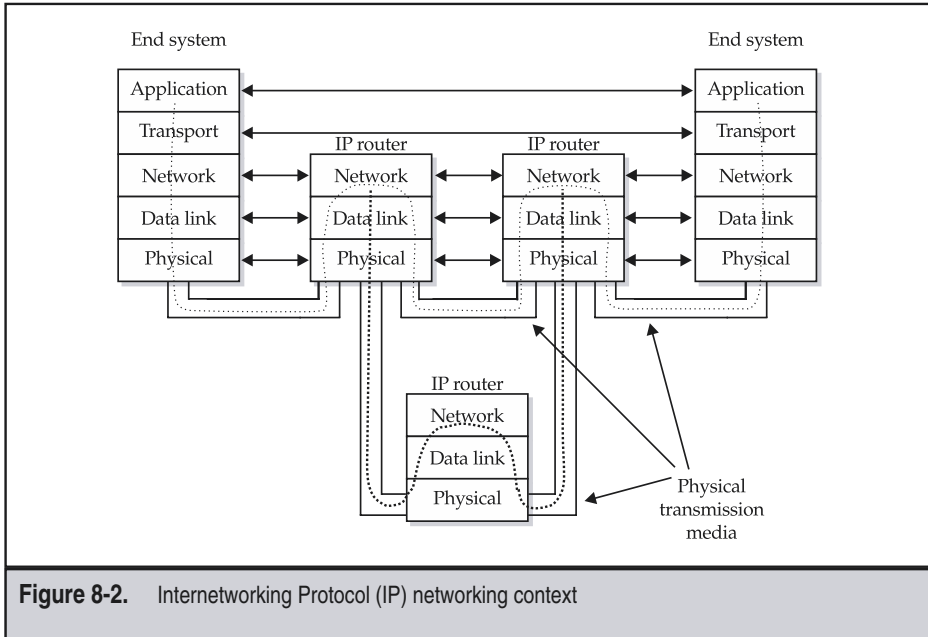


Figure 8-2. Internetworking Protocol (IP) networking context

depth in the next chapter. Packets between the same pair of end systems may not always traverse the same physical path, as indicated by the two sets of dashed lines interconnecting the end systems in Figure 8-2. One path goes horizontally through the two intermediate systems, while the other traverses all three intermediate systems. This occurs because the routers may dynamically discover a better route and change the forwarding path between transmissions of packets by end systems, as illustrated by the dashed line at the bottom of the figure showing how the path can be through the third intermediate system. Each IP router operates at the physical, data link, and network layers, as shown in the figure. The transport and application layers use protocols such as TCP, UDP, FTP, and HTTP, as described in the previous section. Let's take a detailed look at the formats and protocols involved in TCP/IP. Since IP is designed to run over a wide range of data link layer protocols, this section describes only those uniquely defined by the IETF. We then principally focus on the specifics of layer 3—the internetwork layer protocol. Finally, the section concludes with an overview of TCP, which provides such services as sequencing, error detection, and retransmission.

IP provides a connectionless datagram delivery service to the transport layer. IP does not provide end-to-end reliable delivery, error control, retransmission, or flow control; it relies on higher-layer TCP to provide these functions. A major function of IP concerns the routing protocols, which provide the means for devices to discover the topology of the network, as well as detect changes of state in nodes, links, and hosts. Thus, IP

routes packets through available paths and around points of failure. IP has no notion of reserving bandwidth; it only finds the best available path at the moment. Most of the routing algorithms minimize an administratively defined routing “cost.” The next chapter provides further details of routing.

Address design within an IP network is a critical task. A network may have a large number of end-user and server devices, or “hosts.” If every router in a large network needed to know the location of every host attached to every other router, the routing tables could become quite large and cumbersome. A key concept used in routing is that of *subnetting*. Effectively, subnetting breaks the host address space down into multiple subnetworks by masking the bits in the host address field to create a separate subnetwork for each physical network. This means that a router need only look at a portion of the address, therefore dramatically reducing network routing table size. The next chapter presents an example of subnetting.

Generic Link Layer Protocols for IP

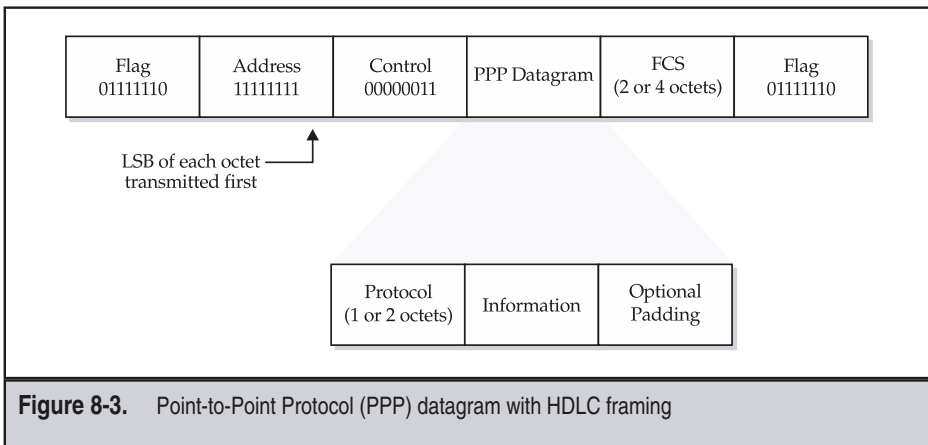
The earliest generic link layer protocol designed specifically for access to the Internet derived from a proprietary protocol implemented by 3Com in the early 1980s to interconnect computers and routers. The predominant versions of UNIX operating systems implemented this simple protocol, called Serial Line IP (SLIP), and the IETF documented it in 1988 in RFC 1055. SLIP provides a means for delimiting IP packets on a serial link using specific characters. If the user data contains these specific characters, then SLIP uses a character-stuffing mechanism similar to the IBM BSC protocol studied in Chapter 4. Since the protocol is so simple, implementations readily achieved a high degree of interoperability. However, SLIP had several shortcomings: it only supported IP, it performed no error detection or correction, and it required preconfiguration of IP addresses. Furthermore, it required assignment of an IP address to every Internet user.

Although these shortcomings weren’t serious problems in the early 1990s, the increasing popularity and sophistication of Internet users required an improved, more feature-rich, data link layer protocol. Also, since the current Internet address space was being rapidly consumed by waves of new users, a means to more efficiently dole out IP addresses was essential. Responding to these needs, in 1994 the IETF specified the Point-to-Point Protocol (PPP), which has now superseded SLIP in most implementations. PPP now supports the automatic configuration and management of the data link layer between dial-up users, as well as multiprotocol routers over a wide range of serial interfaces using data link- and network-level negotiation protocols [RFC1661]. The basic negotiation procedure involves one party proposing a particular option and parameters, followed by the other party either accepting the proposal, or rejecting it.

The PPP Link Control Protocol (LCP) automatically establishes, configures, and tests a connection. The LCP protocol negotiates parameters such as authentication, maximum packet size, performance monitoring, and header compression. The PPP Network Control Protocol (NCP) is specific to each network layer protocol. Ever wonder how ISPs can sign up thousands of users, yet provide the same dial access number to all of them? They

do so by using the IP NCP to dynamically assign one of a prereserved block of IP addresses to each dial-up user as that user calls in on the common access number. Thus, PPP allows an ISP to efficiently share a limited IP address space (a valuable commodity), since only users who are logged on actually use an IP address. Therefore, an ISP must only have a number of IP addresses equal to the number of dial-up ports in use, and not for the total number of subscribers. This feature of PPP stretches the limited address space of the version 4 Internet Protocol (widely used today until replaced by version 6) [Tannenbaum 96].

PPP assumes that some other protocol delimits the datagrams, for example, HDLC as specified in RFC 1662. Figure 8-3 illustrates the basic format of an HDLC-framed PPP datagram. Unfortunately, the IETF uses a different convention for indicating the order of bit transmission than the ISO and ITU HDLC standards use, as shown in the figure. In the IETF notation, the bits of binary numbers are transmitted from right to left. In other words, bit transmission is from Least Significant Bit (LSB), that is, bit zero, to Most Significant Bit (MSB). Standard HDLC flags delimits the PPP datagram. PPP defines an octet-level stuffing method to account for HDLC flags (a binary '01111110') occurring within user data streams. The all-stations HDLC address (all 1's) and the control field encoding identify the HDLC frame as Unnumbered Information with the Poll/Final bit set to zero. PPP uses a standard HDLC frame check sequence (FCS) to detect transmission errors, with RFC 1662 giving an example of an efficient software implementation of the FCS function. Inside the PPP header, the one-octet (optionally two-octet) protocol field identifies the protocol for the particular PPP datagram. Examples of standard protocol fields are LCP, padding protocol, PPP Network Level Protocol ID (NLPID), and various NCP protocol identifiers. Optionally, the network layer protocol may insert a variable-length pad field to align the datagram to an arbitrary octet boundary to make software implementation more efficient.



The authentication feature of PPP allows the network to check and confirm the identity of users attempting to establish a connection. Web browser, operating system, hub, router, and server vendors have embraced PPP as the data link layer of choice for accessing the Internet. IP also operates over a number of other protocols by treating each as a data link layer protocol, such as frame relay, X.25, SMDS, ATM, and MPLS.

IP Version 4 (IPv4) Packet Format

Figure 8-4 illustrates the format of the version 4 IP packet [RFC791, RFC1812]. The 4-bit version field specifies the IP protocol version. Each node first checks the version field before processing the datagram. Such use of the version field will be critical in the migration from IP version 4 to IP version 6. Next, the IP Header Length (IP HL) field specifies the datagram header length in units of 32-bit words, the most common length being 5 words, or 20 octets when no options are present. If options are present, then the IP HL field indicates the number of words used for the options, for example, a route trace. Historically, the 8-bit Type of Service (TOS) field contains a 3-bit precedence field, plus three separate bits specifying other service attributes, and two unused bits. The precedence field ranges from 0 (i.e., normal priority) through 7 (i.e., network control), indicating eight levels of precedence. The three individual bits request low delay, high throughput, and high reliability. Since the semantics of the TOS byte were never precisely defined, a number of incompatible implementations arose. The IETF has redefined this byte for two purposes, detailed as follows: The first is in support of Differentiated Services, called Diffserv for short. The second is for explicit congestion notification by TCP. The Total Length field specifies the total IP datagram length for the header plus the user data.

The identification field, flags, and fragment offset fields control fragmentation (or segmentation) and reassembly of IP datagrams. The Time to Live (TTL) field specifies how many hops the packet can be forwarded in the network before the packet is declared “dead” and hence disposable. Typically, intermediate nodes or routers decrement the TTL field at each hop. When TTL reaches zero, intermediate nodes discard the packet. Therefore, a packet cannot circulate indefinitely through a complex set of networks.

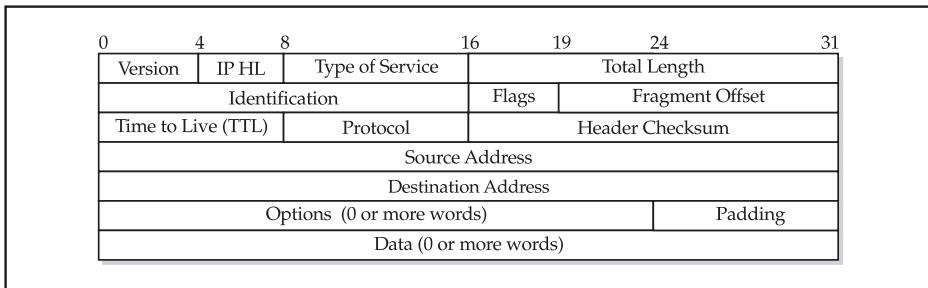


Figure 8-4. IP version 4 (IPv4) datagram format

The protocol field identifies the higher-level protocol type (e.g., TCP or UDP), which then specifies the format of the data field. The header checksum ensures integrity of the header fields through a simple bit-parity check that is easy to implement in software.

The one-word source and destination IP addresses are required fields discussed in more detail in the next section. The options field can specify security level, designate source routing, or request a route trace. Two types of source routing option give either a complete list of routers for a complete path, or a list of routers that must be visited in the path. In the route trace, each intermediate router adds its own IP address to the packet header options field (increasing the IP Header Length field, of course). The options field can request that each router add a timestamp as well as its IP address when performing a route trace. The data field contains higher-layer protocol information or user data.

Internet Protocol (IP) Addressing

The Internet currently uses a 32-bit global network addressing scheme. Each user, or “host” in Internet parlance, has a unique IP address that is 4 octets in length, represented in the following dotted decimal notation,

XXX.XXX.XXX.XXX

where XXX ranges from 0 to 255 decimal, corresponding to the range of 00000000 to 11111111 binary. There are 2^{32} , or over 4 billion, possible IP addresses. You would think that this would be enough addresses, but the Internet recently had to drop a previously used strict hierarchical structure that grouped addresses into three classes: A, B, and C. The class determined the maximum network size, measured by the number of hosts. Although network address classes no longer formally exist in the Internet today, we briefly describes them as an introduction to classless addressing. Table 8-1 illustrates the

IP Address Class	Bits for Network Address	First-Byte Network Address Range	Total Number of Networks	Bits for Host Address	Hosts per Network
A	7	0–127 ¹	126	24	16,777,214
B	14	128–191	16,384	16	65,534
C	21	192–223	2,097,152	8	254
D ²	NA	224–254	NA	NA	NA

Notes: ¹Values 0 and 127 are reserved for the all zeros and all ones addresses.
²Values reserved for multicast.

Table 8-1. Characteristics of Legacy Internet Address Classes

key properties of the legacy Internet address classes. The IETF also reserved another block of Class D addresses for multicasting. Note that the high-order bits represent the network address, while the remaining bits represent the hosts in the network.

A central authority, the Internet Assigned Numbers Authority (IANA), was established to assign IP addresses to ensure that they were unique. This responsibility has been delegated to the following organizations for distribution to ISPs within geographic regions: APNIC (Asia-Pacific Network Information Centre), ARIN (American Registry for Internet Numbers), and the RIPE (Réseaux IP Européens) Network Coordination Centre (NCC). Once an IP address prefix has been assigned, a network administrator then assigns the host addresses within its class A, B, or C address space however he or she wishes. Reusing IP addresses is a bad idea, since address conflicts arise when networks are interconnected. This assignment scheme worked fine, until the public's attraction to the information content available on the Web made the Internet enormously popular. The problem arose because anyone asking for an IP address asked for the largest size they could justify. In the early days of the Internet, the address space seemed limitless, so the administrators generously assigned larger address classes to any user request. This inefficient allocation meant that a lot of IP addresses went unused. Many organizations further exacerbated the problem by requesting Class A and B addresses using PPP and then assigning them inefficiently, leaving large blocks of addresses unused. The Internet community attempted to resolve the crisis a bit late by reallocating unused Class A and C addresses to the many new service providers, who then dynamically allocated addresses using PPP to individual users clamoring to get on the Internet. Now, modern routing protocols treat IP addresses as 32-bit numbers without a class structure.

The Internet called this new scheme Classless Inter-Domain Routing (CIDR) and began deployment in 1993 [RFC1817, RFC1518, RFC1519] to improve scaling of the Internet routing system. CIDR generalizes the concept of variable-length subnet masks (VLSMs), thus eliminating the rigid historical structure of network classes (A, B, and C). Interior (intradomain) routing protocols supporting CIDR are OSPF, RIP II, Integrated IS-IS, and E-IGRP. Only one exterior (interdomain) routing protocol, BGP-4, currently supports CIDR.

Next Generation IP—IPv6

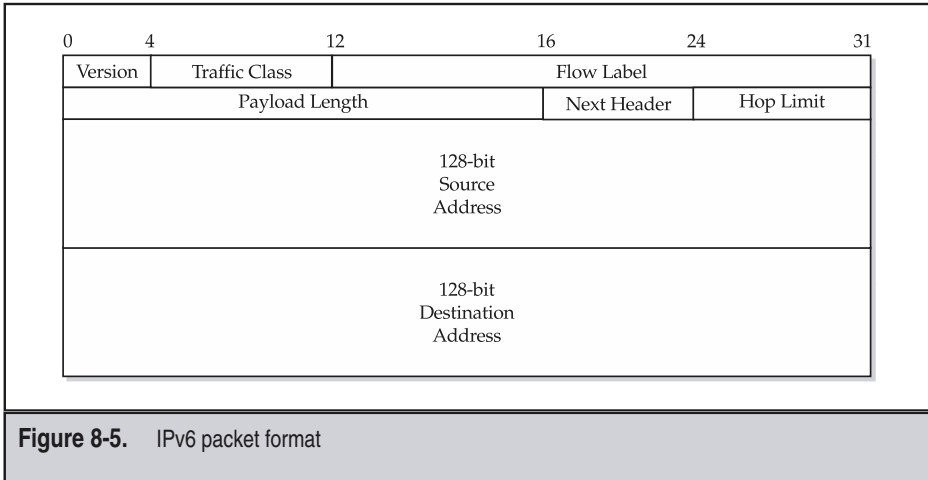
It is true that “the Internet is running out of addresses!” The more efficient allocation of IPv4 addresses through CIDR, along with the availability of routing protocols that support the variable-length subnet masks, will provide for only a limited number of years for Internet growth. For example, CIDR is now moving into the Class A address space reserved for expansion (64.0.0.0 through 126.0.0.0) [RFC1817]. Anticipating the exhaustion of the IP address space, the IETF issued a call for proposals for a successor to the current Internet Protocol in 1992 [RFC1550]. The twenty-one original proposals were reduced to a short list of seven serious proposals, resulting in the recommendation of RFC 1752 [RFC1752] for an IP next generation (IPng) protocol in January 1995. IPng supersedes IPv4 and is now formally referred to as IPv6. The IETF issued the primary RFCs for IPv6 in December 1995. These include RFC 1883 for general definitions of IPv6 [RFC1883], RFC

1884 for addressing [RFC1884], RFC 1885 for ICMPv6 [RFC1885], and RFC 1886 for Domain Name Services (DNS) extensions [RFC1886]. IPv6 contains the following additions and enhancements over IPv4:

- ▼ Expansion of the address field size from 32 to 128 bits
- Simple dynamic autoconfiguration capability
- Easier multicast routing with addition of a “scope” field
- An anycast feature, where a host sends a packet to an anycast address, which the network delivers to the closest node supporting that function
- Capability to define quality of service for individual traffic flows using the resource reservation protocol (RSVP)
- Reduction of overhead by making some header fields optional
- More flexible protocol design for future enhancements
- Authentication, data integrity, and confidentiality options
- Easy transition and interoperability with IPv4
- ▲ Support for all IPv4 routing algorithms (e.g., OSPF, RIP, and BGP)

The new IPv6 supports all the traditional protocols that IPv4 did, such as datagram service; FTP file transfers; e-mail; X Window; Gopher; and, of course, the Web. Furthermore, IPv6 also supports approximately 340×10^{36} individual addresses. To ease migration, IPv4 addressing is a proper subset of the IPv6 address space.

Figure 8-5 illustrates the version 6 IP packet format [RFC1883]. The *Version* field allows routers to examine the first four bits of the packet header to determine the IP version. The IPv4 packet header allocates a version field in the first four bits as well, so that routers can support both IPv4 and IPv6 simultaneously for migration purposes. The 8-bit *traffic class* field is identical in format to the redefinition of the IPv4 TOS byte for Diffserv and TCP ECN. The *Flow Label* field is intended for use by protocols such as the resource reservation protocol (RSVP) (described later in this chapter) to guarantee bandwidth and QoS for streams of packets involved in the same flow. That is, as part of a signaling protocol, a flow label would be assigned such that a router could look at this shorter field to determine what processing to apply. The *Payload Length* field indicates the number of bytes following the required 40-byte header. The *Next Header* field identifies the subsequent header extension field. There are six (optional) header extensions: hop-by-hop options, source routing, fragmentation support, destination options, authentication, and security support. The last extension header field identifies the higher-layer protocol type using the same values as IPv4—typically, TCP or UDP. The *Hop Limit* field determines the maximum number of nodes a packet may traverse. Nodes decrement by 1 the *Hop Limit* field each time they forward a packet, analogous to the Time to Live (TTL) field in IPv4, discarding the packet if the value ever reaches zero. The source and destination addresses are both 128 bits in IPv6, four times as large as the address fields used in IPv4. Therefore, the required IPv6 header is 40 bytes; however, the optional extension header fields can make the overall header considerably larger.



Although the IETF continues to update standards so that new and modified protocols can use IPv6, commercial adoption of IPv6 has been very slow. The use of CIDR and more judicious use of IPv4 address space have reduced the need for IPv6's broadly expanded address space. Some experts predict that a proliferation of IP-addressable devices—for example, widespread wireless Internet access or home-area networking, will—eventually create a need for IPv6.

Quality of Service in IP Networks

Traditionally, the Internet offered only a single QoS, best effort, with available capacity, delay, and loss characteristics dependent on instantaneous load and the state of the network. Network designers controlled QoS by provisioning routers, links, and routing parameters according to historical traffic patterns. When the only Internet users were researchers, universities, and government agencies, best-effort performance was adequate. This was true because much of the communication was non-real-time, for example, e-mail and file transfer. Interestingly, the use of a limited multicast backbone called MBONE to broadcast portions of IETF meetings over the best-effort Internet of the early 1990s was a principal driver for adding QoS awareness to IP [RFC1633]. During intervals of congestion, video and audio applications may not work when provided best-effort service.

Many industry analysts believed that a significant differentiator between IP and ATM was Quality of Service (QoS). However, the IETF has been active in developing standards to add QoS capabilities to IP. The earliest effort was called Integrated Services (Intserv). It was based upon the resource reservation protocol (RSVP), which allows application sessions to signal requests for different levels of quality for a specific amount of traffic. This per-flow signaling paradigm was at odds with the connectionless nature of IP and created scalability challenges for IP routers. Consequently, it has not been widely adopted.

Responding to these concerns, the IETF then developed a stateless, per-packet approach to QoS by redefining the TOS byte in the IP header in the Differentiated Services (Diffserv) standard. This section describes the Intserv and Diffserv approaches to IP QoS.

Integrated Services (Intserv) and RSVP

There is an old saying that “when everything is high priority, there is no priority.” Analogously, RSVP over IP only delivers QoS if the underlying layer 2 network delivers QoS. The most straightforward way to assure QoS in the underlying layer 2 network is to ensure that it isn’t congested. In the LAN, Switched Ethernet often fills the bill by allocating a dedicated 10 or 100 Mbps segment to each server or client host. RSVP grew out of work first published in 1993 by researchers at Xerox and the University of Southern California (USC) [Zhang 93]. The IETF established the Intserv working group [RFC1633], which resulted in publication of version 1 of the RSVP specification in 1997 (after over a dozen drafts contributed to RFC 2205). It is interesting to note that in the applicability statement for RSVP, the IETF recommends that RSVP be limited to small numbers of multimedia users in intranets only. Furthermore, RSVP does not scale well on backbone trunks supporting large numbers of reservations. Another protocol must serve the aggregate backbone bandwidth reservation and QoS guarantee requirements. RSVP also needs some mechanism for deciding which applications get reservations for high-quality resources; otherwise, applications (or their human controllers) will indiscriminately ask for the highest quality service available. Recent work in the IETF defines policy servers—which either approve or reject individual RSVP requests. Also, the IETF specified some basic security measures in RFCs 2206 and 2207 to mitigate the chance of false reservations or service theft.

Okay, you say, now that I have this background, tell me what RSVP is. How will it impact me? And why should I care? The answer is that extensions to RSVP are used within MPLS standards to establish label switched paths (LSPs). The remainder of this section presents some background on the basic components and operation as an introduction to a description of the MPLS-specific extensions to RSVP covered in Part 3.

RSVP is a resource reservation setup (also called “signaling”) protocol specifically designed for an integrated services Internet that supports end-to-end QoS. Unlike the connection-oriented services studied so far, RSVP employs receiver-initiated setup of resource reservations for multicast or unicast data flows.

Receiver applications in IP hosts utilize RSVP to indicate to upstream nodes their traffic requirements in terms of bandwidth availability, acceptable jitter, and available buffer space. RSVP is not a routing protocol; but like routing protocols, it operates in the control plane, not in the data forwarding path shown by the solid arrow and shaded boxes in Figure 8-6.

A *packet classifier* in each RSVP-capable device utilizes a filter specification based upon fields in the L3 and possibly L4 packet headers that identify a flow to determine the QoS class of incoming data packets; it then selects the route (i.e., next hop). Each node also utilizes a *packet scheduler*, employing methods such as packet-level traffic shaping, priority queuing, and weighted fair queuing to achieve the requested QoS through that node.

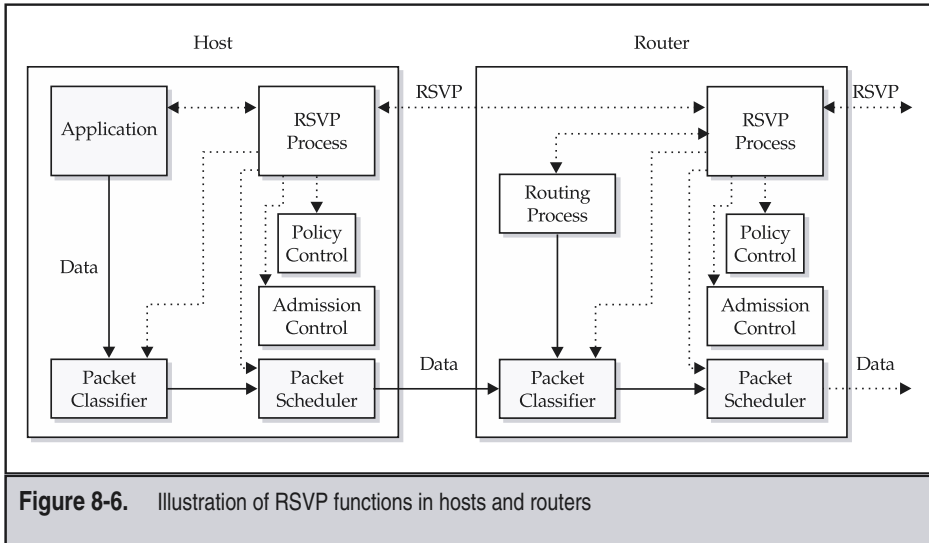


Figure 8-6. Illustration of RSVP functions in hosts and routers

The admission control function, resident in RSVP-aware nodes along the path between the destination and the source, interprets the flow specification in RSVP control packets and determines whether the node has sufficient resources to support the requested traffic flows. The node may also perform policy control, for example, to ascertain that the requester has the right to make such reservations. If either the admission or policy control process checks fail, the RSVP process in the node returns an error message to the requesting application. If the node does not accept a request, it must take some action on the offending flow, such as discarding the packets or treating them at a lower priority. Thus, every intermediate node must be capable of prioritized scheduling and selective discard. If a layer 2 network interconnects layer 3 devices (e.g., routers), then it, too, may need to be QoS aware. This is an important point, as many IP networks span multiple data link layer types and intermediate networks, making a ubiquitous, end-to-end implementation of RSVP unlikely. One device or data link layer in the midst of an end-to-end RSVP flow may invalidate any traffic or quality-level guarantees. RSVP-capable systems maintain “soft-state” about RSVP traffic flows; that is, the state will time out and be deleted unless it is periodically refreshed.

IETF RFC 2211 defines the controlled-load service that applies to the portion of a flow that conforms to a Traffic Specification (Tspec). RFC 2211 defines the end-to-end behavior observed by an application for a series of network elements providing controlled-load service in terms of the behavior visible to applications receiving best-effort service “under unloaded conditions” from the same series of network elements. Although this definition may seem somewhat vague, it is not. As shown in Part 6, the discipline of queuing theory defines this notion rather precisely in terms of the likelihood of packet delivery and the statistics regarding the delay incurred. Specifically, RFC 2211 states that “a very high

percentage of transmitted packets will be successfully delivered by the network.” RFC 2211 also states that “the transit delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum transmit delay.”

IETF RFC 2212 states that Guaranteed QoS applies to only the portion of a flow that conforms to a Traffic Specification (Tspec) and a desired service (Rspec) parameter. The Rspec defines the actual service rate R delivered by the network and some additional terms that precisely bound the actual delay delivered to the flow. Interestingly, the Guaranteed QoS protocol bounds the maximum delay encountered by an individual flow by exchanging information in RSVP messages. Note that this maximum end-to-end delay will not change as long as the IP routing path remains constant. However, Guaranteed QoS does not control minimum or average delay. As noted earlier in this chapter, applications like video and audio playback require bounded delay variation, which Guaranteed QoS supports.

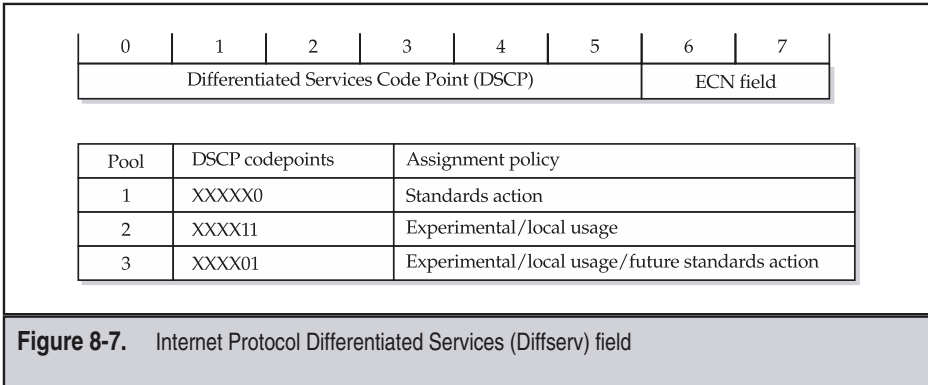
Differentiated Services (Diffserv)

The IETF has specified an approach for the differentiated services in the Internet called *Diffserv*. It aims to provide services differentiated on performance utilizing weighted priority queuing and quasi-statically configured routing [RFC2475, RFC2474]. In *Diffserv*, the performance (i.e., QoS) measures to which differentiated service applies are responsiveness and availability. *Diffserv* requires that edge routers classify traffic flows into a member from a set of categories based upon the TCP/IP header fields in what is called a *microflow*. *Diffserv* utilizes a standard field within the IPv4 or IPv6 header to indicate the result of this classification.

Figure 8-7 illustrates the redefinition of the TOS byte defined in RFCs 2474 and 3168. This one-byte header is present in the IPv4 or IPv6 packet header as summarized previously. Because this byte is present in every IP packet header, each node can provide differentiated services on a per-hop basis. What is necessary is a standard interpretation of what these per-hop behaviors are. RFC 2474 requires that implementations must match the entire six-bit Differentiated Services Code Point (DSCP) when determining the packet-handling mechanism necessary to provide a Per Hop Behavior (PHB). These PHBs are the building blocks from which an end-to-end differentiated service can be constructed. As indicated in Figure 8-7, specific bit patterns within the DSCP field are the subject of standardization, experimental usage, or local usage by an Internet service provider. The low-order two bits of the TOS byte are defined in RFC 3168 for TCP Explicit Congestion Notification (ECN), as described later in this chapter.

RFC 2474 assigns DSCP codepoints of the form “XXX000” from Pool 1 to a class selector codepoint. These codepoints provide backward compatibility to the IP Precedence field. A packet with a class selector codepoint having a higher numerical value should experience a higher probability of timely forwarding than a packet having a class selector codepoint of a lower numerical value. Furthermore, an RFC 2474-compliant implementation must implement at least two independently forwarded classes of traffic.

Currently, *Diffserv* is unidirectional, allowing a user to specify performance separately in each direction. The envisioned services include the concepts of a traffic conditioning



agreement (TCA) and a service-level agreement (SLA). The traffic and performance specifications may vary not only by direction, but also by geographic region and time of day. The parameters may be qualitative or quantitative. An example of a qualitative service is simple prioritization. An example of quantitative service uses traffic parameters analogous to those defined for RSVP, namely, the peak and average rates along with a burst size. Quantitative performance parameters are also analogous to RSVP, and include latency and packet loss.

The network must provision its routing and PHBs in response to the contracted traffic levels and historical traffic patterns. Currently, the Diffserv approach primarily considers long-term allocation and does not address the issue of dynamic reservation. Examples of PHBs include an Assured Forwarding (AF) group that specifies four subgroups, each with selective discard preference as a means of congestion avoidance [RFC2597] and an Expedited Forwarding (EF) behavior [RFC2598] intended for applications that require low loss, delay, and delay variation.

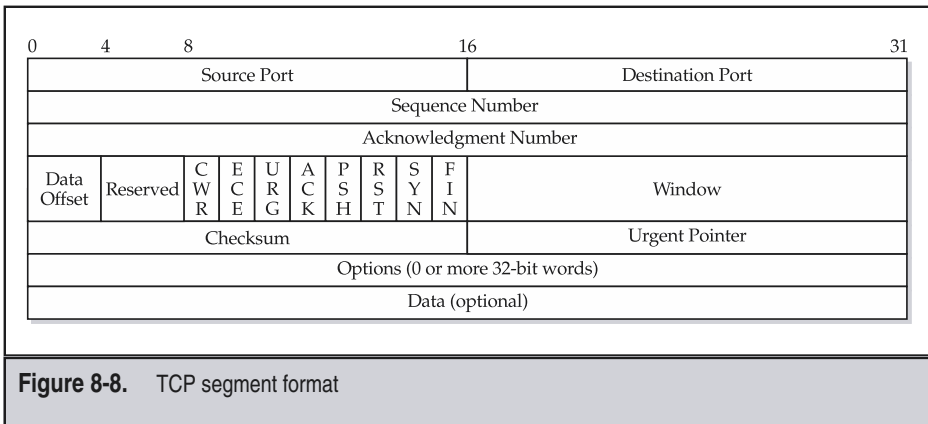
An ISP then defines a Per Domain Behavior (PDB), which is a concatenation of PHBs [RFC3086]. The objective is that as ISPs form service-level agreements between themselves using PDBs, differentiated service will eventually spread across the Internet. Chapter 20 provides more details on Diffserv.

Transmission Control Protocol (TCP)

Now, our coverage of TCP/IP moves up to the transport layer, namely, the Transmission Control Protocol (TCP). Figure 8-8 illustrates the version 4 TCP packet format and meanings specified in RFC 793 and RFC 1122. TCP employs the source and destination port numbers to identify a specific application program running in the source and destination hosts. The 16-bit port number in conjunction with the 32-bit host address form the 48-bit *socket* identifier. Port numbers less than 256 are called “well-known ports” as defined in RFC 1700 and are reserved for standard services. For example, the *Sequence Number* field identifies the position of the sender’s byte stream in the data field. The *Acknowledgment*

Number field identifies the sequence number of the next byte expected at the receiver. The *Data Offset* field tells how many 32-bit words are in the TCP header. The default header length is five words, as shown in Figure 8-8. The code bits field contains eight bits: CWR, ECE, URG, ACK, PSH, RST, SYN, and FIN. URG indicates that the Urgent Pointer is used. The ACK bit indicates that the Acknowledgement Number field is valid. The PSH (i.e., push) bit indicates that TCP should deliver the data to the destination port prior to filling an entire software buffer in the destination host. The RST bit indicates that the connection should be reset. TCP also uses the RST bit to reject invalid segments and refuse a connection attempt. TCP employs the SYN bit to establish connections and synchronize sequence numbers. The FIN bit releases a TCP connection. The CWR and ECE bits are used in TCP explicit congestion notification, as described later in this section. The 16-bit *Window* field identifies the amount of data the application is willing to accept, usually determined by the remaining buffer space in the destination host. The *Checksum* applied across the TCP header and the user data detects errors. The *Urgent Pointer* field specifies the position in the data segment where the urgent data begins—if the URG code bit indicates that this segment contains urgent data. The options field is not mandatory but provides additional functions, such as a larger window size field as specified in RFC 1323. Another popular option is selective retransmission, as specified in RFC 1106, instead of the default go-back-n protocol—an important protocol difference that dramatically increases throughput on channels with excessive bit errors or loss, as studied in Chapter 25. The padding field aligns the TCP header to a 32-bit boundary for more efficient processing in hosts.

TCP is a connection-oriented protocol and therefore has additional, specific messages and a protocol for an application to request a distant connection, as well as a means for a destination to identify that it is ready to receive incoming connection requests.



Example of TCP/IP Operation

Figure 8-9 shows an example of a TCP/IP network transferring data from a workstation client to a server. TCP assumes that the underlying network is a connectionless datagram network (for example, IP) that can deliver packets out of order, or even deliver duplicate packets. TCP handles this by segmentation and reassembly using the sequence number in the TCP header, while IP does this using the fragment control fields in the IP header. Either method, or both, may be used. The client on the left-hand side of the figure segments the user data ABCD into four TCP segments. Router R1 initially routes IP datagram A via the X.25 network, as shown in Figure 8-9. R1 then becomes aware of a direct connection to the destination router, R2, and routes the remaining datagrams (B, C, and D) via the direct route. This routing action causes the datagrams to arrive at the destination server out of order, with datagram A traversing the much slower X.25 network and arriving significantly later. On the right-hand side of Figure 8-9, the TCP stack running in the server resequences the datagrams and delivers the block of data to the destination socket in the original order. IP performs a similar process using fragmentation and reassembly on an individual packet basis.

This operation by TCP/IP of accepting datagrams out of order, and being able to operate over an unreliable underlying network, makes it quite robust. No other standard modern data communication protocol has this attribute.

Traffic and Congestion Control Aspects of TCP

TCP works over IP to achieve end-to-end reliable transmission of data across a network. TCP flow control uses a sliding window flow control protocol, like X.25; however, the window is of a variable size, instead of the fixed window size used by X.25. The tuning and refinement of the TCP dynamic window flow control protocol has been the subject of a great deal of research. The following example is a simple case of the Van Jacobson "Slow Start" TCP algorithm [RFC2001], an enhancement to the initial TCP implementation designed to dynamically maximize throughput and prevent congestion collapse. TCP keeps track of a congestion window, which is never greater than the maximum window size reported by the receiver in the TCP packet shown in Figure 8-8.

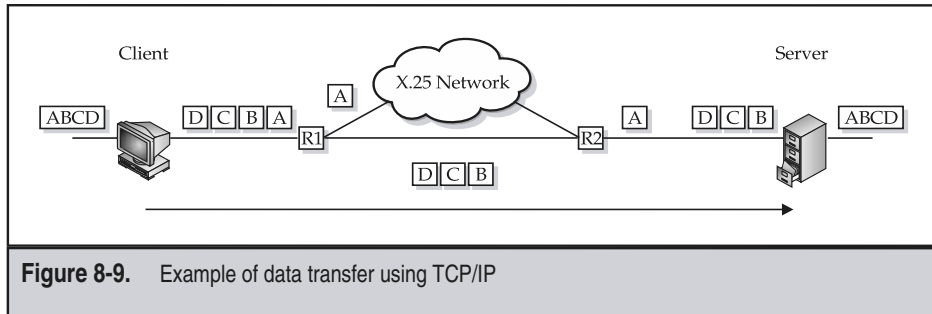


Figure 8-9. Example of data transfer using TCP/IP

Figure 8-10 illustrates a simplified example of key concepts in the dynamic TCP window flow control protocol between a workstation sending to a server. The sender starts with a congestion window size equal to that of one TCP segment (segment 0). The IP network delivers TCP segment 0 to the destination server, which acknowledges receipt. The sending workstation then increases the congestion window size to two segments. When the destination acknowledges both of these segments, the sender increases the window size to four segments, doubling the window size for each received acknowledgment. At this point, the IP network becomes congested and loses the fifth and sixth segments (the figure indicates the lost packets by X's). The sender detects this by starting a timer immediately after sending a segment. If the timer expires before the sender receives an acknowledgment from the receiver, then the sender retransmits the segment. Upon such a retransmission timeout, the sender resets its window size to one segment and begins the process again. The timeout may be immediate—or, typically, 500 milliseconds—in an attempt to “piggyback” the acknowledgment onto another packet destined for the same socket identifier.

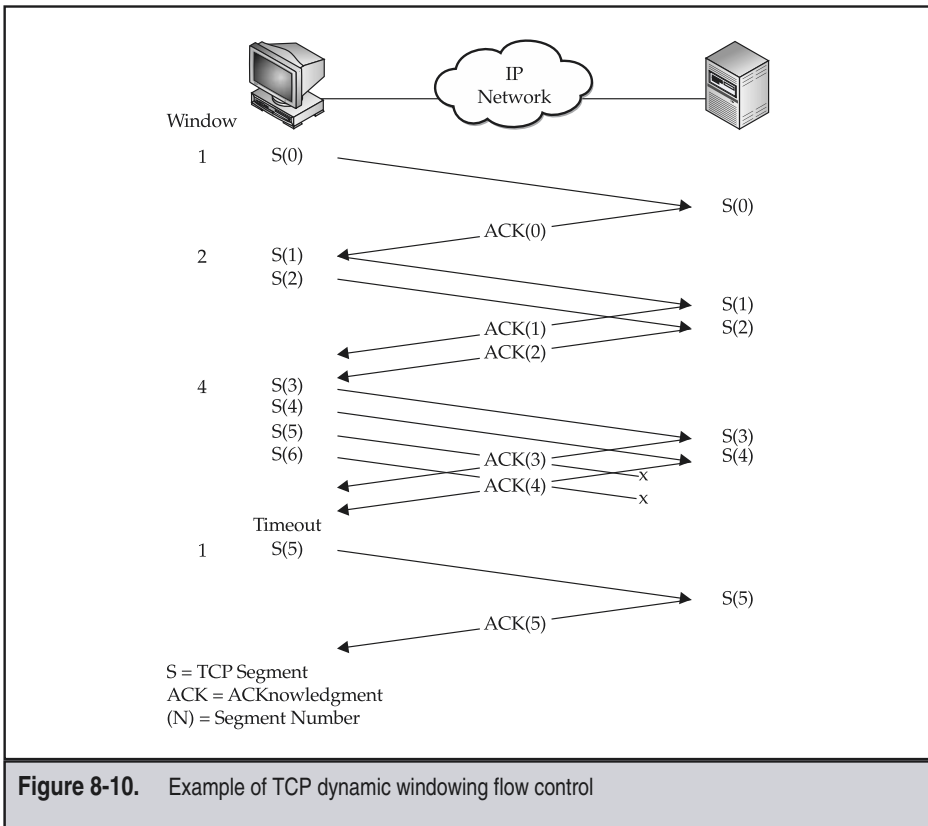


Figure 8-10. Example of TCP dynamic windowing flow control

Note that TCP actually uses byte counts and not individual segments; however, if all segments are of the same size, then our simple example is accurate. The astute reader will observe that TCP is geometrically increasing the window size (i.e., 1, 2, 4, 8, and so forth), so this is not really a slow start at all. TCP congestion control has another function that limits the interval of geometric increase until the window size reaches a threshold of one half the congestion window size achieved before the previous unacknowledged segment. After this point, the TCP increases the congestion window by one segment for each round-trip time, instead of doubling it during the geometric growth phase, as shown in the preceding example. This linear phase of TCP window growth is called *congestion avoidance*, while the geometric growth phase is called *slow start*.

Hence, during steady-state operation when there is a bottleneck in the IP network between the sender and the receiver, the congestion window size at the sender has a sawtooth-like pattern, as illustrated in Figure 8-11. In this example, the segment size is constant, and the timeout occurs when the congestion window size reaches 16 packets. This type of oscillating window behavior occurs when multiple TCP sources contend for a bottleneck resource in an IP network—for example, a busy trunk connecting routers in the Internet or an access circuit connecting to a popular Web server. Since the World Wide Web's HTTP protocol runs over TCP, this phenomenon occurs during intervals of congestion on the Internet.

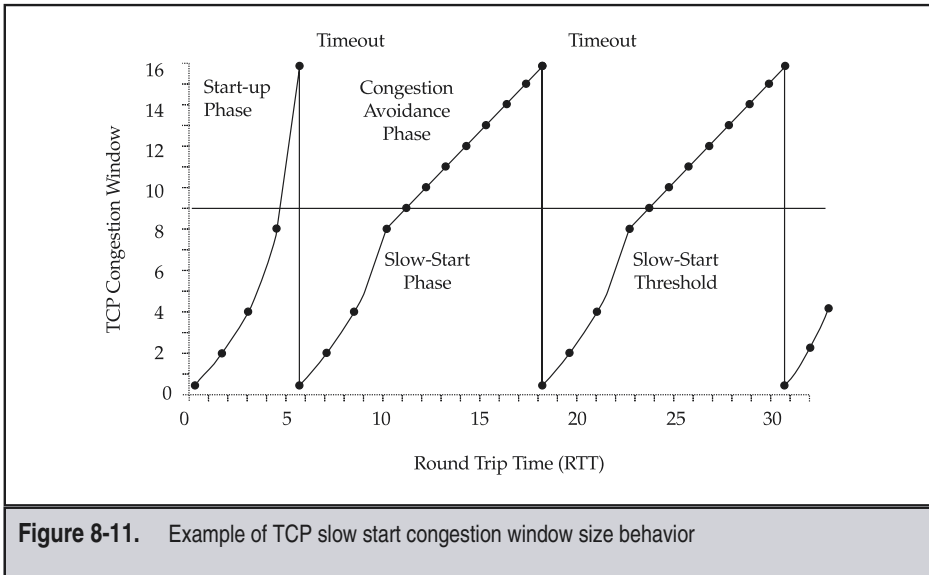


Figure 8-11. Example of TCP slow start congestion window size behavior

Because the TCP adaptive windowing protocol operates in units of the round-trip time, locations closer together will experience better response time performance because the TCP congestion window increases exponentially in units of the RTT during the slow start phase. Since a typical Web transaction involves 15–20 packets, the sum of the congestion window values for TCP slow start for the first five RTTs is 31 (i.e., $1 + 2 + 4 + 8 + 16$). This means that TCP requires four to five round-trip times to transmit the packets associated with a typical Web page. If the RTT is 100 ms, then TCP adds a perceptible half-second delay to response time. On the other hand, if the RTT is only 10 ms, then the additional 50 ms of response time that TCP slow start adds to response time is barely noticeable. This is the reason why many Web sites request that users identify the site closest to their location when downloading a large file. RFC 2001 describes further enhancements to TCP called *fast retransmit* and *fast recovery*. TCP generates a duplicate acknowledgment whenever it receives an out-of-order segment, since it is acknowledging the last byte received in sequence. Fast retransmit uses duplicate acknowledgments to detect a lost segment and attempts to avoid the throughput-robbing reduction of the congestion window size to one after a timeout by retransmitting only the lost segment. Fast recovery follows after the transmitter sends the missing segment, reverting to the linear window size increase algorithm of congestion avoidance even if the transmitter is in the slow start phase.

Unfortunately, TCP's automatically adjusted window size can cause a phenomenon called *synchronization*, or phasing, where the sawtooth patterns from multiple TCP sources all line up at some congested point in the Internet and create significant queuing, and hence greater variability in response time. In order to avoid congestion at routers, researchers at Berkeley invented an algorithm called *Random Early Detection (RED)* [Floyd 93]. The objective of RED is to fairly distribute the effects of congestion across multiple user flows competing for a congested router port by randomly discarding TCP traffic flows when a time-averaged buffer level exceeds a predetermined threshold. Many routers now implement RED for TCP flows because it improves the overall performance observed by end users. Other refinements and enhancements of the RED algorithm include dropping all packets from a randomly identified flow, implementing thresholds on a per-flow basis, and weighting the drop probability in accordance with prioritization across different flows.

Another active queue management protocol has been standardized by the IETF in RFC 3168 for TCP, called *Explicit Congestion Notification (ECN)* [Floyd 94]. ECN uses the low-order two bits of the TOS byte shown in Figure 8-7 for an end system to indicate whether it does or does not support ECN-Capable Transport (ECT), as well as a means for a router to indicate Congestion Experienced (CE). A basic idea of ECN is for routers to set the CE codepoint before discarding ECT-marked packets. This provides motivation for end systems to implement ECN. An ECN-capable TCP end system that receives a CE-marked packet responds by setting the ECN Echo (ECE) bit in the TCP header of the next ACK headed toward the sender. The TCP sender uses the CWR flag to acknowledge receipt of the ECE bit. A TCP sender treats the receipt of ECN congestion notification (i.e., by receipt of the ECE bit in the TCP flags) as if it were a single lost packet. That is, the TCP sender would reduce its window size, as shown in Figure 8-11. See www.aciri.org/floyd/red.html for more information on RED and ECN.

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) [RFC768] requires no message exchange between the sender and receiver before sending data; in contrast with TCP, it is connectionless. The principal function performed by UDP, and a function performed by TCP as well, is that of application-level multiplexing. UDP does this using the header shown in Figure 8-12. The combination of source and destination port numbers with the source and destination IP addresses creates a unique association between the sending and receiving applications. The UDP Length field gives the total number of bytes in both the UDP header and the UDP data field. If used, the UDP Checksum field is the one's complement sum of the fields in the UDP Pseudoheader field, the UDP Header, and the UDP Data, where it performs an error check on the data delivered to the destination application. If the UDP Checksum field is not used, it is set to zero, and the receiver performs no error checking on the delivered data.

UDP is more efficient than TCP for transport of multimedia data—its header is only 8 bytes, while the TCP header is 20 bytes in length. However, there is a price paid for such simplicity, in that the application layer must sometimes perform additional functions when operating over UDP. For example, the Real-Time Transport Protocol (RTP) must perform additional timing and buffering functions when transporting voice and video. Also, it is important to note that since UDP has no congestion detection or flow control, unregulated UDP applications could steal capacity from better-behaved TCP applications that use flow control.

As mentioned earlier, a number of important protocols use UDP—for example, SNMP for network management; DNS for name-to-address lookups; and RTP for voice and video over IP, as described in the next section.

Real-Time Transport Protocol (RTP)

RFC 1889 defines the Real-Time Transport Protocol (RTP), which supports services with real-time characteristics, such as interactive voice and video. RTP performs functions including payload type identification, sequence numbering, time stamping, and delivery monitoring. Typically, applications run RTP over UDP and can operate in either a point-to-point or multicast mode. Since RTP runs only on end systems, it empowers the application to determine what to do with the sequencing and spacing of IP packets it

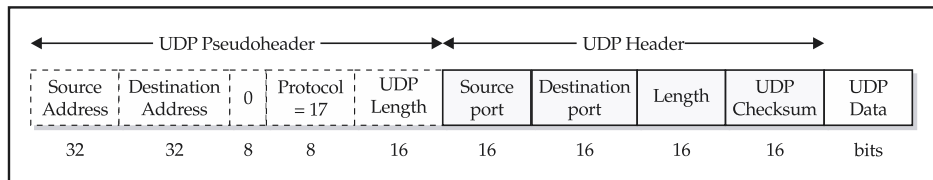


Figure 8-12. User Datagram Protocol format

receives. For example, a receiver may use the RTP sequence numbers to discard out-of-order packets, or to reconstruct the transmitted packet sequence by using a reassembly playback buffer. RTP operates in concert with the RTP Control Protocol (RTCP) to monitor delivery performance and convey some information about participants in an active session.

RFC 1889 states that RTP is a protocol framework designed for extensibility. The base protocol implements a number of functions, which are best understood by examining the contents of an RTP packet shown in Figure 8-13. The Flags and Options field contains information about the protocol version, indicates the presence of padding (for example, when block encryption is used), identifies header extensions, and enumerates the optional Contributing Source (CSRC) identifiers included in the packet. It also contains a marker bit that may be used to delineate framing boundaries within the packet stream. The Payload Type field identifies the RTP payload format according to a profile associated with a specific assigned number. A single RTP stream uses only a single payload type. The next two fields provide services to higher-layer applications. The source increments the Sequence Number field for each RTP data packet transmitted, beginning with a random value to complicate security attacks. It is often used by the receiver to detect packet loss or to perform reordering to restore the transmitted packet sequence. The Timestamp field indicates the instant at which the first byte of the RTP payload was generated, with time granularity determined by the profile. An application may use the timestamp to eliminate packet jitter introduced by the datagram network or restore proper timing during playback. A randomly chosen Synchronization Source (SSRC) identifier identifies the sequence number and timestamp grouping for use by the receiving application(s). Up to 32 CSRC identifiers follow this field if the application uses a mixer—for example, an audio conference. Optional header extensions precede the payload, which may be followed by a single byte indicating the pad length if this option was specified in the options field.

If RTP is used in conjunction with an IP network that supports RSVP or Diffserv, performance can actually be quite good. However, when used over a best-effort IP network, RTP-based application performance may degrade during periods of congestion. Fortunately, RTCP provides a means for applications to measure performance and potentially reroute traffic under these conditions. For example, some Voice over IP networking solutions measure performance using RTCP, and, if performance is too poor, the system temporarily routes calls over more expensive telephone network circuits during periods of poor IP network performance.

Flags/ Options	Payload Type	Sequence Number	Time Stamp	SSRC ID	CSRC ID(s)	Header Extensions	Payload	Pad Length
9	7	16	32	32	32		Variable	8 bits

Figure 8-13. Real-Time Transport Protocol (RTP) packet header format

Service Aspects of TCP/IP

Typically, TCP/IP implementations constitute a router, a TCP/IP workstation, server software, and network management. TCP/IP protocol implementations span Windows, UNIX, DOS, VM, and MVS environments. A majority of UNIX users employ TCP/IP for internetworking. Many Network Operating System (NOS) vendors now integrate TCP/IP into their implementations.

Standards define operation of IP over a number of network, data link, and physical layer services. At the network layer, standards define IP operation over X.25 and SMDS. At the data link layer, other standards define IP operation over PPP, frame relay, Ethernet, and ATM. IP operation over circuit-switched and dedicated physical layer facilities is also defined, such as packets over SONET.

Internet service providers (ISPs) provide access to IP users connected via dedicated access lines or dial-up modem pools. Every major country around the world has one or more ISPs. In countries with large volumes of Internet traffic, some ISPs provide backbone transport for other ISPs. Some of these ISPs are directly connected, while many are not. If ISPs are not directly connected, then the IP routing protocols described in Chapter 9 provide the means for packets to traverse multiple networks such that every advertised unique IP address on the planet is reachable from every other IP address.

SWITCHED MULTIMEGABIT DATA SERVICE

ATM adopted a number of concepts originally developed in support of SMDS. As background to the upcoming part on ATM, this section covers the service aspects of SMDS and specifics of the Distributed Queue Dual Bus (DQDB) protocol defined in the IEEE/ISO 802.6 standard.

Origins of SMDS

The idea of metropolitan area networks (MANs) began when the IEEE began work in 1982 on standards for transmission of voice, compressed video, LAN interconnectivity, and bulk-data transfer. It was first presented to the cable television (CATV) community, which didn't tune into the idea. Eventually, a Bell Labs MAN standard proposal developed in parallel with the ex-Burroughs FDDI venture called MST (Multiplexed Slot and Token) became IEEE Project 802.6. The IEEE 802.6 standard is based upon the Distributed Queue Dual Bus (DQDB) technology [ISO8802-6]. The DQDB architecture was invented at the University of Western Australia, and hardware was first produced by QPSX LTD (a University of Western Australia and Telecom Australia spin-off).

As SMDS was created as a MAN service by Bellcore, it is in the purest sense a service definition and *not* a protocol. The first realization of SMDS was defined using the DQDB technology, as specified in the mode of the IEEE 802.6 standard that defined connectionless data-transport service using 53-byte slots to provide integrated data, video, and voice services over a MAN. Although the IEEE 802.6 standard also defines connection-oriented

isochronous services, SMDS supported only a connectionless datagram service primarily targeted for LAN interconnection.

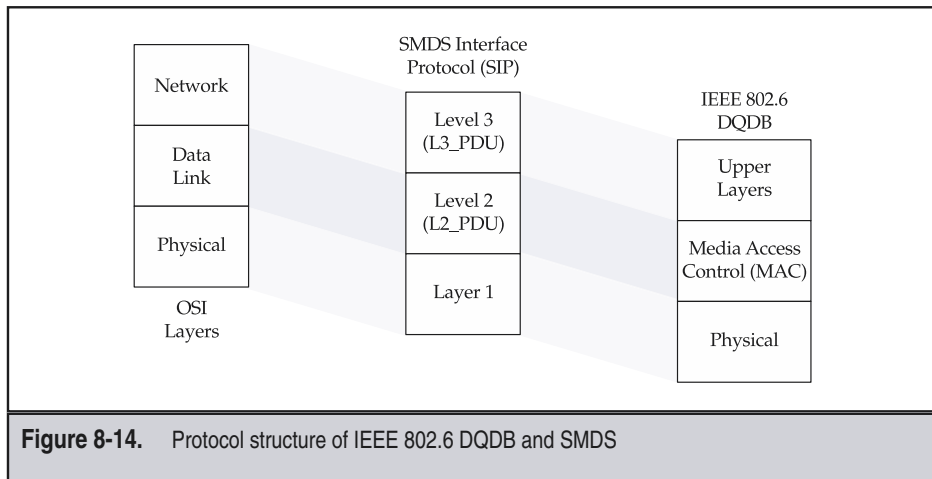
SMDS utilizes one of two forms of cell switching defined in the IEEE 802.6 (DQDB) or ATM AAL3/4, as described in Chapter 12. Central-office switch vendors first introduced SMDS using the DQDB architecture. Versions of SMDS service have been offered by XCs, LECs, and PTTs worldwide. However, many customers have migrated from SMDS to other services.

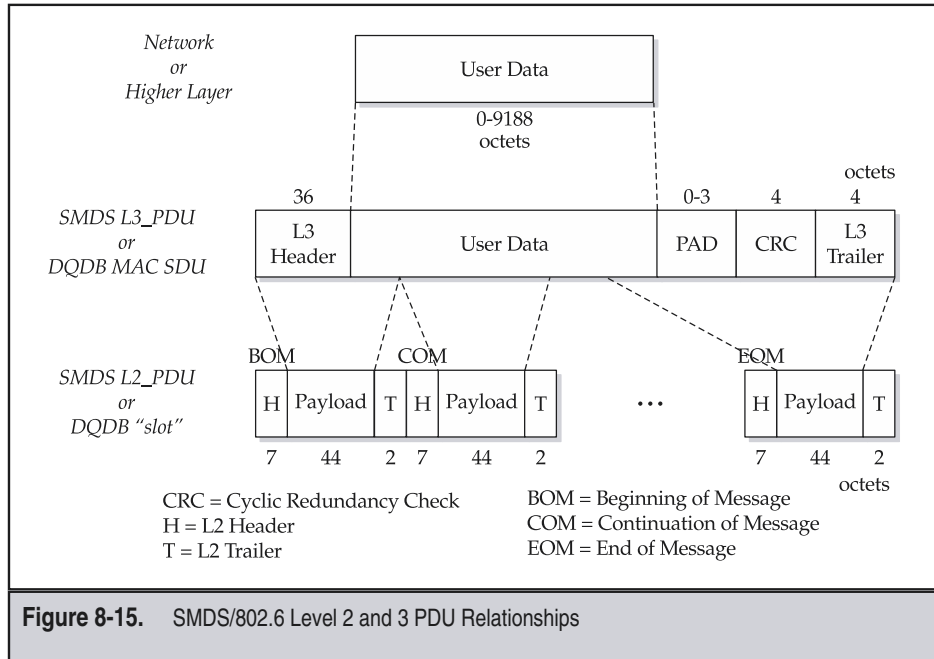
SMDS/IEEE 802.6 Protocol Structure

SMDS and the IEEE 802.6 DQDB protocol have a one-to-one mapping to each other, as illustrated in Figure 8-14. The SMDS Interface Protocol (SIP) has protocol data units (PDUs) at levels 2 and 3. The level 2 SIP PDU corresponds to the DQDB MAC PDU of the IEEE 802.6 standard. The level 3 SIP PDU is treated as the upper layers in IEEE 802.6. There is also a strong correspondence between these levels and the OSI reference model, as shown in the left-hand side of the figure.

SMDS/802.6 Protocol Data Unit (PDU) Formats

Figure 8-15 illustrates the relationship between the user data, the level 3 SMDS PDU, and the level 2 SMDS PDU. The user data field may be up to 9188 octets in length. The level 3 protocol adds header and trailer fields, padding the overall length to be on a 4-octet boundary. Level 2 performs a segmentation and reassembly function, transporting the level 3 payload in 44-octet segments. The level 2 PDU has a 7-octet header and a 2-octet trailer resulting in a 53-octet slot length, the same length as an ATM cell. The level 2





header identifies each slot as being either the Beginning, Continuation, or End of Message (BOM, COM, or EOM). The cells are transmitted headers first.

Figure 8-16 illustrates the details of the SMDS level 3 PDU (L3_PDU) format. The first two octets and last two octets of the SMDS L3_PDU are identical to ATM's AAL3/4 Common Part Convergence Sublayer (CPCS) described in Chapter 12. The SMDS L3_PDU header contains the SMDS Source and Destination Addresses (SA and DA) and a number of other fields. Most of these other fields are included for alignment with the IEEE 802.6 protocol and are not actually used in the SMDS service. When the SMDS level 3 PDU is segmented by level 2, all information needed to switch the cell is carried in either an SSM or BOM slot. This design means that an SMDS switch need only examine the first slot to make a switching decision.

The addressing plan for SMDS and the Connectionless Broadband Data Service (CBDS) employs ITU-T Recommendation E.164, which in the United States is similar to the North American Numbering Plan (NANP) used for telephone service. As the SMDS E.164 address is globally unique, SMDS provides the capability for ubiquitous connectivity. The IEEE 802.6 standard also allows the option for 48-bit IEEE media access control (MAC) addresses to be employed in the DA and SA fields.

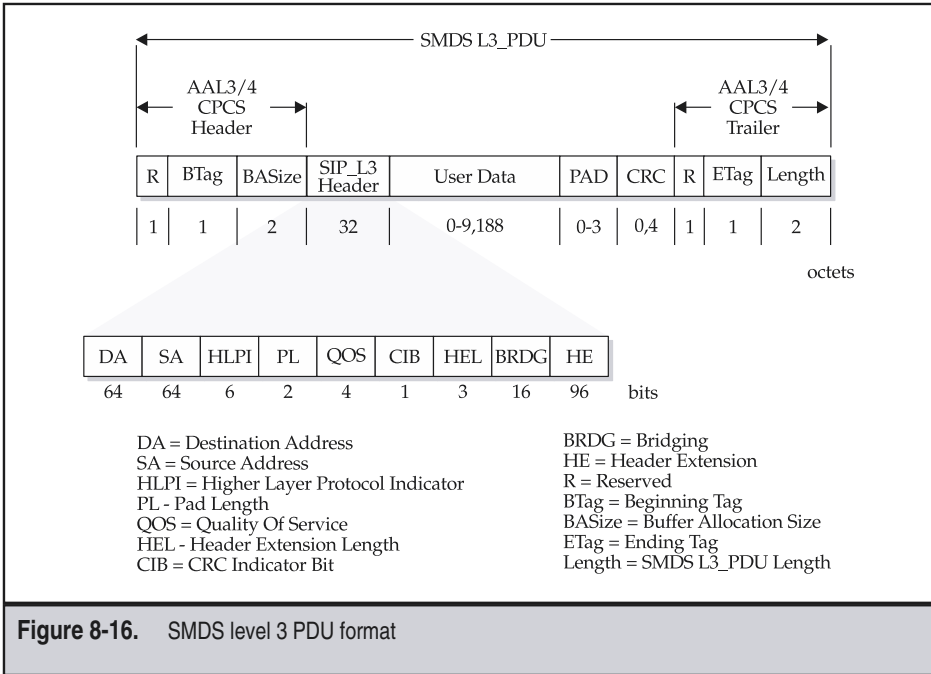
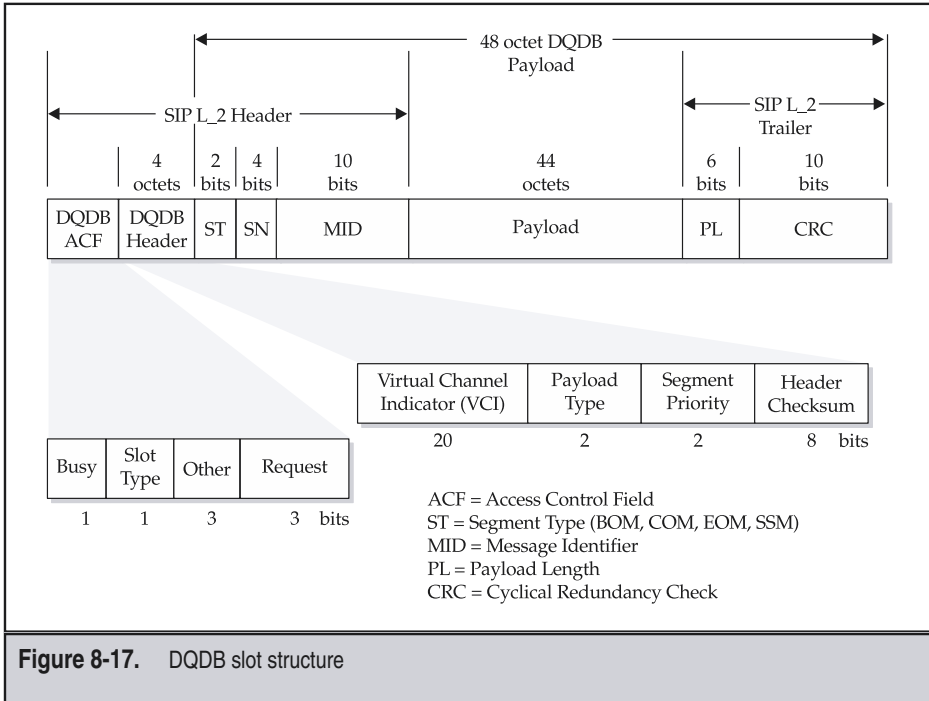


Figure 8-16. SMDS level 3 PDU format

Figure 8-17 illustrates the 48-octet SMDS level 2 PDU format encapsulated in a 53-octet DQDB slot. The other four octets of SMDS level 2 overhead in the DQDB payload are used for the SMDS Segment Type (ST), Message Identifier (MID), Payload Length, and a Cyclical Redundancy Check (CRC) on the 44-octet payload. The SMDS level 2 overhead and function are identical to the ATM AAL3/4 SAR described in Chapter 12. The ST field identifies a Single Segment Message (SSM), a Beginning of Message (BOM), Continuation of Message (COM), or End of Message (EOM) slot. The MID field associates the BOM with any subsequent COM and EOM segments that make up an SMDS L3_PDU. When an SMDS switch receives an SSM or BOM segment, the destination address determines the outgoing link on which the slots are transmitted.

The DQDB Access Control Field (ACF) and header provide a distributed queue for multiple stations on a bus, provide self-healing of the physical network, provide isochronous support, and control management functions. Note that the DQDB ACF and header taken together are exactly five bytes, exactly the same size as the ATM cell header. This choice was made intentionally to make the design of a device that converted between DQDB slots and ATM cells simpler.



DQDB and SMDS Operation

This section describes the distributed queuing and self-healing ring properties of the IEEE 802.6 DQDB protocol with reference to Figure 8-18. Two unidirectional buses A and B, running clockwise and counterclockwise, respectively, interconnect a number of nodes, often configured in a physical ring. Even though the physical configuration may be a ring, logical operation is still bus oriented. Nodes read from both buses, usually passing along any data to the next node in the bus. Any node may assume the role of Head of Bus (HOB) or End of Bus (EOB) according to the rules of the DQDB protocol. The HOB generates 53-octet slots in a framing structure to which the other nodes synchronize. The EOB node simply terminates the bus. Nodes are designed such that if one fails or is powered down, it will continue to passively pass data. Therefore, each node effectively has four ports, two for each bus. Normally, one node would be the HOB for both buses, as shown for node C in Figure 8-18. However, in the event of a failure of one of the buses connecting a pair of nodes, the DQDB protocol ensures that the nodes on either side of the break become the new HOB within a short period of time. This allows network designers to plan for link failures and ensures that a single link failure will not affect the entire network.

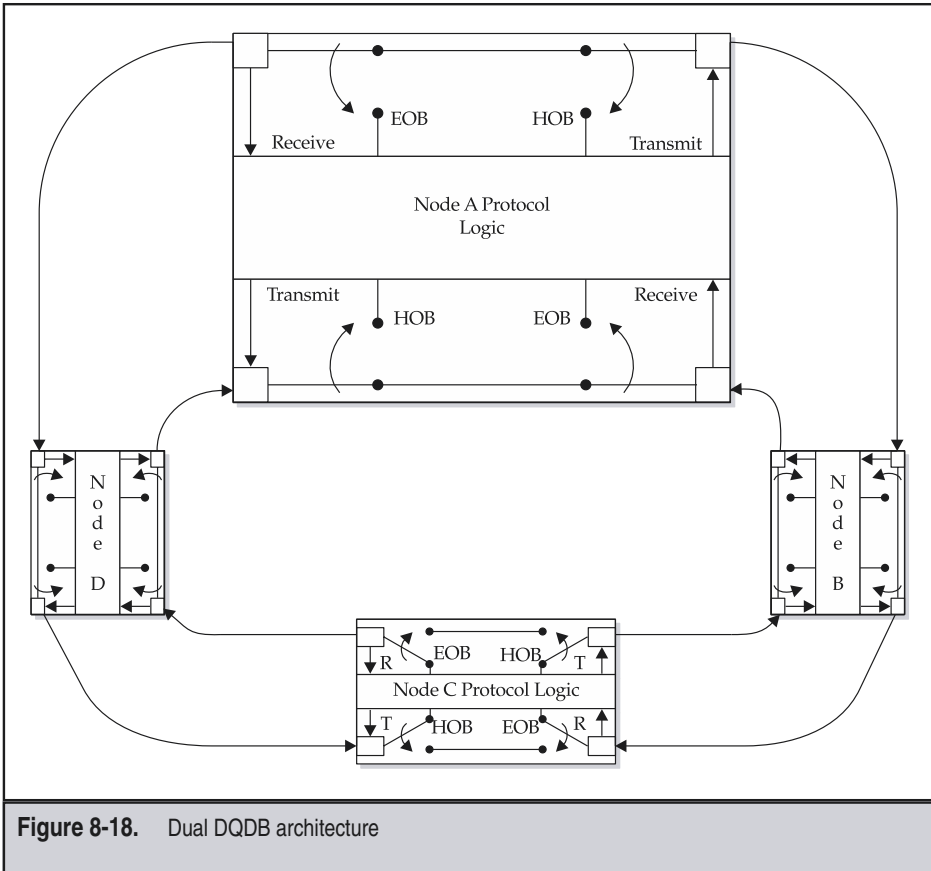


Figure 8-18. Dual DQDB architecture

The Busy and Request bits in the DQDB Access Control Field (ACF) implement a distributed queue. Each node has two counters: one for requests and the other as a countdown for transmission. If a slot passes through the node with the request bit set, then the node increments the request counter; otherwise, the node decrements the request counter. Thus, the request counter reflects how many upstream nodes have slots to send. When a node has data to send, it writes a logical 1 to the request bit in the first slot received that has the request bit equal to 0 and loads the countdown register with the value of the request counter. The countdown timer is decremented each time a slot passes by in the opposite direction. Therefore, when the countdown counter reaches 0, the slot can be sent because all of the upstream nodes have already sent the slots that were reserved in the opposite direction.

This elegantly simple protocol, however, has several problems that complicate the IEEE 802.6 standard. First, the nodes that are closer to the head end of the bus have first access to the request bits, and they can dominate the traffic on the bus, effectively drowning out users at the end of the bus. Second, provisions must be made for stations to join and leave the bus and handle bit errors. The IEEE 802.6 standard defines procedures to handle all of these cases.

Example of SMDS over DQDB Operation

Figure 8-19 illustrates an example of SMDS over DQDB operation. Three DQDB buses, configured as physical rings, are interconnected as shown. A series of slots from a node on the far left are generated with the Destination Address (DA) of a node on the far right. The first slot carries the address for the subsequent slots with the same MID. The nodes use the DQDB protocol to queue and transmit the slots corresponding to this datagram. The result is that a reassembled datagram is delivered to the destination. In general, the MID values differ on each DQDB segment. Once a DQDB segment completes transmission of an L3_PDU, the MID numbers can be recycled as indicated in the figure.

Traffic and Congestion Control Aspects of DQDB and SMDS

There are two aspects of traffic control in SMDS, operating at the DQDB and SMDS service levels. We summarize the important attributes of each aspect in this section.

DQDB Access Control

Prearbitrated access enables the service provider to allocate bandwidth to isochronous connections with a requirement for constrained jitter. This allows priority bandwidth to

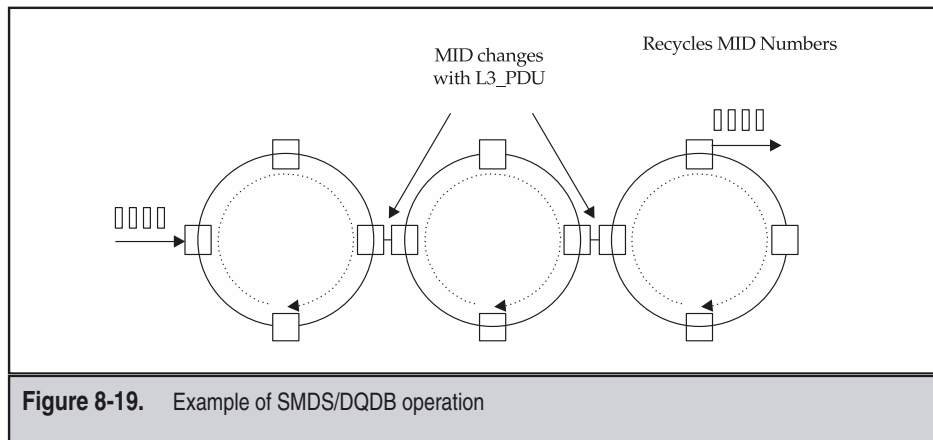


Figure 8-19. Example of SMDS/DQDB operation

be provided to the services that are most affected by variations in delay—namely, video and voice. Video and voice are prime examples of data requiring prearbitrated access to bandwidth because of their intolerance of variation in delay.

Queued arbitrated access is the other access control method. Queued arbitrated access has distance limitations. The longer the distance between the stations, the less effective the priority queuing. This is usually allocated to lower-priority services, or services that can tolerate longer delays and retransmissions more readily than video or voice. This service also assumes that bandwidth balancing is disabled. Bandwidth balancing is discussed next.

Unfortunately, queuing fairness becomes an issue at utilization rates higher than 80 percent, such as during periods of high network activity on DQDB networks. Typically, users located at the end of the bus receive poorer performance than those at the head of the bus because the transmit tokens are taken before the reservation slot arrives. The standards groups invented bandwidth balancing to give users an “equal share” of the available bandwidth by equally dividing bus bandwidth among all users on the bus.

The network designer should strive to design the subnetworks and CPE that interface with the DQDB bus in a manner to ensure that peak traffic conditions do not cause excessive delay and loss of traffic, especially to end-of-bus users. This can be accomplished by the effective use of these techniques.

SMDS Sustained Information Rate

SMDS also defined an open-loop flow control mechanism called Sustained Information Rate (SIR), based on the aggregate of all data originating on the SMDS access line regardless of its destination. Thus, there are no levels of bandwidth management granularity as found in frame relay, ATM, and MPLS. SIR uses a credit manager rate enforcement method where no more than M out of N cells may contain nonidle slots or cells so that $SIR = M * 34 / N$ Mbps. The value of M controls the number of consecutive slots/cells that can be sent at the line rate. Data arriving at a rate higher than the SIR rate is discarded at the originating SMDS switch.

Service Aspects of SMDS

Switched Multimegabit Data Service (SMDS) is a combination packet- and cell-based public data service that supports DS1, E1, E3, and DS3 access speeds through a DQDB Subscriber to Network Interface (SNI). SMDS is also supported through a Data Exchange Interface (DXI) at speeds ranging from 56 Kbps up to and including 45 Mbps. SMDS is a connectionless switched data service with many of the characteristics of local area networks (LANs). SMDS provides the subscriber with the capability to connect diverse LAN protocols and leased lines into a true switched public network solution.

SMDS offers either a point-to-point datagram delivery service or a point-to-multipoint service, defined by a group multicast address. SMDS service operates on both the E.164 source and destination addresses. At the source access line, the SMDS network screens the source address to authenticate the source address's subscription. At the source access line, the SMDS network also screens on the destination address to limit the

destinations reachable from a particular access line. This functionality is comparable to access control lists, or filters, found in firewalls and some routers.

In the early 1990s, experts billed SMDS as the gap filler between frame relay and ATM; however, SMDS is primarily only of historical interest. Central office switch vendors were primary players for initial SMDS cell switches. Public SMDS services based on the IEEE 802.6 standard architecture were slow to emerge after they were rolled out in the United States and many European countries. Lack of effective marketing, the requirement to purchase special CSU/DSUs, the tremendous success of frame relay, and early ATM standardization probably had a lot to do with the low penetration rate of SMDS services. Internationally, a close relative of SMDS is the Connectionless Broadband Data Service (CBDS) as defined by the European Telecommunications Standards Institute (ETSI) standards.

SMDS was probably ahead of its time: although IP supports multicast, that service has also been slow to catch on. Over the past few years, IP is clearly the network protocol of choice for the desktop, and, therefore, SMDS had little chance.

REVIEW

This chapter covered connectionless data services, IP and SMDS, in terms of the origins, protocol formats, functions, operation, traffic aspects, and commercial service aspects. First, we summarized the alternative to the OSI protocol stack embraced by most of the networking world—namely, the de facto standards of the Internet Protocol (IP) suite. The text detailed the concepts of resequencing and handling unreliable underlying networks, central to TCP/IP, using descriptions and examples. We introduced the recently defined method for IP-based networks to support multiple Quality of Service classes and guarantee bandwidth through the Resource Reservation Protocol (RSVP). The text then described the modern notion of a dynamically adjusted window flow control protocol implemented widely in TCP. Finally, the chapter summarized the novel concepts of a distributed-queue, self-healing network in the treatment of SMDS and DQDB. We also looked at some of the origins of ATM's cell structure in DQDB's slotted protocol. This chapter provides the reader with a basic background of the connectionless data communication services and concepts. These concepts reappear within ATM and MPLS and related protocols in the remainder of the book.

CHAPTER 9

LANs, Bridging, and Routing

Most major enterprises embraced local area networks (LANs) in the 1980s, and now even some residences have LANs. Network designers then invented bridging to interconnect multiple LANs to provide greater connectivity. Meanwhile, incompatible LAN standards created the need for routers in the environment of diverse interconnected LANs. Since LANs, bridges, and routers utilize some unique concepts, we begin by introducing the terminology related to bridging, routing, and internetworking used in the remainder of this book. Next, we review LAN protocol standards and bridging in some depth as an introduction to Part 4 regarding the ATM Forum's LAN Emulation (LANE), MPLS tunneling support for Ethernet pseudowires, and VPNs. Finally, we cover the subjects of address resolution and routing as background for the discussion of ATM's routing protocol, the Private Network-Network Interface (PNNI) in Chapter 15, as well as ATM and MPLS protocol support for the Internet Protocol as described in Chapters 14 and 19.

BRIDGING, ROUTING, AND INTERNETWORKING

This section introduces the basic terminology of local area networks and how bridges connect LANs. We also introduce the closely related subjects of routing and internetworking. The section concludes with a discussion of some of the key issues regarding address assignment, address resolution, route selection, and scalability.

Basic Terminology

Let's first review some basic LAN and internetworking terminology of bridging and routing with reference to Figure 9-1, based upon RFC 1932, which defines the framework for the IETF's work on IP over ATM.

A *host* (also called an *end system*) delivers and receives IP packets to and from other hosts. A host does not relay packets. Examples of hosts are workstations, personal computers, and servers.

A *router* (also called an *intermediate system*) also delivers and receives IP packets, but it additionally relays IP packets between end and intermediate systems. All members of an *IP subnet* can directly transmit packets to each other. There may be repeaters, hubs, bridges, or switches between the physical interfaces of IP subnet members. Ethernet or Token Ring LANs are examples of an IP subnet. However, multiple Ethernets bridged together may also be a subnet. The assignment of IP addresses and subnet masks determines the specific subnet boundaries described in more detail in this chapter.

Bridging makes two or more physically disjoint media appear as a single bridged IP subnet. Bridging implementations occur at the media access control (MAC) level or via a proxy address resolution protocol (ARP).

A *broadcast subnet* allows any system to transmit the same packet to all other systems in the subnet. An Ethernet LAN is an example of a broadcast subnet.

A *multicast-capable subnet* provides a facility that enables a system to send packets to a subset of the subnet members. For example, a full mesh of ATM point-to-multipoint connections provides this capability.

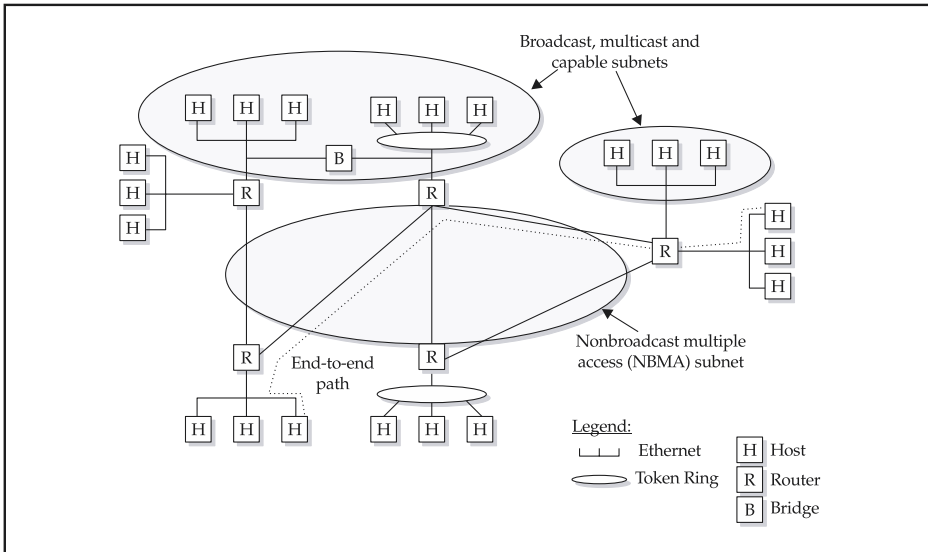


Figure 9-1. Basic LAN and internetworking terminology

A *nonbroadcast multiple access (NBMA) subnet* does not support a convenient multidestination connectionless delivery capability as broadcast and multicast-capable subnetworks do. A set of point-to-point ATM or MPLS connections is an example of an NBMA subnet.

An *internetwork* is a concatenation of networks, often employing different media and lower-level encapsulations, that form an integrated larger network supporting communication between hosts. Figure 9-1 illustrates a relatively small internetwork when judged in comparison with *the Internet*, which comprises tens of thousands of networks.

The term *network* may refer to an Ethernet cable or to a collection of many devices internetworked across a large geographic area sharing a coordinated addressing scheme.

Vendors of intermediate and end systems strive to implement an efficient process for deciding what to do with a received packet. Possible decisions are local delivery, or forwarding the packet on to another external interface. *IP forwarding* is the process of deciding what to do with a received IP packet. IP forwarding may also require replacement or modification of the media layer encapsulation when transitioning between different LAN media.

IP routing involves the exchange of topology information that enables systems to make IP forwarding decisions that cause packets to advance along an end-to-end path toward a destination. Sometimes routing is also called a *topology distribution protocol*. We cover this important topic later in this chapter.

An *end-to-end path* is an arbitrary number of routers and subnets over which two hosts communicate—for example, the path illustrated by the dotted line in Figure 9-1.

Routers implement this path by the process of IP forwarding at each node as determined by an IP routing algorithm.

An *IP address resolution protocol (ARP)* provides a quasi-static mapping between an IP address and the media address on the local subnet.

Scalability refers to the ability of routing and address resolution protocols to support a large number of subnets, as well as handle the dynamics of a large internetwork. In networks with large numbers of interconnected subnets, routers must exchange large amounts of topology data and store the resultant forwarding information in high-speed memory. Furthermore, as network size increases, so does the likelihood that some network elements are changing state, thus creating the need to update routing topology information. Hence, scalability impacts the required processing power and storage for the routing protocols within routers.

Address Assignment and Resolution

A key requirement in any communications network is that unique addresses be assigned to each of the entities that want to communicate. This is the case in the telephone network, where every phone in the world has a unique number. It is also found in the 48-bit IEEE 802.3 Media Access Control (MAC) assignments built into every Ethernet interface. Administrators must assure that every user or “host” in the Internet has a unique IP address.

Assuring that the address assignments are unique and that they efficiently administer an address space presents some challenges. Not only must addresses be handed out, but a means for users to return addresses and request additional blocks of addresses is also required. Furthermore, if there is more than one administrative authority, then the scope of assignments allocated to each administration must be clearly defined. It is sometimes difficult to predict the demand for addresses. For example, telephone companies periodically realign area codes because the demand differs from the forecast of just a few years ago. If an administrative authority hands out blocks of addresses too freely, then the network can run out of unique addresses well before the limit determined by the number of bits in the addresses, and this has occurred with many of the IP address blocks already assigned.

Once you have your own address and the address of someone that you wish to communicate with, how do you resolve the address of the desired destination into information about how to get it there? First, consider the following simple analogy: Let’s say that you have spoken to an individual on the telephone for the first time and have agreed to meet him or her at a party to which you both have been invited by the same host. Once you arrive at the party, you can find the individual (resolve the address) in one of two ways: you can jump up on stage, grab the microphone, and broadcast your presence—or you can locate the host and ask to be introduced to the individual. Broadcast (i.e., the microphone) is commonly used in shared-medium LANs to resolve addresses. A problem arises, however, when the volume of broadcast traffic approaches the level of user traffic, like when it’s difficult to hear at a crowded party. The analogy to having someone who has the information (the party’s host) resolve the address (match the name with the individual) is like that of an address resolution protocol (ARP) server used in some designs to

support LAN and internetworking protocols. The analogy of your jumping up on stage and announcing your intentions is the broadcast ARP protocol commonly used on LANs described at the end of this chapter.

Routing, Restoration, and Reconfiguration

After resolving the destination address, there is the issue of what is the best way to reach that address through a network of nodes. A commonly used solution to this problem assigns each link in a network a *cost*, and then employs a routing algorithm to find the least-cost route. This cost may be economic, or it may reflect some other information about the link, such as the delay or throughput. One analogy is as follows: You are traveling from your house to the grocery store. One route takes more gas yet takes less time, and the other route takes less gas yet takes more time (e.g., because of construction delays). Each route has a different set of costs (time and money). You choose the cost measure most important to you—that is, do you want to minimize time spent or gas used—and you take that route choice. Routing protocols employ similar decision criteria and cost metrics.

Routing algorithms exchange information about the topology—that is, the links that are connected and their associated costs—in one of two generic methods [Perlman 92, Black 92]. The first is a *distance vector* algorithm where neighbor nodes periodically exchange vectors of the distance to every destination subnetwork. This process eventually converges on the optimal solution. The second is where each router learns the entire *link-state* topology of the entire network. Currently, this is done by flooding only the changes to the link-state topology through the network. Flooding involves copying the message from one node to other nodes in the network in a tree-like fashion such that at least one copy of the message is reliably received by every node. Therefore, an important trade-off in a flooding protocol is minimizing transmission of unnecessary copies, yet ensuring reliable and rapid delivery of one copy. The link-state approach is more complex but converges much more rapidly than the distance vector approach. *Convergence* is the rate at which the knowledge of network topology at every node goes from an unstable state to a stable state. When the topology of the network changes due to a link or node failure, or the addition of a new node or link, or because of a link metric change, the flooding protocol must disseminate this information to every other node. The *convergence time* is the interval required to update all nodes in the network about the topology change.

The distance vector method was used in the initial data communication networks such as the ARPAnet and is used by the Internet's Routing Information Protocol (RIP). A key advantage of the distance vector protocol is its simplicity. A key disadvantage of the distance vector protocol is that the topology information message grows larger with the network, and the time for it to propagate through the network increases as the network grows. Another disadvantage of the distance vector algorithm is that it uses hop count instead of a weighted link metric. Minimum-hop routing leads to some pathological route choices in certain network topologies. Convergence times in the order of minutes are common in distance vector algorithm implementations.

The link-state advertisement method is a more recent development designed to address the scalability issues of the distance vector technique. A fundamental tenet of the approach is to reliably flood an advertisement throughout the network whenever the state of a link changes. Examples of link-state change are adding a new link, deleting a link, and an unexpected link failure. Thus, each node obtains complete knowledge of the network topology in the convergence time t (usually several seconds). After any change, each node computes the least-cost routes to every destination using an algorithm such as the Dijkstra algorithm [Perlman 92, McDysan 02]. Since every node has the same topology database, they all compute consistent next-hop forwarding tables. Examples of link-state routing protocols are the Internet's Open Shortest Path First (OSPF); the Private Network-Network Interface (PNNI) used in ATM networks; and the OSI IS-IS Routing Protocol, where IS stands for Intermediate System. Key advantages of link-state protocols when compared with distance vector approaches are a reduction in topology update information that must be propagated and decreased convergence times. A key disadvantage is the increased complexity of these methods and consequently increased difficulty in achieving interoperability between different vendor implementations.

Another important class of routing protocols is the path vector approach, the foremost example being the Border Gateway Protocol (BGP) that binds the various ISPs that make up the Internet together. BGP works by establishing a TCP session between routers, which then establish a BGP session whose status is checked using keep-alive messages when no other traffic flows. BGP update messages contain a vector of autonomous system (AS) numbers regarding the best path to reach a particular IP address prefix, along with a number of optional attributes. A router can apply a set of policies to the set of path vectors learned from its BGP neighbors and then typically picks the path with the shortest number of AS hops.

Routing is a complicated subject, and the preceding descriptions serve as only a brief introduction sufficient for our purposes. This chapter provides more detail after first covering LAN standards and bridging protocols. Readers interested in even more detail should consult more detailed descriptions, such as [Perlman 92] for OSI-based routing, and References Comer 91, Tann 96, Streenstrup 95, Moy 98, and Huit 95 for IP-based routing. PNNI and traffic engineering extensions to IGP's will be discussed in more detail when we describe ATM and MPLS routing and signaling protocols in Part 3.

IEEE LOCAL AREA NETWORKING (LAN) STANDARDS

This section summarizes the most popular LAN standards in terms of protocol layering concepts with example deployment scenarios, and then reviews of each of the major LAN protocols. The coverage begins with the Logical Link Control (LLC) and then moves on through the Media Access Control (MAC) specifications for Ethernet, Token Ring, Fast Ethernet, gigabit Ethernet, and FDDI.

Layered LAN Protocol Model

LAN protocols implement the protocol stack shown in Figure 9-2. The Logical Link Control (LLC) and Media Access Control (MAC) sublayers of the IEEE 802.X standards map to the data link layer, while the actual physical medium (e.g., twisted pair) that interconnects stations on a LAN maps to the physical layer. The MAC layer manages communications across the physical medium, defines frame assembly and disassembly, and performs error detection and addressing functions. Figure 9-2 shows how the LLC layer interfaces with the network layer protocols through one or more service access points (SAPs). These SAPs provide a means for multiplexing within a single host over a single MAC layer address, as illustrated in the figure. As we shall see later, address fields within the LLC portion of the data link frame specify the source and destination SAPs.

Typical LLC and MAC Sublayer Implementations

The IEEE 802.2 standard defines the logical link control layer, while the IEEE 802.3 through 802.12 standards define various aspects of MAC layer protocols. Figure 9-3 shows some examples of the physical relationship between LLC and MAC interface points in devices connected to the two Ethernet networks and the Token Ring network in the center of the figure. Starting in the upper left-hand corner, a host runs multiple applications, each with a separate LLC, and interfaces to an Ethernet LAN via a single MAC address. In the upper-middle portion of the figure, a bridge forwards MAC frames according to information obtained by a bridging protocol between the Ethernets and the Token Ring. In the upper right-hand corner of the figure, a LAN hub supports a terminal

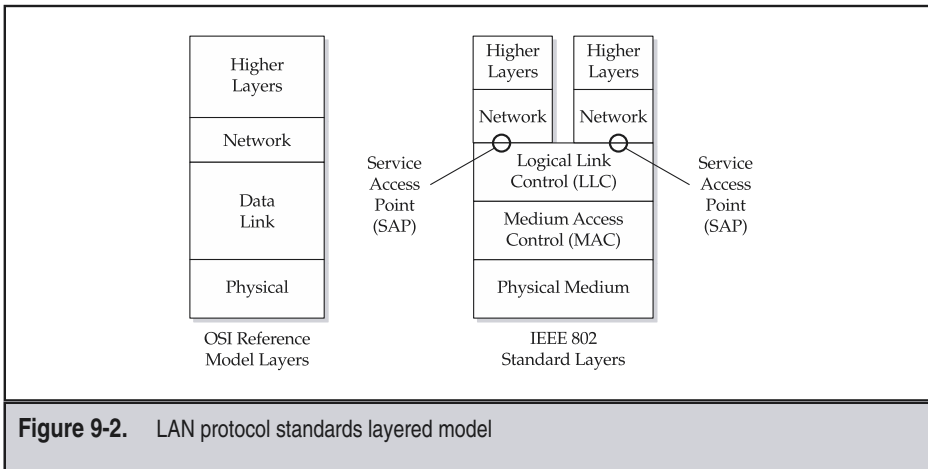


Figure 9-2. LAN protocol standards layered model

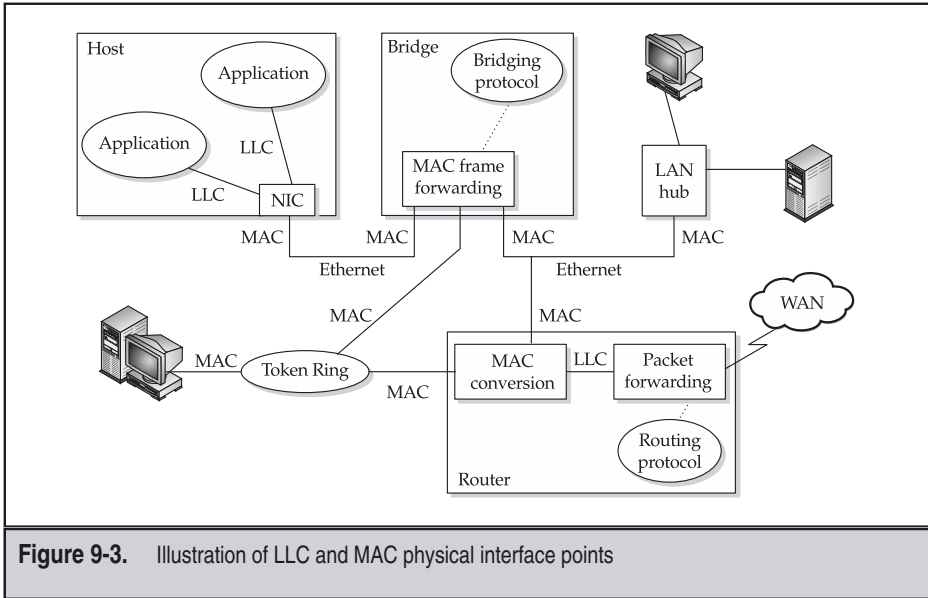


Figure 9-3. Illustration of LLC and MAC physical interface points

and workstation. In the lower right-hand corner, a router interconnects an Ethernet, the Token Ring, and a wide area network. The router has multiple MAC addresses but forwards packets according to information in the LLC portion of the packet header. Routers also run a routing protocol over the network layer connecting LLC layer entities. Routers forward packets according to the network layer header contained inside the LLC PDU. In the lower left-hand corner of the figure, a workstation interfaces via the Token Ring MAC interface. Now that we've seen the basic layering concepts and have some context in terms of example implementations, let's take a look at some further details of the LLC and MAC sublayers.

The Logical Link Control (LLC) Sublayer

The IEEE Standard 802.2 defines the LLC protocol that hides the differences between various MAC sublayer implementations from the network layer protocol. This allows systems on very different types of LANs—for example, Token Ring and Ethernet—to communicate. LLC provides services to the network layer that are either connection-oriented or connectionless. Connection-oriented service uses peer-to-peer communications and provides acknowledgments, flow control, and error recovery. There are three classes of services provided in the LLC: unreliable datagram, acknowledged datagram service, and connection-oriented service.

When the LLC layer receives user data, it places it in the information field and adds a header to form an LLC protocol data unit (PDU), as shown in Figure 9-4. Destination and Source Service Access Point (DSAP and SSAP) address fields along with a control field precede a variable-length information field in the LLC PDU. The 802.2 standard defines the least significant bit of the DSAP address to identify a group address. The least significant bit of the SSAP address field is a Command/Response (C/R) bit for use by LLC services. Implementations may use only the high-order 6 bits of the DSAP and SSAP fields, limiting the total number of SAPs to 64.

The 802.2 standard defines SAP values for a number of ISO protocols, but none for some other important protocols such as IP. Since many potential combinations of LAN types and protocols exist, the IEEE extended the LLC control header using a subnetwork access point (SNAP) structure. Normally, the LLC DSAP and SSAP are one byte, and the LLC controls one or two bytes. When both the DSAP and SSAP values equal x'AA', and the LLC control field equals x'03', then the extended LLC header adds two fields—a three-byte Organizationally Unique Identifier (OUI) for defining an organization that assigns a two-byte Protocol Identifier (PID). One example would be an Ethernet frame carrying an IP datagram, where the OUI and PID values would be x'000000' and x'0800', respectively. Figure 9-5 illustrates the subnetwork access point (SNAP) structure. RFC 1340 lists the assigned LLC1/SNAP header types. As described in Chapter 18, the multiprotocol encapsulation over ATM employs this extended LLC/SNAP header.

The Media Access Control (MAC) Sublayer

The Media Access Control (MAC) sublayer manages and controls communications across the physical medium, assembles and disassembles frames, and performs error detection and addressing functions. Table 9-1 summarizes some key attributes of the IEEE 802 series of standards for MAC sublayer protocols.

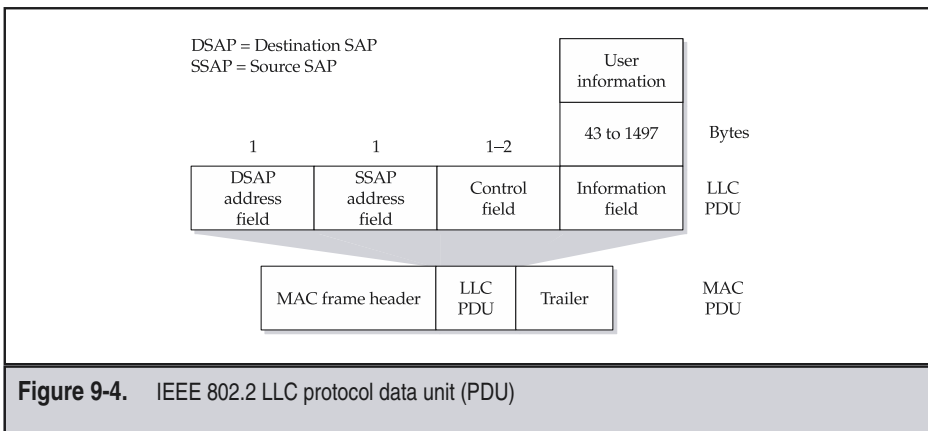


Figure 9-4. IEEE 802.2 LLC protocol data unit (PDU)

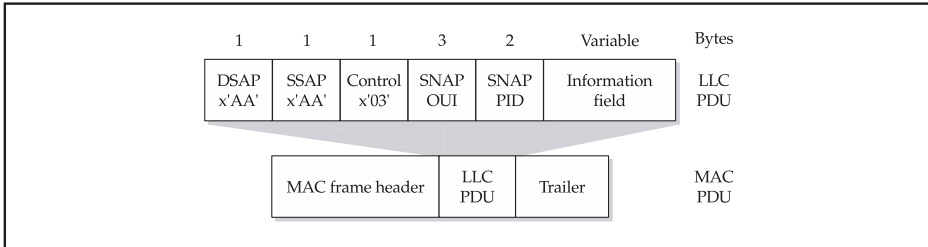


Figure 9-5. IEEE extended LLC/SNAP header

The MAC layer adds a header and a trailer to the LLC PDU prior to transmission across the physical medium. Each MAC layer PDU contains a unique 6-octet (48-bit) MAC address for the source and destination stations. The IEEE assigns 24-bit identifiers called *Organizationally Unique Identifiers (OUIs)* to each enterprise manufacturing Ethernet interfaces. The manufacturer uniquely assigns the remaining 24 bits, resulting in a unique 48-bit address known as the physical address, hardware address, or MAC address. Thus, a unique address identifies every LAN network interface card (NIC). Each

IEEE Standard	Standard Title	MAC Speed (Mbps)	Physical Media Supported	Maximum Payload Size (Bytes)
802.3	Ethernet, Fast Ethernet, Gigabit, 10 Gbps Ethernet	1, 10, 100, 1000, 10,000	Coax, 2W UTP, STP, fiber	1500
802.4	Token Bus	1, 5, 10	Coax	8191
802.5	Token Ring	1, 4, 16	STP	5000 ¹
802.6	Distributed Queue Dual Bus (DQDB)	34, 44, 155	Coax, fiber	8192
802.9a	IsoEthernet	16	2W UTP, STP	
802.12	100VG-AnyLAN	100	4W UTP, STP, fiber	

Notes: 2W UTP is 2-wire unshielded twisted pair.
4W UTP is 4-wire unshielded twisted pair.
STP is shielded twisted pair.

¹ Computed for 4 Mbps media speed and a token-holding time of 10 ms.

Table 9-1. Important Attributes of MAC Sublayer Standards

station on a LAN examines the destination MAC address in a received frame to determine whether the frame should be passed up to the LLC entity addressed by the DSAP field in the LLC PDU. We now examine the most common MAC protocols: Ethernet, Token Ring, Fast Ethernet, gigabit Ethernet, and FDDI.

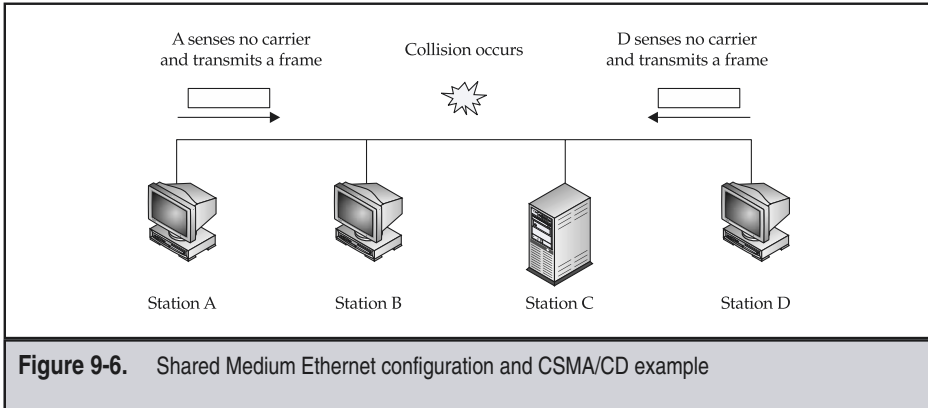
Ethernet and the CSMA/CD 802.3 MAC Sublayer

Dr. Robert M. Metcalfe invented Ethernet in the late 1970s at the Xerox Palo Alto Research Center. It was based upon some concepts used in the ALOHA system deployed to implement radio communication in the Hawaiian islands [Tannenbaum 96]. Following successful trials of the Xerox Ethernet, a multivendor consortium of DEC, Intel, and Xerox (abbreviated DIX) published the first Ethernet specifications in 1980, with the first Ethernet products appearing shortly thereafter in 1981. The original Ethernet specification covered operation at 10 Mbps over a 50 ohm coaxial cable, also called Thick Ethernet. Indeed, the term “Ether” refers to the broadcast medium of an electrically terminated coaxial cable with “vampire” taps connecting each station to the cable. In a classical case of vendor-driven standards development, this proprietary standard provided the basis for the IEEE 802.3 committee. The IEEE 802.3 standard, adopted by the ISO, IEC, and ANSI [IEEE802.3], defined a Media Access Control (MAC) protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD) at several speeds over a variety of different media, as indicated in Table 9-1.

An analogy helpful in understanding CSMA/CD is a Citizen’s Band (CB) radio channel. Users first listen to see if anyone is already talking (Carrier Sense) before speaking. Everyone has an equal chance to seize the radio channel (Multiple Access), whether they have anything useful to say or not. Once a user begins speaking, no one interrupts until the transmission completes, indicated by the speaker saying “over.” Two people may begin transmitting simultaneously, in which case everyone gets a garbled signal (Collision Detection). The CBER’s then back off and repeat the process, with the contending users hopefully waiting different amounts of time to transmit again. As long as the channel isn’t too full, collisions don’t occur too frequently and useful communication results. Ethernet CSMA/CD works in a very similar manner to this CB channel example, but with some specific refinements to make the collision resolution process work smoothly. Now let’s take a more in-depth look at the real CSMA/CD.

CSMA/CD is a fancy name for a relatively simple protocol (analogous to the shared CB radio channel discussed previously) that allows stations to transmit and receive data across a multiple access medium, called a “segment” or “collision domain,” shared by two or more stations, as illustrated in Figure 9-6. Note that LAN switches support a single user per segment, decreasing the number of contending users to two: the LAN switch port and the host LAN interface card. We use the notation of a single line with multiple taps depicted in this figure throughout this text to indicate an Ethernet segment.

When a station has data to send, it first senses the channel for a carrier transmitted by any other station. If it detects that the channel is idle, the station transmits an 802.3 frame in serial bit form, as stations A and B do in the example of Figure 9-6. If the transmitting



station receives its own data frame without any errors, as determined by the frame check sequence, then it knows that the frame was delivered successfully. However, due to the finite propagation delay of electromagnetic signals, the transmissions of two stations may still collide at other points on the shared medium if they begin transmitting at nearly the same time as illustrated in the figure. Stations detect such collisions via means such as an increase in received signal power or a garbled signal. In order to ensure that all stations detect the collision event, each station that detects a collision immediately sends out a short burst of data to garble the signal received by other stations. Eventually, the transmitting station also detects the collision and stops transmitting. Following a collision event, the end stations initiate an exponential back-off algorithm to randomize the contending station's next attempt to retransmit the lost frame using contention slots of fixed duration. The design of the back-off algorithm normally results in the resolution of collisions within hundreds of microseconds. The 802.3 CSMA/CD system can achieve throughput approaching 50 percent under heavy load conditions due to these collisions and retransmissions [Tannenbndum 96, MDysdn 98].

Figure 9-7 shows the IEEE 802.3 CSMA/CD MAC PDU frame (a) compared to a DIX Ethernet frame (b). The IEEE uses the convention where the order of octet and bit transmission is from left to right, as shown in the figure. The 802.3 frame differs from the Ethernet frame only in the interpretation of the two bytes after the address fields, unfortunately making the protocols incompatible. The seven-byte preamble field contains the bit pattern '10101010', which allows all stations to synchronize their clocks to the transmitter. The Starting Frame Delimiter (SFD) contains the "start-of-frame" character, '10101011', to identify the start of the frame. The next two six-byte (48-bit) fields identify the MAC-layer destination and source addresses formed from the IEEE-assigned 24-bit OUI and the 24-bit unique address assigned by the manufacturer of the Ethernet interface card. The 802.3 standard defines some unique addresses. An address of all ones is reserved for the broadcast address. The high-order bit of the address field is zero for ordinary addresses or one for group addresses, which allow a set of stations to communicate on a single address.

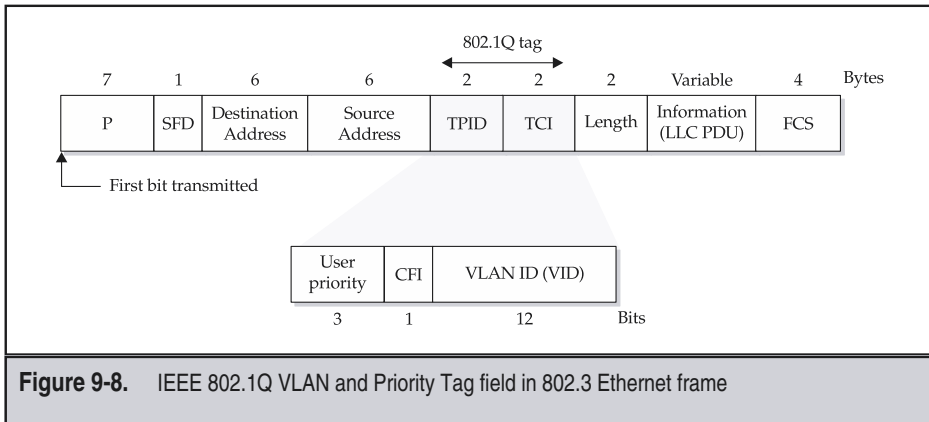
Protocol ID (TPID) and a two-byte Tag Control Information (TCI) field. The TPID allows an Ethernet switch to distinguish between an 802.1Q tag or another type of element, such as an LLC/SNAP header. The TCI field has three components, as shown at the bottom of Figure 9-8. The 3-bit user priority field is defined as described previously. The Canonical Format Indicator (CFI) bit communicates whether a Routing Information Field (RIF) is present. Finally, the 12-bit VLAN ID (VID) field allows support of up to 4095 VLANs on a single physical 802.3 interface. A null VID field means that the frame contains only user priority information.

The 802.1Q VLAN standard is a bridging protocol that allows formation of logically separate groups of LAN-attached stations and broadcast domains across a network of Ethernet switches and bridges. VLAN switches and bridges use a Generic Attribute Registration Protocol (GARP) tailored to VLANs, called GVRP, to communicate information about how station MAC addresses are mapped to individual VLANs. Several types of VLANs are supported. A port-based VLAN involves a VLAN switch tagging frames received from a user LAN interface with a VLAN tag configured for that interface. In this way, sets of ports can be part of distinct VLANs. If a LLC/SNAP VLAN tag is used (not shown in Figure 9-8), it is also possible to configure VLANs according to protocol type—for example, separate VLANs for IP and IPX.

Token Ring

The IBM labs in Zurich, Switzerland, developed the Token Ring protocol in the late 1960s, with the first commercial products appearing in 1986. The IEEE adopted Token Ring as the 802.5 MAC standard, which also works with the IEEE 802.2 logical link control layer.

A token ring is actually a closed loop of unidirectional point-to-point physical links connecting ring interfaces on individual stations, as shown in Figure 9-9. This text uses the notation of a circle to indicate a token ring. As shown in the figure, the ring interfaces operate in one of three modes: transmit, listen/modify, or powered-down. In the listen/modify mode, the ring interface inserts a one-bit delay so that each station can receive, and possibly modify, the bit before transmitting the bit on to the next station, as shown



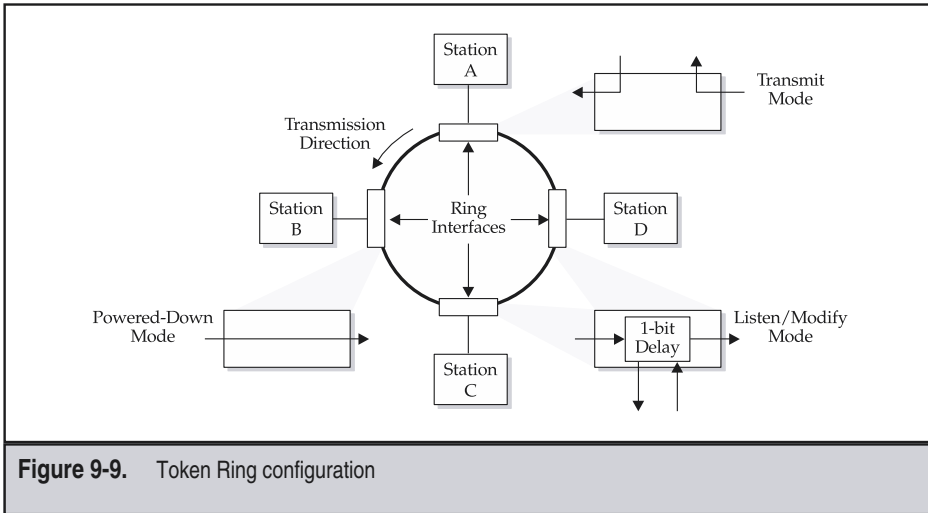
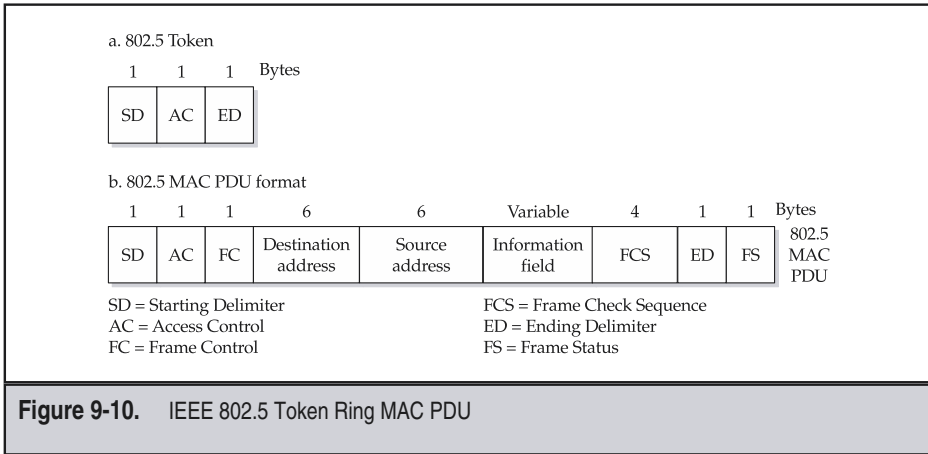


Figure 9-9. Token Ring configuration

for stations C and D in the figure. Since stations actively participate in the transmission of bits, the standard requires that they electrically pass the bit when they are in a powered-down mode, for example, using a relay as illustrated for station B in Figure 9-9. A unique three-byte pattern, called a *token*, circulates around the ring when all stations are idle. Thus, a token ring must have enough delay to contain an entire token, sometimes requiring the insertion of additional delay via one station dynamically designated as the ring monitor.

As the token circulates in the ring, each station can seize the token, temporarily remove it from the ring, and send one or more data frames to the next station on the ring, as shown for station A in Figure 9-9. The data frame then circulates around the ring to every station. The destination station's ring interface copies the frame and forwards it to the next station on the ring. The originating station receives its own data frame and can verify correct reception through the frame check sequence. The originating station then removes the frame from the ring. The station holding the token can transmit data for a maximum token holding time that has a default value of 9.1 ms, which constrains the maximum MAC frame size to approximately 4500 bytes on a 4 Mbps token ring or 14,500 bytes on a 16 Mbps token ring. Once the station completes data transmission (or reaches the holding time limit), it stops transmitting data frames and sends the token to the next station. Hence, Token Ring LANs have greater minimum delay than Ethernets because a station must wait for the token prior to transmitting any data. On the other hand, since only the station holding the token can transmit data, no collisions occur as in CSMA/CD. Therefore, under heavy load conditions, Token Ring protocol throughput approaches maximum efficiency.

Figure 9-10a shows the IEEE 802.5 Token Ring, and Figure 9-10b shows the 802.5 MAC PDU format. The Starting Delimiter (SD) and Ending Delimiter (ED) fields mark



the beginning and end of the token as well as the MAC PDU. The Access Control (AC) field contains a bit that identifies the frame as the token. The AC field also provides for a multiple-level priority scheme according to a complex protocol. The Frame Control (FC) field distinguishes data frames from frames used to control the Token Ring protocol. As in all IEEE MAC protocols, six-byte (48-bit) MAC addresses identify the source and destination stations. The Frame Check Sequence (FCS) field uses a CRC-32 code to detect bit errors. Stations that detect an error through the FCS set a bit in the Ending Delimiter field, which also serves to mark the last frame in a sequence of frames sent by a single station. The destination can acknowledge receipt of a data frame using bits in the Frame Status (FS) byte.

Every Token Ring must also have a monitor station that performs various functions related to ring maintenance. The monitor ensures that there is always a token, and it also detects orphan frames created by a station crashing in the middle of a transmission, which means that the frame is not removed and would circulate indefinitely. Like many real-world Ethernet networks, Token Ring implementations typically are a physical star homed on a hub in a wiring closet. This physical design also allows the hub to automatically repair a broken ring due to failures in the wiring.

100 Mbps Fast Ethernet

Ethernet is the most popular LAN technology in use today. However, as workstation and server technology advanced, 10 Mbps simply wasn't fast enough. Furthermore, many LAN managers attempted to utilize complex designs of bridges and routers to meet the capacity demand. Since FDDI ended up being a complex, and expensive, protocol, the IEEE reconvened the 802.3 committee in 1992 to upgrade the popular 802.3 LAN standard to 100 Mbps. Two competing proposals emerged: a conservative proposal to simply increase the speed of the current 802.3 protocol, and another proposal to rework the entire protocol to give it new features that resulted in the establishment of the 802.12 committee.

As often happens in standards bodies, both proposals won out. The first, covered in this section, resulted in an updated 802.3 specification in 1996 [IEEE802.3] that added specifications for 100 Mbps, commonly called Fast Ethernet or 100 Base-T [Saunders 96], which quickly became the predominant 100 Mbps LAN standard. The second, covered in the next section, resulted in the 802.12 standard called 100VG-AnyLAN.

Basically, Fast Ethernet speeds up the existing CSMA/CD media access control mechanism from 10 Mbps to 100 Mbps. All frame formats, procedures, and protocols remain pretty much the same. This means that applications designed for 10 Mbps Ethernet can run essentially unchanged over 100 Mbps Fast Ethernet. One important change is that the shared media topology was eliminated in favor of exclusive use of a star topology. The designers chose to use twisted pair copper media at distances up to 100 m and fiber optic cables for distances up to 2000 m.

The 802.3 standard update defines three physical media: 100BASE-T4, 100BASE-TX, and 100BASE-FX. The 100BASE-T4 standard uses four unshielded twisted pairs from each device back to a hub, a situation commonly available in telephone-grade office telephone wiring. Avoiding the need to rewire an existing office is an increasingly important practical consideration for network designers. Since the cost of LAN technology continues to decrease with respect to labor costs, the cost of rewiring can easily exceed equipment costs. The standards achieve higher effective throughput over the shielded twisted pair, and especially fiber optic media, since transmission is full duplex. The 100BASE-TX standard uses two data-grade twisted pairs. The 100BASE-FX standard uses a pair of optical fiber cables defined by the same ANSI standard used for FDDI. User equipment (i.e., a DTE) may interface to any of these three physical media through a 40-pin Media Independent Interface (MII), or directly via the 4-pair, 2-pair set of wires or pair of optical fibers.

However, Fast Ethernet still has the variable delay and collision-limited throughput that the old slow Ethernet did. Furthermore, the central hub-based design limits effective network diameter to approximately 200 meters, less than ten percent of the old Ethernet. But at only two to three times the price of 10 Mbps Ethernet, the 100 Mbps Fast Ethernet is a good value for a ten-fold increase in performance for a LAN.

100VG-AnyLAN

The IEEE 802.12 standard [IEEE802.12] defines the competing proposal for high-speed Internet, also called 100VG-AnyLAN in the literature [Saunders 96]. As the name implies, it supports operation with any (existing) LAN protocol—namely, Ethernet or Token Ring—but not both at the same time with new stations supporting the 802.12 standard. A key feature of 100VG-AnyLAN is the demand priority scheme implemented in a hierarchy of LAN repeaters. The level 1, or root, repeater may have several lower levels of repeaters connected via cascade ports. Local ports connect 100VG-AnyLAN end node devices to any of the repeaters.

Like Fast Ethernet, 100VG-AnyLAN avoids any shared media and uses switching hubs exclusively. The demand priority protocol utilizes a round-robin polling scheme where individual stations request priority for each MAC frame from the repeater. The

802.12 standard gives prioritized service, across a hierarchy of LAN repeaters, to frames sensitive to variations in delay over frames not as sensitive to variations in delay, requiring only best-effort service by selectively granting permission to transmit in response to prioritized end station requests. Furthermore, since the access control algorithm is deterministic, no collisions occur and throughput of the LAN can approach 100 Mbps. Thus, 100VG-AnyLAN overcomes the main disadvantages of the traditional CSMA/CD Ethernet protocol: lack of prioritized service and collision-limited throughput.

The 802.12 standard requires that 100VG-AnyLAN hubs support either the traditional 10 Mbps Ethernet frame format or the standard Token Ring frame format (but not both) on the same LAN without bridges. Therefore, the IEEE 100VG-AnyLAN standard provides an easy migration path from existing 10 Mbps Ethernet and 4/16 Mbps Token Ring networks. 100VG-AnyLAN can operate over four-pair UTP (up to 100 m), two-pair STP (up to 200 m), and fiber optic cable (up to 2000 m). However, 100 VG-AnyLAN has seen little commercial adoption, since the marketplace has adopted switched 100 Mbps and Gigabit Ethernet. Here, the lesser expense and higher-speed capacity of Gigabit Ethernet has won out over the higher complexity of the 802.12 standard.

Gigabit and 10 Gbps Ethernet

The objective of Gigabit Ethernet was to be as backward compatible with its other Ethernet ancestors as possible. However, a different physical layer was needed to operate at ten times the speed of Fast Ethernet. The IEEE used the Fiber Channel standard as the basis for this new physical layer standard to accelerate time to market. This approach had several important advantages. By preserving the 802.3 Ethernet frame structure and management framework, Gigabit Ethernet allows seamless upgrades for existing LAN backbones while preserving existing investments in hardware and software. In essence, Gigabit Ethernet is just a faster Ethernet. On the other hand, the new protocol retains some disadvantages from its ancestors. The maximum frame size is still only 1500 bytes, and the overhead is not insignificant when compared with other layer 2 protocols like frame relay or MPLS. Full-duplex operation inserts a 96-bit interframe gap and requires a 64-byte minimum frame size.

Following hot on the heels of the success of Gigabit Ethernet, the IEEE 802.3ae committee has been busily at work since 1999 defining a standard with an expected completion date of mid-2002, with initial products planned shortly thereafter. Like its predecessors, 10 Gbps Ethernet is intended to support 802.3 compatibility, but at the next factor of ten higher speed. At such high speeds, of course, some things may be done differently closer to the physical layer. See www.10gea.org for more information.

FIBER DISTRIBUTED DATA INTERFACE (FDDI)

This section summarizes the base Fiber Distributed Data Interface (FDDI) capabilities, as well as the additional support for isochronous traffic in FDDI-II.

Basic Fiber Distributed Data Interface (FDDI)

The American National Standards Institute (ANSI) issued the first Fiber Distributed Data Interface (FDDI) standard in 1987 [X3.139], targeting high-performance LANs and campus backbones. Therefore, FDDI is the senior citizen in the class of other high-performance LAN standards. With age come maturity and stability, which many experts acknowledge as a virtue of FDDI [Saunders 96]. Unfortunately, FDDI's station management protocol was too complex, making the resulting chip sets and implementations expensive in comparison with other alternatives [Tannenbaum 96]. Also, since FDDI achieved only limited penetration in the marketplace, it never benefited from the volume production manufacture of integrated circuits.

Similar to Token Ring, FDDI also utilizes a ring topology made up of physical point-to-point fiber optic connections attaching a circular arrangement of stations. However, FDDI uses two counter-rotating rings instead of one for greater reliability, as illustrated in Figure 9-11. As indicated in the figure, dual attached stations (also called class A stations) connect to both rings. Data flows in one direction around the primary ring, with the secondary ring providing an alternate path that the dual attached stations use to form a longer ring in the event of a failure in the primary ring.

FDDI standards define physical interfaces for single-mode fiber using lasers, but most implementations use multimode fiber with cheaper (and safer) light-emitting diode (LED) transmitters. A special type of dual attached station, commonly called an FDDI concentrator [Saunders 96], supports simpler (and cheaper) single attached stations such as workstations and high-performance servers, as shown on the right-hand side of Figure 9-11. As shown on the left-hand side of the figure, single attached stations (also called class B stations) may connect to only the primary FDDI ring, but could be disconnected in the event of a primary

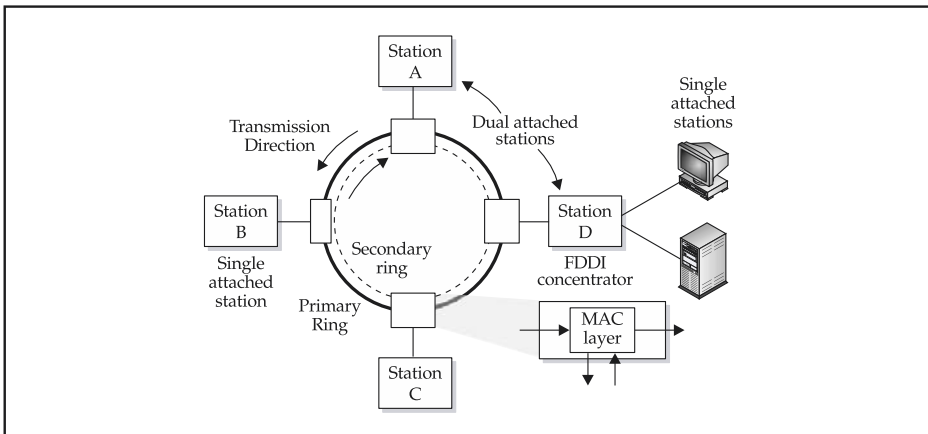


Figure 9-11. FDDI counter-rotating ring and station attachment configuration

ring failure. Typically, FDDI networks use a star-wiring arrangement, where a patch panel or hub allows for easier cable management, as well as the addition and deletion of stations.

Figure 9-12 shows a conceptual model of the FDDI protocol stack [Taylor 98]. As shown in the figure, each station has a Physical Medium Dependent (PMD) sublayer that defines optical levels and signal requirements for interfaces to the fiber optic cables via Medium Interface Connectors (MICs). The physical protocol (PHY) layer defines framing, clocking, and data encoding/decoding procedures. In a manner similar to Token Ring, powered-down dual attached stations convey the optical signal so that both the primary and secondary rings remain intact. Active stations implement a Media Access Control (MAC) sublayer that defines token passing, data framing, and the interface to the IEEE 802.2 Logical Link Control Layer (LLC), as indicated in Figure 9-12. The Station Management (SMT) function interacts with the PMD, PHY, and MAC sublayers, as shown on the right-hand side of the figure, to provide management services. SMT functions include provision of unique identifiers, fault isolation, station configuration, status reporting, and ring configuration. SMT also provides the means of inserting and removing stations from the FDDI ring.

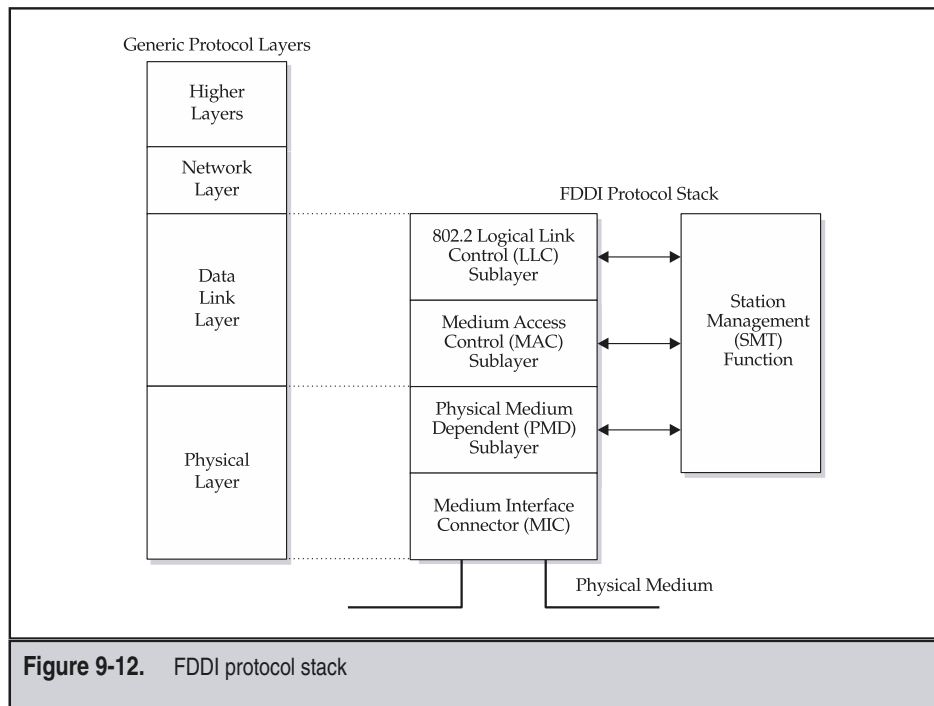
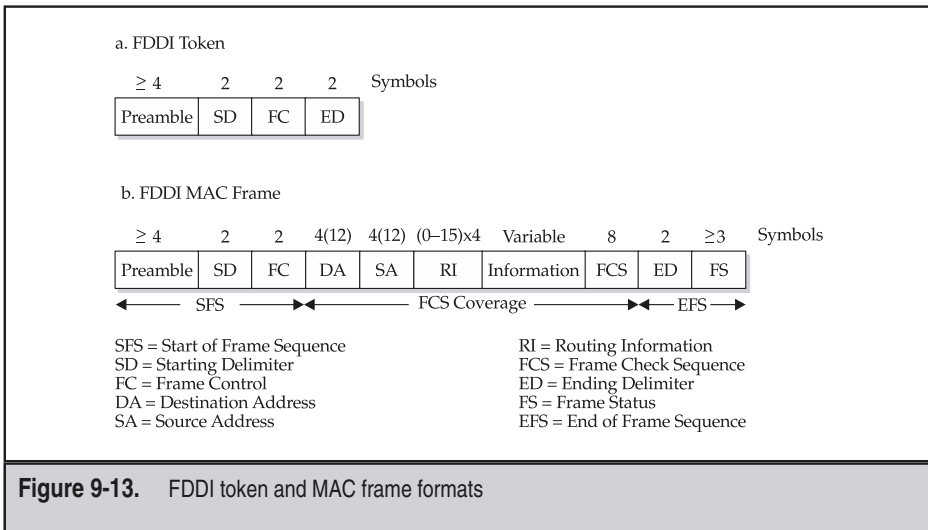


Figure 9-12. FDDI protocol stack

Basic FDDI operation is similar to that of 802.5 Token Ring, with one important exception. As in Token Ring, in FDDI, a station must first seize the token before transmitting. However, since an FDDI ring may contain a maximum of up to 1000 stations with up to 2 km between each station, or a total ring circumference of 200 km, the delay involved could be tens of milliseconds in waiting for the frame to circulate around the ring so that the transmitting station can remove it and reinsert the token. If FDDI operated as Token Ring does, and required other stations to wait for the token prior to transmission, throughput would be dramatically reduced. Therefore, the FDDI standard allows the transmitting station to insert the token immediately after completing transmission of its frame(s). Hence, subsequent stations can transmit their frames before the sending station removes its frame from the ring. This modification to the token passing protocol allows FDDI to achieve nearly 100 percent transmission efficiency under heavy loads in a manner similar to Token Ring. Unfortunately, FDDI suffers from the same issue of increased latency due to the token rotation time as the Token Ring protocol does. Furthermore, due to the shared media design of FDDI, the bandwidth available to each user decreases as more stations are added to the ring. Moreover, the variable delay caused by waiting for the token makes FDDI inappropriate for multimedia and voice applications. Realizing this, the FDDI designers defined a follow-on standard called FDDI-II to support real-time services such as voice and video, as summarized in the next section.

The ANSI X3.239 standard defines the FDDI MAC protocol. Note that the FDDI token and MAC frame shown in Figure 9-13 is similar to that used in Token Ring. The FDDI physical layer utilizes a scheme called 4 out of 5 encoding, where every four bits of the MAC layer protocol are actual encoded as five baud on the physical medium to reduce



the cost of FDDI equipment. In other words, each FDDI symbol either contains four bits of MAC information or performs special functions on the FDDI ring. The token and MAC frame both begin with a Preamble field containing at least four symbols. Other stations recover clocking from the preamble, since the line coding does not provide automatic timing recovery as it does on Ethernet. Next, a two-symbol Starting Delimiter (SD) field precedes a 2-symbol Frame Control (FC) field. The preamble, SD, and FC fields make up a Start of Frame Sequence (SFS), as indicated in the figure. In the token, the FC field has the token bit set to 1, while the FDDI MAC frame has a value of 0 in the token bit. The token then concludes with a 2-symbol Ending Delimiter (ED) field. The FDDI MAC PDU then contains 4- (or 12-) symbol Destination Address (DA) and Source Address (SA) fields. Next, an optional Routing Information (RI) field composed of 0 to 60 symbols in multiples of 4 symbols precedes an optional information field. An 8-symbol (32-bit) Frame Check Sequence (FCS) provides error detection for the fields indicated in the figure. An End of Frame Sequence (EFS) brackets the FDDI MAC frame with a 2-symbol Ending Delimiter (ED) and a Frame Status (FS) field of 3 or more symbols. The maximum total FDDI MAC frame length is 9000 symbols, or 4500 bytes.

Hybrid Ring Control (FDDI-II)

Recognizing the problems with basic FDDI in supporting multimedia traffic, ANSI issued a second version of the FDDI protocol, called FDDI-II or Hybrid Ring Control (HRC) [X3.186], in 1992. This effort also resulted in a modified Media Access Control layer (MAC-2) specification in 1994 [X3.239]. The aim of FDDI-II was to transport multiplexed asynchronous packet data along with isochronous circuit-switched data over FDDI LANs and MANs. Figure 9-14 shows the FDDI-II protocol structure. As part of the set of optional HRC layers, a new hybrid multiplexer layer sits between the physical and MAC sublayers, as shown in the shaded portion of the figure. The standard uses either the existing FDDI MAC layer or an enhanced version called MAC-2. The optional HRC capability also adds an Isochronous MAC (IMAC) sublayer supporting one or more Circuit Switching multiplexers (CS-MUX). The MAC and MAC-2 layers support conventional data applications via a set of asynchronous services using the IEEE 802.2 standard Logical Link Control (LLC) sublayer, as shown in the figure. The new IMAC sublayer supports circuit-type services, such as constant bit rate digitized voice and video. FDDI-II uses a deterministic multiplexing scheme to split the 100 Mbps bandwidth up into 16 synchronous frames operating at 6.144 Mbps, which is 96 times the standard 64 Kbps telephony rate we studied in Chapter 6. The choice of this unit of bandwidth supports exactly four standard North American DS1 circuits (4×24) or three standard International E1 circuits (3×32). For isochronous traffic, the node-to-node delay reduces to 125 μ s plus propagation delay, while the lower-priority asynchronous traffic operates as in basic FDDI similar to Token Ring. The hybrid multiplexer layer strips off the header and passes the frame to the appropriate MAC layer. Enhanced station management functions support the hybrid operation of isochronous and asynchronous data on the same physical ring.

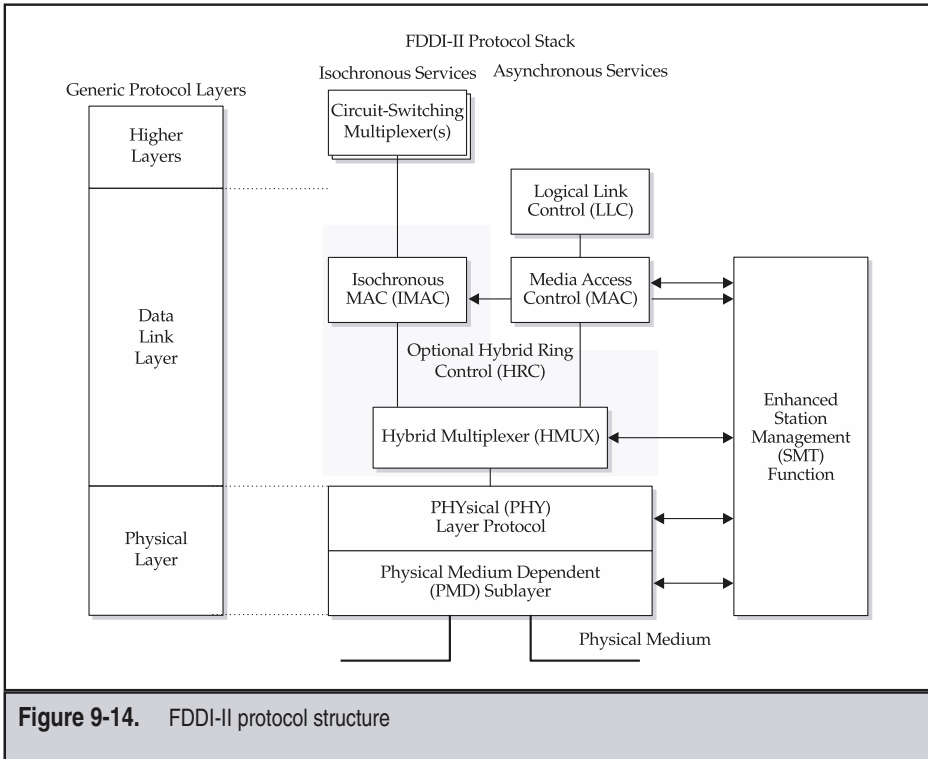


Figure 9-14. FDDI-II protocol structure

Theoretically, FDDI-II allows interconnection of PBX equipment to the LAN and MAN, combining voice and data communications on the same fiber. Practically, however, few vendors implemented FDDI-II, since other technologies that support multiple qualities of service, like 100VG-AnyLAN, low-speed LAN ATM network interface cards, and prioritized Ethernet cost much less.

BRIDGING CONCEPTS, SYSTEMS, AND PROTOCOLS

Bridging performs several critical functions in local area networks. Initially, bridges provided connectivity between LANs of the same media type—for example, Ethernet to Ethernet or Token Ring to Token Ring. Many network designers deployed bridges to scale local area networks beyond the limits of a single LAN collision domain to support greater distances, more hosts, or larger aggregate bandwidth. Inevitably, the generation of incompatible LAN standards created the need for more sophisticated bridges that could

translate between different LAN types as enterprises continued interconnecting local area networks. Enterprises first employed bridges to link LANs across the hallway, and then across entire continents in the 1980s. As we shall see, the design of bridged networks achieves plug and play operation to a greater extent than routed networks. But as the complexity of large bridged networks increased, a need arose for automatic topology discovery and reconfiguration. This section provides a summary of the key concepts, configurations, and protocols involved in bridging.

Bridging Context

As depicted in Figure 9-15, bridges operate on user data at the physical and MAC sublayers. Bridges provide more functions than LAN repeaters, which simply extend the distance of the LAN physical medium. Bridges use less processing than a router, which must process both the data link and network layer portions of packets. Since bridges operate at only the MAC sublayer, multiple network layer protocols can operate simultaneously on the same set of bridged LANs. For example, IP, IPX, and OSI network layer protocols can operate simultaneously on the same set of bridged Ethernets.

Communication between bridges using a bridging protocol occurs at both the logical link control (LLC) and the media access control (MAC) sublayers, as depicted in Figure 9-16. Similar to the notion of separate planes of operation for user and control data defined in ISDN, the bridging protocol operates separately from the forwarding of user MAC frames depicted in Figure 9-15. Bridging protocols automatically determine the forwarding port for the MAC frame bridging function. Examples of bridging protocols described later in this section are the Ethernet Spanning Tree Protocol and the Token Ring Source Routing Protocol.

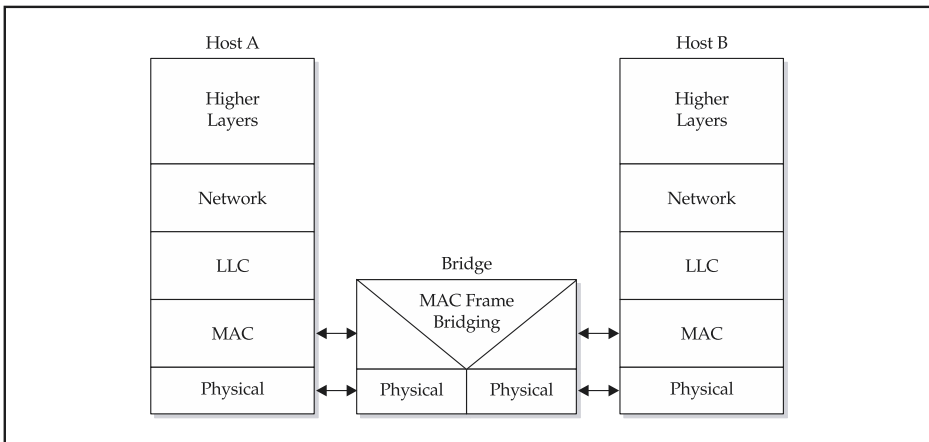
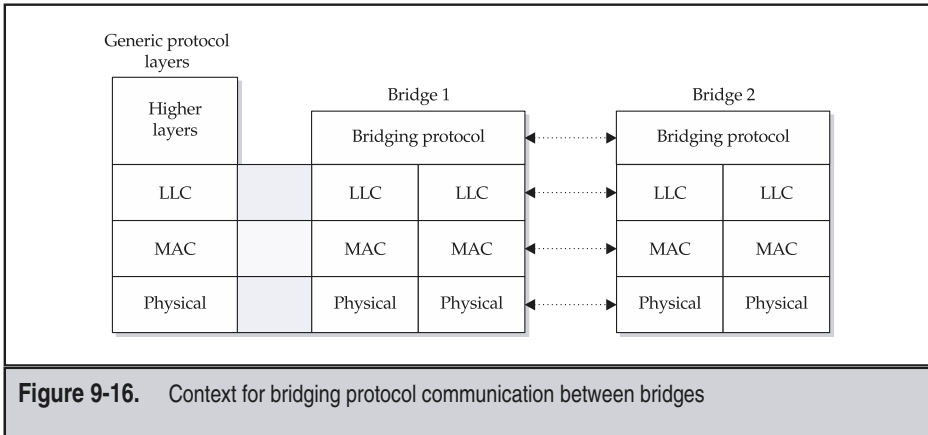


Figure 9-15. MAC bridging context for forwarding user MAC frames



A Taxonomy of Bridges

Bridges pass traffic from one LAN segment to another solely on the basis of the destination MAC address. If the destination address of the frame received by the bridge is not local to the bridge, the frame is obviously destined for another LAN. Different types of bridges use different methods to determine the port on which to send the frame. There are four major types of bridges: transparent (or spanning tree), encapsulating, translating, and source routing. This section briefly summarizes the key attributes of each type of bridge and then explores the spanning tree and source routing algorithms in greater detail. For a very readable, detailed treatment of bridging, see [Perlman 92]. We conclude with some observations on the topic of bridge design.

When operating in *transparent* mode, bridges at both ends of a transmission support the same physical media and link layer (MAC-level) protocols from the IEEE 802.X suite, although the transmission speeds may differ. Transparent bridges utilize the Spanning Tree Protocol described in the next section to automatically determine the bridged network topology and reliably forward frames according to the MAC address.

Sometimes, network designers require bridges to interconnect dissimilar LANs—for example, Ethernet and Token Ring. Unfortunately, the IEEE committees developing LAN standards did not agree on the same conventions, thus creating the need for MAC-level conversion. Although this may seem trivial, it's not: for example, even the bit order differs on these media! Bridges must support a *translation* mode to interconnect dissimilar physical LAN media and MAC sublayer protocols. A number of issues must be resolved for conversion between LAN protocols, including changing bit transmission order, reformatting, checksum recalculation, and control field interpretation [Tannenbaum 96]. Translation bridges cannot overcome different maximum frame sizes on different LAN types. Instead, the network administrator must ensure that users employ a maximum network layer packet size that doesn't exceed the most constraining LAN. For example, the maximum

Ethernet frame size is 1500 bytes, while a 4 Mbps token ring has a maximum frame size of 4000 bytes. In this case, all hosts should use a maximum packet size of less than 1500 bytes.

When operating in *encapsulation* mode, bridges at both ends of the transmission use the same physical and MAC-level LAN protocols, but the network between the bridges may be a different physical medium and/or MAC-level protocol. Encapsulating bridges place MAC frames from the originator within another MAC layer envelope and forward the encapsulated frame to another bridge, which then disencapsulates the MAC frame for delivery to the destination host.

Two typical examples of encapsulation bridging in local and wide area networking are encapsulation bridges to interconnect multiple Ethernet segments via a high-speed FDDI backbone and use of a serial WAN link to connect two encapsulation bridges using CSU/DSUs supporting bridging between two remotely located LANs.

Source route bridging can interconnect source and destination Token Ring LANs through three intermediate source route bridges connected by a transit LAN and/or a WAN serial link. As we study later in this section, source route bridging automatically distributes network topology information so that the source can determine the hop-by-hop path to the destination through specific intermediate bridges. The explorer packets used for this topology discovery add additional traffic to the network, but bridges and hosts cache the topology information for subsequent use. The name “source route bridging” derives from the fact that the source determines the entire route.

Spanning Tree Protocol

Early bridged LAN implementations simply broadcast MAC frames on every port except the one upon which they received the frame. Soon, network designers found that this basic form of bridging had several problems [Perlman 92]. The broadcast bridges worked fine as long as the LAN physical network topology had only a single path between any bridges. Unfortunately, if a single link or bridge failed, then the bridged network was down. The IEEE’s 802.1D Spanning Tree learning bridge Protocol (STP) [IEEE802.1D] solved this problem and delivered reliable, automatically configured network bridging. The Spanning Tree Protocol dynamically discovers network topology changes and modifies the forwarding tables to automatically recover from failures and revert to an optimized configuration as the topology changes.

The 802.1D Spanning Tree Protocol provides reliable networking by utilizing an algorithm that determines a loop-free topology within a short time interval. The STP algorithm runs continuously to react to link failures, as well as automatically add and delete stations and LANs. The resulting path through the bridged network looks like a tree rooted at the bridge with the lowest numerical MAC address. All other bridges forward packets up the tree toward this root bridge. Intermediate nodes then forward packets back down the tree to the destination leaf. The destination bridge then simply transmits the frame to the destination LAN.

Figure 9-17 illustrates a simple example of this property of the spanning tree algorithm. Figure 9-17a illustrates the physical topology of seven LANs, labeled A through G,

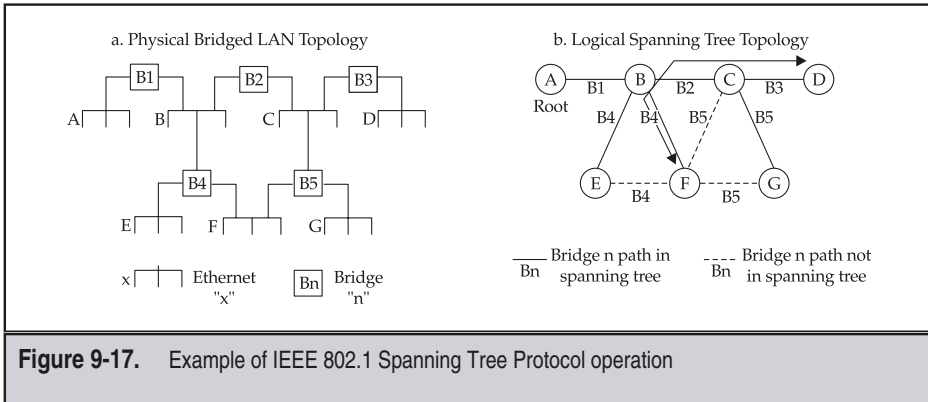


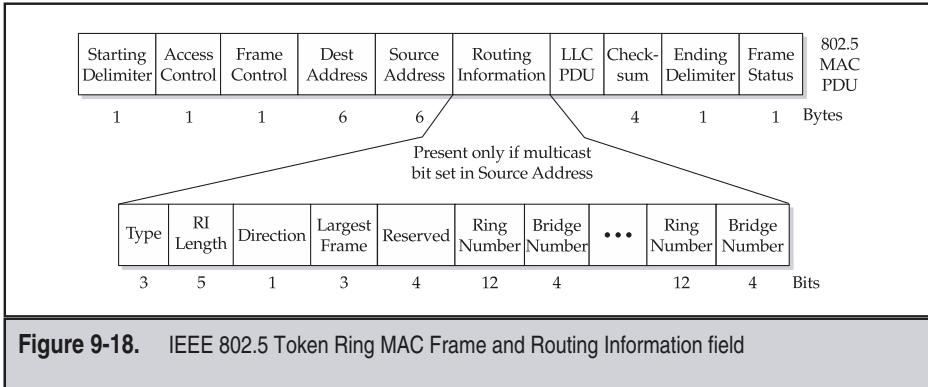
Figure 9-17. Example of IEEE 802.1 Spanning Tree Protocol operation

interconnected by five bridges, labeled B1 through B5. Observe that the physical topology has multiple paths between LANs B through G. However, the spanning tree algorithm resolves to a single logical topology of a tree rooted in the port on LAN A in Bridge 1 (B1), as shown in Figure 9-17b. Note that traffic in such a bridged network often does not take the most direct path. For example, LAN frames between LANs F and D will not flow through B5 and B3 but instead flow through B4, B5, and B3. Designers can control which links the STP bridge chooses when multiple parallel paths connect bridges by setting administrative costs in the STP algorithm. In addition, network designers can choose the MAC address utilized by the bridges to control the resulting topology to a certain extent. However, WAN network designers should carefully employ these techniques to minimize traffic flowing across the WAN in the spanning tree.

Source Routing Protocol

Although STP bridges offer the convenience of plug and play operation, they inefficiently utilize link and bridge port capacity, as illustrated in the previous example. The token group of IEEE 802.5, with support from IBM, responded to this challenge by designing the Source Routing Protocol (SRP). Each SRP LAN station specifies the end-to-end routing for each frame for bridging between token LANs. Hence, source route bridges utilize bandwidth more efficiently than spanning tree bridging; however, they require more configuration before they will operate. Stations utilize the SRP for frames destined to stations on other LANs by setting the multicast bit in the source address. This convention works because no station should ever transmit from a multicast address. Each frame contains a complete set of routing information that describes the sequence of bridges and LANs the frame must traverse from the source to the destination station. LAN stations obtain the information to compute this optimal path from explorer packets broadcast periodically throughout the bridged network.

SRP utilizes an addressing scheme in the routing information field illustrated in Figure 9-18, which shows the IEEE 802.5 (Token Ring) MAC PDU fields. The Routing



Information (RI) field starts with a two-byte control header, consisting of a type field to distinguish explorer packets from source routed packets, a length field that limits the source route to 14 hops, a direction bit indicating left-to-right or right-to-left route scanning, and a largest frame size indicator. The remainder of the RI field contains a sequence of 12-bit token ring and 4-bit bridge numbers, defining the hops from source to destination. In real implementations, each bridge uses a pair of ring-bridge number pairs, so in practice the maximum number of bridges traversed on a source route is seven. Note that the ATM Forum's Private Network-Network Interface (PNNI) and MPLS explicit routing also use a source routing paradigm in the interest of efficiently utilizing link and port capacity, as covered in Part 3.

Bridge Network Design

Careful future planning is required when deploying a bridged network solution. The network engineer who employs a bridged network may find that the design resembles a bridge designed to accommodate a horse and carriage that must now carry automobile and truck traffic. Bridges are best used in networks with few nodes that have a limited geographic extent. Bridging device capacities and speeds vary, supporting low-speed serial links up to DS1 or DS3 across the WAN, and 100 Mbps FDDI or 155 Mbps ATM in the LAN. Higher speeds are needed to support high-speed LANs connected to the bridge, such as 100 Mbps Ethernet and 16 Mbps Token Ring. Bridges provide either local or remote, or both local and remote support.

Although simplicity along with true plug and play operation are major advantages for bridging, there are some major disadvantages as well. Until a transparent bridge learns the destination LAN, it broadcasts packets on all outgoing LAN ports. When destinations are unreachable or have problems, applications resend data, further intensifying traffic overload conditions. Also, the spanning tree bridge uses LAN bandwidth inefficiently by sending all traffic up a tree toward a root node, and back out to destinations.

Discovery packets used by source route bridges to determine the network topology add additional network overhead traffic. The amount of memory in bridges for storing MAC addresses is also a limiting factor to the size of bridged networks.

Higher-layer protocols, such as NetBEUI, also generate significant amounts of broadcast traffic, which bridges forward to every host and also repropagates throughout the network. These phenomena and others can create broadcast storms, a problem that increases with the size of the bridged network and the number of attached users. Broadcast storms can bring a network to its knees. To minimize these problems, smart bridging techniques provide some level of traffic isolation by segmenting the bridged network into domains that restrict broadcast to a limited area. This containment method, coupled with a ceiling on the amount of multicast traffic, provides some control over broadcast storms.

ROUTING CONCEPTS, SYSTEMS, AND PROTOCOLS

Unlike most bridges, routers provide connectivity between like and unlike devices attached to local and wide area networks. Routers operate at the network layer protocol but usually also support link layer bridging. Routers have a common summarized view of the entire network topology, not just locally connected devices, and determine the next hop to forward a packet by considering many factors. The first generation of routers appeared at MIT, Stanford, and CMU in 1983 as the next step in the evolution following their ARPAnet predecessors three years earlier. Routers emerged into the marketplace over the last decade as the hottest thing in networking, with much more intelligence than bridge. The distinction blurs today because many so-called bridges, hubs, and LAN switches have been enhanced to also perform routing functions.

Larger networks usually implement some form of routing protocol that automatically discovers neighbors, distributes topology, and computes optimized routes. This section begins by defining generic router functions and operation. The text then surveys only the modern link-state routing protocols, and not the older historical protocols like RIP. We then cover how larger networks scale through subnetting as an introduction to the concept of hierarchical routing. Finally, the text introduces the subject of address resolution in LANs as background to similar concepts discussed in Part 5 regarding ATM's support for LAN emulation and IP over ATM.

Packet-Forwarding and Routing Protocol Functions

Routers operate on the user data stream at the physical, data link, and network layers to provide a connectionless service between end systems (or hosts), as shown in Figure 9-19. In contrast to bridges, routers operate on the fields in packets at the network level, instead of the MAC sublayer. Subsequent discussion calls this operation on the user data stream a *packet-forwarding* function. Of course, systems must employ the same network, transport, and application layer protocols when communicating via routers as they would with bridges.

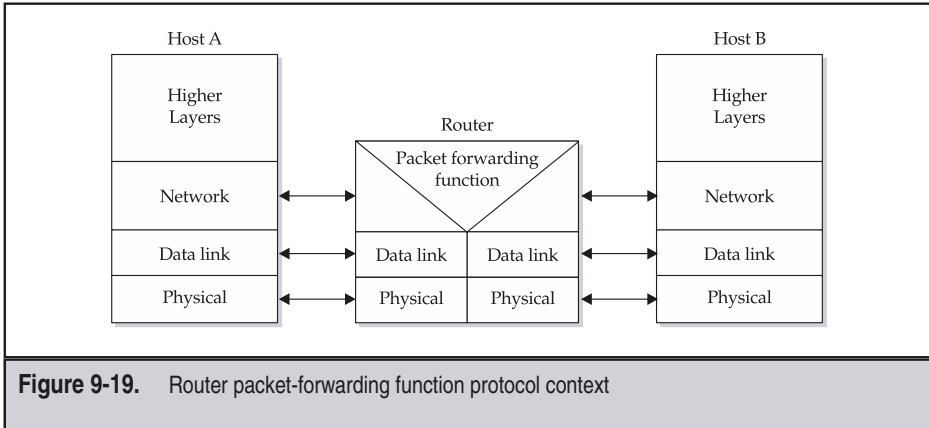


Figure 9-19. Router packet-forwarding function protocol context

Analogous to the separate planes of operation in bridging protocols, routers communicate via routing protocols as illustrated in Figure 9-20. A dotted line indicates the routing protocol communication between peer entities to distinguish from the packet-forwarding communication illustrated by solid lines in the figure. Routing protocols automatically determine the forwarding port for the packet-forwarding function. Examples of routing protocols described later in this section are the topology distribution (i.e., flooding) and the least-cost path determination algorithms.

Routers implement several interrelated functions, as illustrated in Figure 9-21. Starting on the left-hand side of the figure, routers interface to a variety of LAN or WAN media, encapsulating and converting between data link layer protocols as necessary. Thus, routers naturally interconnect Ethernet, Token Ring, FDDI, frame relay, and ATM networks.

The principal function of a router is packet forwarding, often implemented in hardware in high-performance machines. The packet-forwarding function contains a lookup

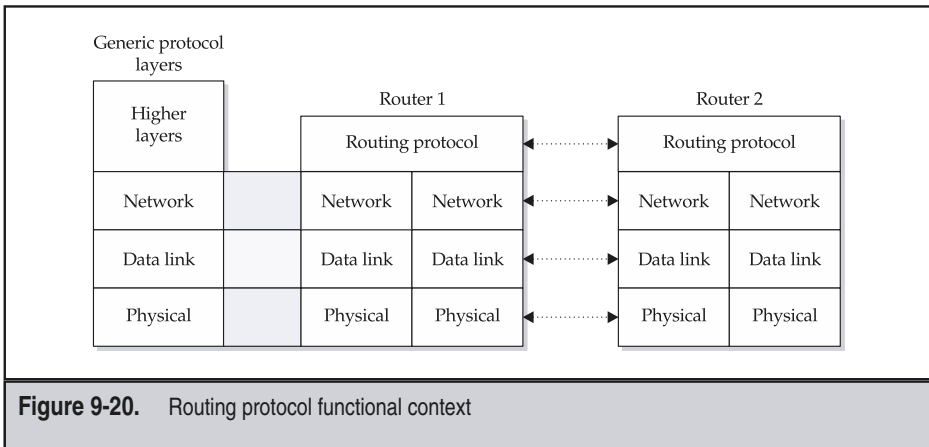


Figure 9-20. Routing protocol functional context

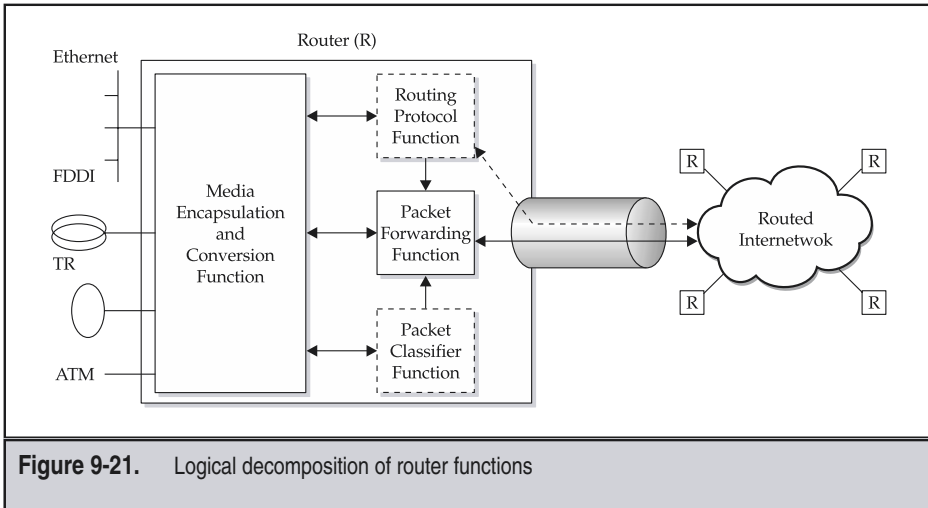


Figure 9-21. Logical decomposition of router functions

table that identifies the physical interface of the next hop toward the destination from the high-order bits contained in the received packet's destination address. The connectionless paradigm requires that each network node (e.g., a router) process each packet independently. Common per-packet processing functions required in real networks include: classifying packets according to addresses and other header fields like the Diffserv/TOS byte, queuing different packet flows for prioritized service, and data link layer conversions. Older routers implemented this complex processing in software, which limits throughput. Practically, using filtering to implement a firewall can limit router throughput significantly; however, new hardware-based routers avoid these bottlenecks.

Routers employ routing protocols to dynamically obtain knowledge of the location of address prefixes across the entire routed internetwork. The routing protocol engine fills in the next-hop forwarding table in the packet-forwarding function. Routing protocols determine the next hop from specific criteria, such as least cost, minimum delay, or minimum distance. Thus, the routing algorithm determines the best way to reach the destination address. A commonly used solution to this problem assigns each link in a network a *cost* and then employs a routing algorithm to find the least-cost routes. This cost may be economic, or it may reflect some other information about the link, such as the delay or throughput.

Thus, routing protocols discover network topology changes and flood these throughout the network. Once a router receives the updated topology data, it recovers from link or nodal failures by updating the forwarding tables. Routers employ protocols to continually monitor the state of the links that interconnect routers in a network, or the links with other networks.

Routers employ routing protocols to dynamically obtain knowledge of the location of address prefixes across the entire routed internetwork. The routing protocol engine fills

in the next-hop forwarding table for the packet-forwarding function. Routing protocols determine the next hop from specific criteria, such as least-cost, minimum delay, minimum distance, or least-congestion conditions. These routing protocols discover network topology changes and provide rerouting by updating the forwarding tables. Routers employ routing protocols to continually monitor the state of the links that interconnect routers in a network or the links with other networks. Routers often limit the number of hops traversed by a packet through use of a “time to live” type of algorithm, for example, the one described for IP in Chapter 8. Routers employ large addressing schemes, typically four bytes worth in IPv4 and eight bytes for IPv6. Routers also support large packet sizes. The other major advantage of routers is their ability to perform these functions primarily through the use of software, which makes future revision and support for upgrades much easier. A corresponding disadvantage is that the more-complex software implementations of routers may be less stable than simpler implementations of bridging LAN switches.

Historically, routing protocols have been divided into two groups: those that operate within the interior of a service provider network, and those that operate exterior to the enterprise or service provider network. Since in the early days of the Internet a router was called a gateway, these two groups of protocols were called an Interior Gateway Protocol (IGP) and an Exterior Gateway Protocol (EGP). Figure 9-22 shows a simple configuration of ISPs, enterprise networks, and end systems that illustrates this terminology. The lines in the figure indicate physical or virtual circuits that interconnect routers. Network administrators configure routers with a relatively small set of data, for example, address assignments and prefix lengths, administrative link costs, and routing policy information. Routers employ an IGP within their domain, such as the Open Shortest Path First (OSPF) or the Intermediate System to Intermediate System (IS-IS) protocol. Interconnected networks usually employ an EGP, such as the Border Gateway Protocol (BGP). The glue that binds the Internet together is ISPs using BGP to exchange information about how to reach specific IP address prefixes that belong to enterprise customers or groups of consumers. In simple configurations, like directly attached networks or devices, manually configured static routing on the exterior of the network is often used. The case of more-complex configurations—for example, the dual homing of enterprise site 2 to ISPs B and C—requires BGP. Dual homing is extremely important for enterprise locations that require high-availability service, since by proper configuration of BGP, traffic can either be load balanced across the two links or cause automatic switchover in the event of a link failure.

Link-State Routing Protocols Defined

The class of routing protocols that use the link-state paradigm replaced the earlier distance vector algorithm. The Internet first began using link-state routing in 1979. The link-state method overcame the slow convergence times of the prior distance vector method. Routers implementing a link-state protocol perform the following four basic functions [Tannenbaum 96]:

- ▼ They say hello to their neighbors, learn addresses, and collect routing “cost” or “distance” information.

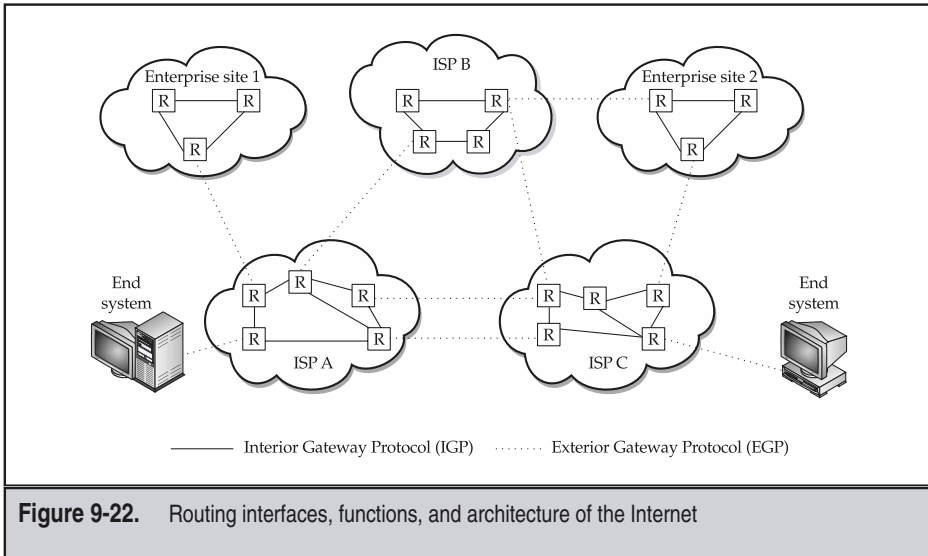
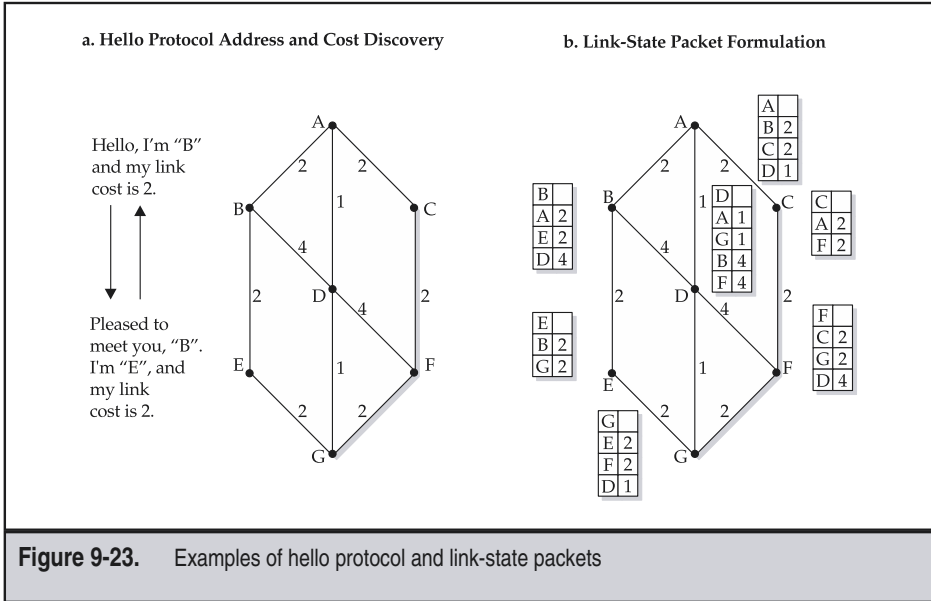


Figure 9-22. Routing interfaces, functions, and architecture of the Internet

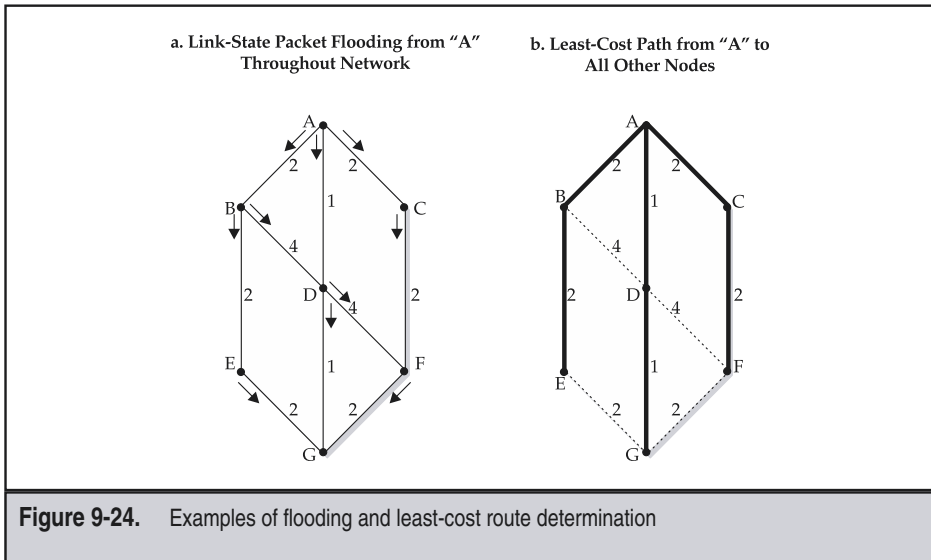
- They collect state information from all their links and place these in link-state packets.
- They reliably and efficiently “flood” the link-state packets throughout the network such that every router quickly converges to an accurate view of the entire network’s topology.
- ▲ They compute the least-cost path to every other router in the network.

Let’s look into each of these steps a little further with reference to the simple example in Figure 9-23 and Figure 9-24. Neighboring routers run a “hello” protocol once they boot up or once a link activates, as shown in Figure 9-23a. The hello protocol messages contain routing “cost” information, which may be economic information about the link or may reflect some other information such as distance, latency, or the capacity of a particular link. Routers also detect link failures when they stop receiving the periodic “heartbeat” of the hello protocol messages from a neighbor.

Each network node assembles its view of the current link states into a packet and forwards it to each of its neighbors, as shown in Figure 9-23b. In general, the node need only transmit the information that has changed since its last broadcast. The link-state packet identifies the source router in the first line, and the destination routers and link costs in each of the subsequent lines in the figure. Routers send link-state packets at start-up time or whenever a significant change occurs in the network. Examples of significant changes are a link or nodal failure, a link or node restart, or a change in the routing cost metric or reachability advertised by an adjacent network.



Intermediate nodes flood link-state packets to every other node in the network, as illustrated in Figure 9-24a for router A. *Flooding* involves replicating the link-state packets in an efficient and reliable manner such that each node quickly and reliably obtains an



identical copy of the link-state topology for the entire network. Nodes receiving multiple copies of the link-state packets discard duplicates. Additional fields in the link-state packets contain a sequence number and an aging count that eliminate duplicate packets and handle other error scenarios. Since each router must have the identical topology database, the link-state protocol acknowledges flooded packets.

Finally, each router computes the least-cost path to every other router in the network. Figure 9-24b illustrates the result of this calculation for router A with the thick solid lines. The net effect of the routing calculation, using the Dijkstra algorithm, is a minimum distance spanning tree rooted at each node, as shown by the upside-down tree rooted in router A in the figure. This concept embodies the essence of routing. Whenever a link or node fails, a new node or link is added, or a link or node is deleted; then the procedure repeats. We call the time for this entire process to complete the *convergence time*. Current link-state routing algorithms converge within seconds in moderate-sized networks. Rapid convergence to a common topology database in every network node is critical to achieving consistent end-to-end routing decisions. If nodes have different topology databases, then they may create routing loops. Since routing transients will inevitably occur, IP explicitly handles the possibility of routing loops through the Time to Live (TTL) field in the IP packet header, which prevents a packet from circulating indefinitely.

Three major implementations of link-state routing protocols dominate the market: the OSI's Intermediate System to Intermediate System (IS-IS) Routing Protocol, the Open Shortest Path First (OSPF) protocol [RFC1247], and the Private Network-Network Interface (PNNI) protocol [PNNI 1.1]. A popular implementation of link-state routing is the OSPF, which uses the Dijkstra, or Shortest Path First (SPF), algorithm for determining routing. All costs for links are designated on the outbound router port, so that costs may be different in each direction (unlike the simple example just given). OSPF also supports a limited form of hierarchical routing by sectioning the network into independent, lower-level areas interconnected by a backbone area.

OSPF routing supports three types of networks: point-to-point, broadcast, and nonbroadcast multiple access (NBMA). Point-to-point links join a single pair of routers. Broadcast networks attach more than two routers, with each router having the ability to broadcast a single message to multiple routers. Nonbroadcast multiple access networks, such as ATM, interconnect more than two routers but do not have broadcast capability. Standard OSPF supports only IP networks, unlike IS-IS, which supports multiple protocols simultaneously. However, proprietary extensions of OSPF are used to support FR, ATM, and other protocols. OSPF also supports bifurcated routing, that is, the capability to split packets between two equal paths. This is also commonly referred to as "load sharing" or "load balancing." As covered in Part 3, MPLS and ATM control plane protocols borrow heavily from the concepts of link-state protocols like OSPF and IS-IS, augmented to include constraints and traffic engineering extensions.

Routing and Logical IP Subnetworks (LISs)

A critical concept in routing in large networks is summarization of host addresses, called *subnetting*. If every router in a network needed to have a routing table entry for every host, then the routing tables would become unmanageably large. Furthermore, routing table size is not always the most critical constraint—sometimes the message processing to update the routing tables practically limits routing table size before physical storage does.

How does an IP host determine when routing is necessary? For example, how do two hosts on the same LAN know that they can directly transmit packets to one another without using routing? The answer, in general, is that when the two hosts are not on the same (bridged) LAN, then routing is needed. Historically, IP preceded LANs. Therefore, IP adopted the conventions of a subnetwork bit mask that constrains address assignments to allow hosts to determine whether routing was required solely by considering the source and destination addresses. The subnet mask convention dictates that IP hosts are in the same logical IP subnet (LIS) if a certain number of high-order bits of their IP addresses match. A station determines whether two IP addresses are on the same subnet by bit-wise ANDing the subnet mask with each address and comparing the results. If both addresses ANDed with the subnet mask result in the same value, then they are on the same subnet. As described in Chapter 8, Classless Inter-Domain Routing (CIDR) generalized the concept of subnet masks even further by allowing them to be of variable length for different subnetworks within the same administrative domain.

Subnet masks use the same IP address format where a certain number of the high-order bits all have the value of binary 1. In dotted decimal notation, it means that the four decimal values of a subnet mask have one of the values listed in Table 9-2. The table also lists the number of consecutive 1's in the subnet mask along with the number of networks and hosts allowed in the one-byte subnet mask. See Appendices C and D of [Spohn 96] for

Subnet Mask Number	Binary Subnet Mask	Number of Networks	Number of Hosts
255	11111111	254	0
254	11111110	126	0
252	11111100	62	2
248	11111000	30	6
240	11110000	14	14
224	11100000	6	30
192	11000000	2	62
128	10000000	1	126
0	00000000	1	254

Table 9-2. Valid Subnet Mask Decimal and Binary Values

more details on subnet mask values and the associated IP address values. For example, the old IP class A, B, and C addresses had implicit subnet masks as follows:

- ▼ Class A 255.0.0.0
- Class B 255.255.0.0
- ▲ Class C 255.255.255.0

The old class-based IP address structure suffered from a degenerate variant of Goldilocks's syndrome in the Three Bears fairy tale: Class A with over 16 million hosts was much too big for all but the largest networks, and class B with 65,000 addresses was also too big for most networks, yet Class C with only 254 hosts was too small for most networks. It was hard to find a "just right" answer. Although the 32-bit address enabled two billion networks, the inefficiency of assigning addresses in only these three sizes threatened to exhaust the Class B address space in 1993. CIDR usage of a variable-length subnet mask allowed Internet administrators to split up the IP address space more efficiently and keep the Internet growing until the next IP version (IPv6) arrives. CIDR also defines the concept of "supernetting," where multiple Class C style addresses are combined into a single route advertisement.

Let's look at a simple example—the Alamo Trader's Market in Texas. This network has an old-style class C address range of 198.62.193.1 to 198.62.193.254. Our example network has four routers, one at the headquarters in Austin and three remote sites at Dallas, Houston, and San Antonio, as illustrated in Figure 9-25. Each site requires up to 10 hosts per router; therefore, we can use the subnet mask of 255.255.255.240, since it allows up to 14 hosts per network (i.e., site), as shown in Table 9-2. We see in the example how most sites, like Dallas, use only three hosts today but can expand to support up to 13 total hosts (198.62.193.34 to .46) with one reserved (.47). This choice allows the network administrator to add more sites in the future. The 14 network addresses available under the subnet mask 255.255.255.240 are (see [Spohn 96] Appendix C):

- ▼ 198.62.193.16
- 198.62.193.32
- 198.62.193.48
- 198.62.193.64
- 198.62.193.80
- 198.62.193.96
- 198.62.193.112
- 198.62.193.128
- 198.62.193.144
- 198.62.193.160
- 198.62.193.176

- 198.62.193.192
- 198.62.193.208
- ▲ 198.62.193.224

The network administrator assigns the address 198.62.193.16 to the headquarters subnetwork in Austin, 198.62.193.32 to the Dallas subnetwork, 198.62.193.48 to the Houston subnetwork, and 198.62.193.64 to the San Antonio subnetwork, as shown in Figure 9-25. The administrator assigns addresses to hosts and servers at each of the locations. Note that we leave out the common Class C address prefix 198.62.193 for clarity in the following. Starting in the Dallas 198.62.193.32 subnetwork, the administrator assigns addresses .34, .35, and .36 to the three hosts (Ken, Sue, and Julie) attached to the router port on the Ethernet segment with address .33. Note that the administrator can add up to 10 more addresses within this subnet (.37 through .46). Moving on to the next subnet, Houston, the network administrator assigns .49 to the router interface on the Ethernet segment and .50, .51, and .52 to the hosts named Bill, Kelly, and Joe, respectively. Finally, she assigns addresses within the San Antonio subnet to the router interface on the Ethernet segment (.65), Rodney (.66), Kim (.67), and Steve (.68). The same address assignment process also applies at the Austin location.

The network administrator in our example now must assign IP addresses and subnet masks to the WAN links in the example in Figure 9-25. Planning for growth of her network,

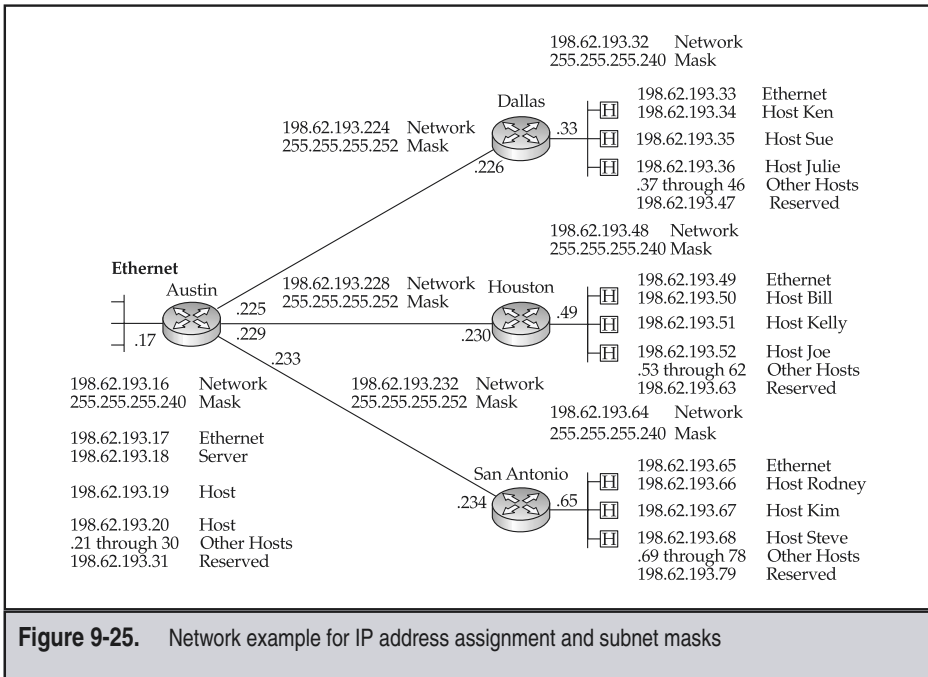


Figure 9-25. Network example for IP address assignment and subnet masks

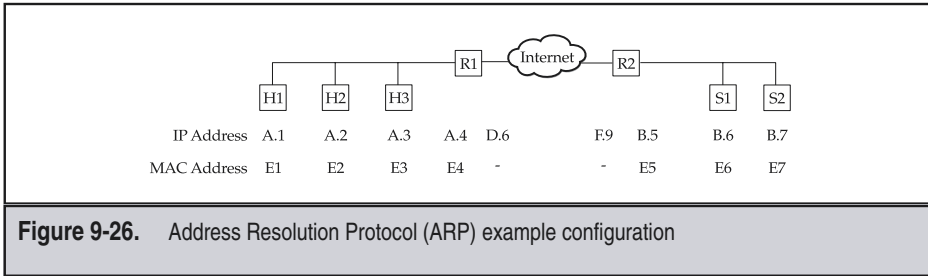
the administrator chooses a different subnet mask for the point-to-point links, since the 255.255.255.240 subnet mask reserves 14 addresses per WAN link when, in fact, only two are needed. This is possible because the routers run a version of OSPF that supports CIDR's variable-length subnet masks. Therefore, the administrator assigns all three WAN links the same subnet mask of 255.255.255.252, which allows exactly two hosts per subnet, as identified in Table 9-2. The network administrator chooses to use this longer subnet mask to split up the 198.62.193.224 network, leaving room to add other subnetworks for planned expansion to other cities in Texas. Hence, the valid network numbers are .224 through .252 in increments of 4 (see [Spohn 96] Appendix D). The administrator assigns the network address 198.62.193.224 with a subnet mask of 255.255.255.252 to the link between Austin and Dallas, as shown in Figure 9-25. The router port in Austin gets the IP address 198.62.193.225, and the Dallas router gets 198.62.193.226. Note that these address choices avoid the all-zeros and all-ones host addresses on the WAN link subnet. The other WAN links are then assigned sequential addresses under the longer 255.255.255.252 subnet mask as shown in the figure.

This assignment allows the administrator to add another 9 subnets under the 255.255.255.240 mask and another 11 WAN links out of the 198.62.193.224 network under the 255.255.255.252 mask. If the routers had not used CIDR, then the administrator would need to assign a separate network to each WAN link, wasting three network address blocks in the process, and thus limiting potential expansion to 6 additional subnets and/or WAN links. Patting herself on the back for such a forward-looking address assignment and clever use of variable-length subnet masks, the LAN administrator of our example heads out of the office for a well-deserved, ice-cold Lone Star beer at her favorite watering hole on Sixth Street in Austin.

Address Resolution Protocol (ARP)

Another concept from local area networking used in several ATM address protocols is that of address resolution. In modern networks, most hosts attach to local area networks with Network Interface Cards (NICs) that understand only MAC-level addresses. When a host wishes to send a packet to another host using a network-level (e.g., IP) address, then the sending host must first determine the MAC-level address of either the destination host or the next-hop router that can progress the packet toward the destination. Note that most real-world applications involve an additional step of first obtaining the network address from a name, such as a Domain Name System (DNS) server. In the interest of brevity, we omit this step in the following examples. From the subnetting discussion in the previous chapter, a host learns whether the destination is in the same subnet by bit-wise ANDing the destination address with the subnet mask. If the destination is in the same subnet, then all that remains is to determine the MAC address of the destination host. If the destination address is on another subnet, then the source must forward the packet to the preconfigured default router address. In this case, the source must determine the MAC address of the default router attached to its subnet.

In RFC 826 [RFC826], the IETF defined the *Address Resolution Protocol (ARP)* to perform exactly this function. Let's see how it works for two examples with reference to the configuration of Figure 9-26. Here we depict several hosts (labeled H1 through H3)



connected to an Ethernet with Router R1, which, in turn, connects to the Internet with IP and Ethernet addresses, as indicated in the figure. We also show another router, R2, connected to the Internet and a local Ethernet with two servers labeled S1 and S2 with IP and Ethernet addresses indicated in Figure 9-26. Note that each router has a separate set of addresses for each port. The serial ports on the routers connected to the Internet do not have Ethernet MAC addresses.

In our first example, Host 1 (H1) wishes to send a packet to Host 3 (H3). Examining H3's IP address using its subnet mask, H1 determines that H3 is on the same subnet. Therefore, H3 prepares an ARP packet containing H3's IP address (A.3) and sends it to the "all Ethernet" broadcast address. H1, H2, and H3 all receive the ARP packet, but only H3 responds with its Ethernet address (E3) in the ARP response packet. Now H1 can place its IP packet addressed to H3 inside an Ethernet MAC frame addressed to E3, and send the MAC frame over the Ethernet. In order to avoid performing this procedure for every packet, hosts cache the resolved Ethernet addresses for recently used IP addresses. Eventually, the ARP cache times out and the preceding process must be repeated. Also, since H3 will typically need to respond to H1's packet, the ARP protocol includes the following clever way to avoid another broadcast message. The ARP packet sent by H1 includes H1's IP address (A.1) and Ethernet MAC address (E1), so that not only can H3 copy this into its ARP table, but so can every other station on the Ethernet. A station may also send an ARP request for itself in a process called *gratuitous ARP* to flush out the cache of other stations and use the current MAC address associated with a particular IP address.

In our second example, Host H1 wishes to send an IP packet to Server S2 at IP address B.7. Comparing S2's IP address to its own ANDed with the subnet mask, Host H1 determines that S2 is not on the same subnet. Therefore, H1 prepares to send the packet to the default router IP address preconfigured by the network administrator in its storage, namely, address A.4 for Router R1. For the first packet sent to R1, Host H1 must ARP for the Ethernet address of A.4 using the preceding procedure. Once Host H1 receives the ARP response from R1, it updates its ARP cache to indicate that Ethernet address E4 corresponds to the IP address of the "default router," A.4. The default router gives, in effect, a target for all destination IP addresses on a subnet different than the sender's. Next, Host H1 takes the packet addressed to S2 (with IP address B.7), places it in a MAC frame with address E4 (i.e., the MAC address of the default router R1), and transmits it on the Ethernet. R1 receives this packet and examines the destination IP address. From the routing

protocol that R1 has been running with the Internet, it determines that the next hop is on the port with IP address D.6. The Internet routes the packet and eventually delivers it to the port on Router R2 IP address F.9. Router R2 compares the destination IP address (B.7) against its internal forwarding table and determines that the interface with IP address B.5 is on the same subnet as the destination address. If this is the first packet destined for IP address B.7, then router R2 must send an ARP packet on the Ethernet to determine the MAC address. Once Router R2 stores the mapping of IP address B.7 to Ethernet MAC address E7 in its ARP cache, it can forward the MAC frame on to S2.

Although these examples may seem somewhat complicated, computers repeatedly perform these procedures very rapidly without difficulty. In fact, if you compare ARP with the possibility of manually configuring all of the mappings between IP addresses and MAC addresses, the inherent simplicity of the concept becomes obvious. The preceding examples illustrate the minimum set of addresses that must be configured in every IP host, namely, the station's own IP address, the default router's IP address, and the DNS address.

BRIDGING AND ROUTING SYSTEMS DESIGN

A great sage once wrote, "bridge when you can, but route when you must." The reason this is true is that bridges offer true plug and play operation, while routed networks require at least some configuration of hosts and routers before they will work. Thus, installing a bridged network is simpler, but scales to only a limited network size. Some devices blur the distinction between routing and bridging by implementing more sophisticated, proprietary bridging protocols that perform some network layer functions.

While routing is more complex, it also provides more features and advantages over bridging, but at a price. Routers *dynamically* reroute traffic over, for example, the least-cost path. Routers reduce the danger of broadcast storms by terminating broadcast sources, such as NetBEUI or Banyan Vines. Routers allow a network designer to build a hierarchical addressing scheme that scales to very large networks, as the construction of the global Internet proves. Routers also provide filtering capabilities similar to those in bridges to restrict access to known users and can also be programmed through filters to block out specific higher-layer protocols in a process commonly called a "firewall." Routers have the additional flexibility to define virtual networks within a larger network definition. Routers using IP solve packet-size incompatibility problems by fragmenting larger packets into smaller ones and reassembling them. However, this solution should be used with care, since it significantly impacts performance due to the additional software processing required.

However, routers do have a few disadvantages. Routing algorithms typically require more system memory resources than bridges, and addressing schemes that require specialized skills to design and manage. Also, true routers cost somewhat more than simple bridges and hubs. Modern routing algorithms and implementations (e.g., IS-IS, OSPF, and BGP) are comparable to bridging (such as STP) in the amount of bandwidth overhead required for topology updates. Many router vendors have implemented multiple

processors within the network interface card, and faster platforms and processors (such as RISC machines) to eliminate throughput problems caused by increased traffic loads of routing protocols. Table 9-3 shows a comparison of bridge and router uses and capabilities.

It is a good idea to bridge when you desire simplicity; have the same LAN medium type across the entire network; have a small centralized LAN with a simple topology; or need to transport protocols that cannot be routed, such as NetBIOS and DEC LAT. Select routing when you want to route traffic by network parameters like least-cost route; have multiple MAC protocol environments; have large, dynamic networks with complex topologies; want optimized dynamic routing around failed links over paths that run in parallel; or have network and subnetworking requirements.

Although many workstations, PCs, and servers have built-in bridging and routing functions, beware of hidden implications. While this packaging seems to offer the cost and management advantages of using only a single device, such products often suffer from limited support, scalability, upgradability, and manageability. Choosing the right device that will grow with your network pays back in benefits, such as lower upgrade costs with minimal operational impact. One option is to purchase a full router rather than a bridge—you may not need routing today, but as your network grows, you may be able to upgrade without having to replace the LAN device. Avoiding the operational impact of downtime and addressing changes may be well worth the additional cost of a later upgrade. Beware, though, that the router may be one generation of technology behind before you finally exploit its routing capability. It may be less expensive in the long run to purchase a router with port expansion, rather than taking the network down and installing

Function	Bridging	Routing
Network addressing	No	Yes
Packet handling	Interprets packet at MAC layer only	Interprets packet at data link and network layers
Packet-forwarding efficiency	Poor for spanning tree, good for source routing	Good for least cost (OSPF), moderate for distance vector (RIP)
Configuration required	None, except for source routing	Some always required, can be quite complex
Priority schemes	No	Yes
Security	Based on hardware isolation of LAN segments	Based on processor-intensive filtering of each packet

Table 9-3. Comparison of Bridging with Routing

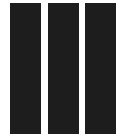
a larger or more feature-rich router later. Port sizing and traffic growth patterns typically dictate the size of the router required. Paying careful attention to network design can help you make the right hardware and software design decisions. For further information on the trade-offs between bridging, routing, and overall network design, see References Perlman 92, Tannenbaum 96, and Spohn 96.

REVIEW

The chapter began with an overview of terminology commonly used in computer communication networks. Next, we surveyed important terminology and concepts at the data link layer critical to LANs, such as Logical Link Control (LLC) and Media Access Control (MAC). The text then covered the most commonly used LAN media: Ethernet, Token Ring, and the Fiber Distributed Data Interface (FDDI). The coverage continued with the first technique used to connect LANs—bridging. Finally, we covered the pivotal concept of routing, including examples of shortest path routing, subnetting, and address resolution. Armed with this knowledge, you now have the background to understand the descriptions in Part 3 of how ATM and MPLS support traffic engineering for IP networks. It also provides background for the descriptions of ATM support for LAN Emulation (LANE), Classical IP over ATM, and Multiprotocol Routing over ATM (MPOA). Finally, this material also provides background useful for placing in perspective the evolving work described in Part 4 on Multiservice emulation and virtual private networks (VPNs) over MPLS.



PART



Foundations of ATM and MPLS: Protocol and Structure

This part describes the basic concepts of ATM and MPLS, the foundational protocols and their structure. Asynchronous Transfer Mode (ATM) will be described within the ITU-T B-ISDN model and the complementary ATM Forum specifications. We summarize how Multiprotocol Label Switching (MPLS) sprang from early vendor-proprietary solutions that culminated in the IETF MPLS architecture and related specifications. The text explains how a central focus of MPLS was initially to support IP more effectively than any preceding technology,

and also introduces background related to recent extensions in support of virtual private networks and services and protocols other than IP.

Chapter 10 provides a high-level introduction to ATM and MPLS, while also introducing the B-ISDN protocol model based on ATM and the evolution of IP protocols toward MPLS. This model employs user, control, and management planes in addition to the concepts of protocol layering already discussed in Part 2 that structure the standards and interoperation of various aspects of ATM and B-ISDN. Chapter 11 then covers the lowest two layers of the protocol reference model: the physical layer, the ATM layer that introduces the cell structure, and the MPLS layer and label structure. We will see that a mapping of these protocols to the rigid OSI layered model is not always possible. Chapter 12 next covers the ATM adaptation layer (AAL), which provides support for higher-layer services, such as circuit emulation, voice, video, and data packets. This chapter also describes the early approaches evolving to support multiple services over MPLS and IP.

Chapter 13 covers a planar protocol model of ATM and MPLS. This model is composed of the user, or forwarding, plane described in Chapter 11, as well as the control plane composed of signaling and routing, which are described later, in Chapters 13, 14, and 15. The chapter describes the overall high-level MPLS architecture as a means to help the reader place the protocol-specific details in their proper context. Chapter 14 covers the basic routing protocol concepts used in MPLS: link-state and path vector protocols. We then look at the specific protocols used to support the MPLS framework. The text summarizes important aspects of the BGP, IS-IS, and OSPF routing protocols in this regard. The chapter also examines the label distribution protocols used in support of establishing MPLS label switched paths. We then describe examples of how this framework can be used, and the current functionality available within stable standards documentation. Chapter 15 concludes Part 3 by covering the specifics of the ATM PNNI signaling and routing protocols, including a description of ATM UNI signaling along with examples of the currently available functions.

Both the ATM and MPLS protocols are currently undergoing tremendous enhancement in functionality and applications. In this part, we focus primarily on well-defined standardized features, while in Part 8, we come back to the more uncharted territory of some features that appear likely to emerge as standards.

CHAPTER 10

Introduction to ATM and MPLS

This chapter introduces the reader to the basic principles and concepts of ATM and the standards structure developed by the ITU. Next, we look at ATM through its various different faces: an interface, a protocol, a technology, integrated access, a scalable infrastructure, and a service. To better understand the origins of MPLS paradigms, we summarize precursor work, such as IBM's ARIS and Cisco tag switching, and then introduce the culmination of this work in the standard IETF MPLS architecture. The chapter concludes with a discussion regarding trade-offs involved in the use of cells versus frames.

INTRODUCTION TO ATM AND B-ISDN

This section describes the Broadband Integrated Services Digital Network (B-ISDN) protocol model and structure. We then present the standards vision of how B-ISDN interconnects with N-ISDN, SS7, and OSI.

B-ISDN Protocol Reference Model

The protocol model for the ITU-T's B-ISDN builds upon the foundation of ATM, as shown at the base of Figure 10-1 from ITU-T Recommendation I.321, which provides a structure for related recommendations. The top of the cube labels the planes, which stretch over the front and side of the cube. The user plane and the control plane span all

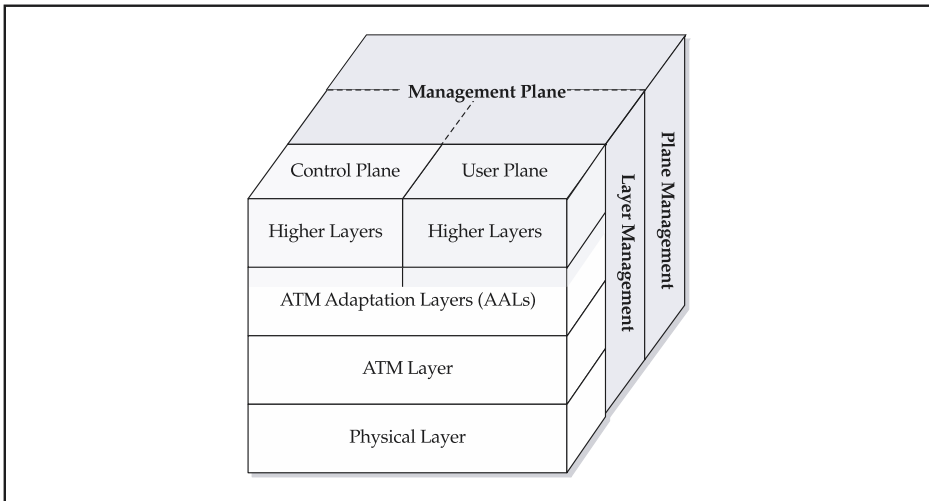


Figure 10-1. B-ISDN protocol model

the layers, from the higher layers down through the AALs, which can be null, to the ATM layer and the physical layer. Therefore, the physical layer, the ATM layer, and the AALs are the foundation for B-ISDN. The user and control planes make use of common ATM and physical layer protocols and use some of the same AAL protocols. However, service-specific components of the AALs and the higher layers differ according to function. Therefore, ATM provides a common foundation over a variety of physical media for a range of higher-layer protocols serving voice, video, and data. All this makes sense if you consider that the original goal for the ATM framework was to support all services.

ITU-T Recommendation I.321 further decomposes the management plane into layer management and plane management. As shown in Figure 10-1, layer management interfaces with each layer in the control and user planes. Plane management has no layered structure and is currently only an abstract concept with little standardization. It can be viewed as a catchall for items that do not fit into the other portions of this model, such as the role of overall system management.

B-ISDN Architecture

Figure 10-2 depicts the vision of how B-ISDN could interconnect with N-ISDN, SS7, and OSI as defined in CCITT Recommendation I.327, which complements the N-ISDN architecture defined in Recommendation I.324. Signaling System 7 (SS7) is the signaling

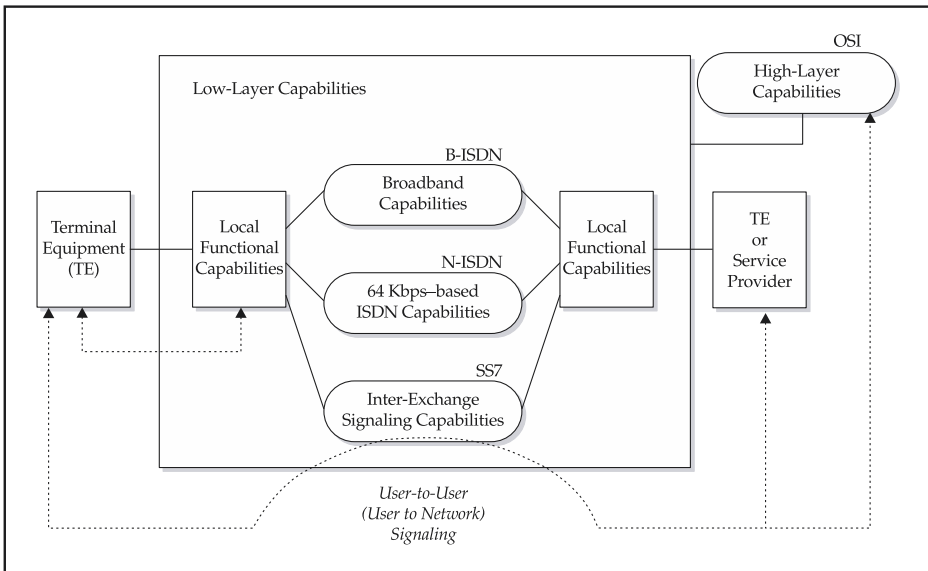


Figure 10-2. Integrated B-ISDN, N-ISDN, SS7, and OSI architecture

protocol that connects switches within a telephone or N-ISDN network. As shown in the figure, SS7, N-ISDN, and B-ISDN are lower-level capabilities that interconnect terminal equipment (TE) or service providers through local functional capabilities. SS7 provides out-of-band interexchange signaling capabilities for telephony and N-ISDN, while N-ISDN provides signaling capabilities for TDM-based services, X.25, and Frame Relay. B-ISDN provides bearer services of various types and signaling. All of these services support higher-layer capabilities. Initially, B-ISDN covered interfaces of speeds greater than 34 Mbps, and hence the choice of the adjective “broadband.” However, the subsequent standardization of 1.5 and 2 Mbps physical interfaces for B-ISDN—which are also standardized for N-ISDN, as summarized in Chapter 6—blurs the distinction with other protocols such as Frame Relay when only speed is considered.

OVERVIEW OF THE APPLICATION OF ATM

ATM plays many roles in modern networks. First, it provides a User-Network (UNI) interface protocol for the simultaneous transfer of voice, video, and data. Second, it acts as a signaling protocol for controlling ATM services. Next, ATM multiplexers and switches utilize ATM as a technology to implement robust, large, and fast switching machines. In addition, many service providers view ATM as an economical, integrated network access method, and as a scalable core network infrastructure. Figure 10-3 illustrates these concepts in a typical ATM network configuration. Let’s now explore each in more detail.

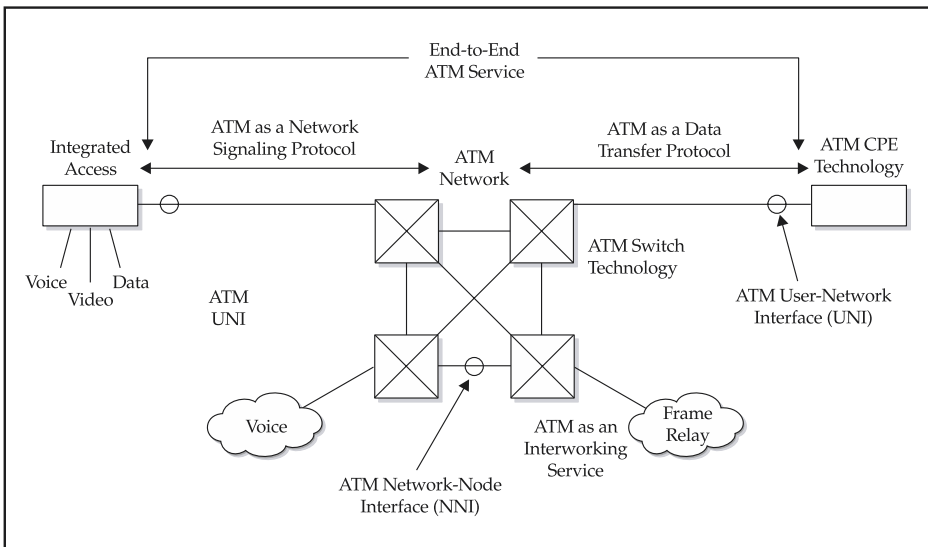


Figure 10-3. ATM's many roles in a network

ATM as a Technology

In a most basic sense, ATM is a technology defined by protocols standardized by the ITU-T, ANSI, ETSI, and the ATM Forum introduced in the previous chapter. ATM is a cell-based switching and multiplexing technology designed to be a general-purpose, connection-oriented transfer mode for a wide range of services. ATM technology comprises hardware and software conforming to ATM protocol standards, which provide multiplexing, cross-connect, and switching function in a network. ATM technology takes the form of a network interface card, router, multiplexer, cross-connect, or intelligent switch in customer premises equipment (CPE). Today, ATM prevails in the switch market and as a WAN interface on traditional data communications products such as routers and hubs, and also to a more limited extent within ATM end systems (NICs) and applications (APIs) that probably will still provide niche solutions in the future, but not widespread commercial use. Carriers use ATM switches of different sizes and capabilities on the edges of their networks, as well as in the backbones. The late 1980s ushered in early prototype central office (CO) ATM switches. The traditional customer premises multiplexer and switch vendors then adopted ATM in the early 1990s. The mid-1990s ushered in the next generation of CO switches, some modified from experience gained from trials and customer premises equipment (CPE) designs. CO switches continued to evolve rapidly in the mid-1990s, becoming larger, faster, and more capable than preceding generations. Simultaneously, router and hub manufacturers began building ATM interfaces for their existing models, as well as including ATM capabilities in their latest designs.

In the late 1990s, computer vendors built ATM network interface cards (NICs) for workstations and personal computers, and the loaded per-port or per-workstation cost of ATM to the NIC approached, but never fell to, the level of a 100 Mbps switched Ethernet LAN. Furthermore, the configuration complexity of an ATM NIC card versus a plug-and-play Ethernet card increased the overall cost of ATM LANs and stymied their adoption. ATM-based LANs did deliver such capabilities as guaranteed capacity, QoS, and virtual networking that Ethernet solutions did not have. But, as described in Chapter 9, the IEEE augmented the Ethernet standard in the late 1990s to support prioritization to achieve QoS and virtual LAN (VLAN) tags to support virtual networking, rendering obsolete some of the differentiation that an ATM-based LAN had. Operating system and application program interface (API) software, developed specifically for ATM-based systems toward the vision of a true end-to-end homogeneous ATM network, was initiated by a few vendors but never realized significant deployment and use because Ethernet LANs became cheaper and easier to operate with nearly comparable features. Today gigabit and 10 gigabit Ethernet are also considered as a viable wide area solution. We will discuss reasons for this in Part 8 and also touch on this when we discuss the IP network solutions provided by ATM protocols such as LANE and MPOA in Chapters 18 and 19.

ATM as a Protocol

ATM is a protocol designed to switch any type of traffic over a common transmission medium by offering to a user connections that are able to perform within a set of detailed conformance parameters that can be tailored to the specific service needs. The ATM protocol therefore offers service provider networks the ability to simultaneously support

video, voice, and data as an evolutionary successor to narrowband ISDN-type services, as well as to support new broadband capabilities. ATM specifies interworking with legacy protocols such as Frame Relay, SMDS, and IP. The ATM Forum's LAN emulation (LANE) and the Multiprotocol over ATM (MPOA) standards enable seamless interworking between ATM-powered devices and legacy LANs employing Token Ring, Ethernet, or FDDI, as further discussed in Chapter 18. Additionally, the ATM Forum has specified a sophisticated private ATM network NNI in the Private Network-Network Interface (PNNI) specification that supports automatic configuration, constraint-based routing, network resource allocation, hierarchical scalability, and resilience to failures. ATM was envisioned to be the future infrastructure for the "next generation"-type services, as well as providing a native ATM service interface. As we describe the numerous protocols that today make up the ATM "protocol suite," we will observe that although the foundation for the protocols is quite mature, several aspects of the effort to support all services have not been commercially successful. Throughout this part and the next, we will point out what protocols have enjoyed commercial success and what might never become widely deployed.

ATM as an Interface

ATM is an interface defined between the user and a network, as well as an interface between networks. Well, what precisely is an interface? An interface defines physical characteristics, ATM cell format, signaling, and management processes. Interfaces also provide the boundary between different types of hardware, while protocols provide rules, conventions, and the intelligence to pass voice, video, and data traffic over these interfaces. Internodal or internetwork interfaces also address aspects unique to interconnection between nodes and connections between networks. Standards give different names to the particular context of a physical interface connecting ports on devices in an ATM network. Figure 10-4 illustrates the commonly used terminology for ATM reference configurations.

The ATM User-Network Interface (UNI) is defined between the user equipment or end system (ES) and switches, also called intermediate systems (ISs). The figure illustrates the ATM Forum terminology and context for private and public UNIs. A private interface is used in the context of equipment wholly owned by one enterprise, while a public interface denotes an interface from a customer site into an ATM service provider node, to which many other customers may be connected. Standards call the connections between ATM switches either a Network-Node Interface or a Network-Network Interface—employing the same NNI acronym to indicate this particular connection of ports. A similar notation applies the adjective "private" in an intraenterprise context, with the public adjective referring to interfaces between service provider switches. Chapter 15 details the ATM Forum's Private Network-Network Interface (PNNI) specification, as well as the interface between carrier networks, the ATM Internetwork Interface (AINI). In Chapter 15, we will also briefly discuss the older ATM Forum Public Broadband Inter-carrier Interface (B-ICI) that AINI essentially is replacing.

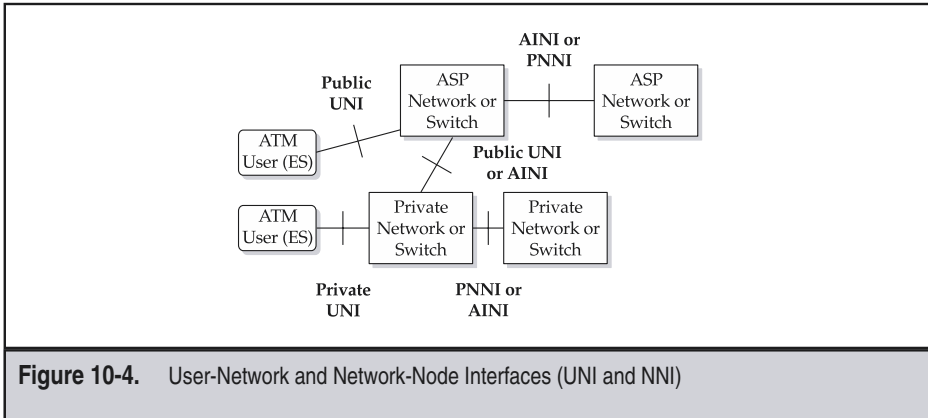


Figure 10-4. User-Network and Network-Node Interfaces (UNI and NNI)

ATM as Integrated Access

Virtually all major service providers provide some form of native public ATM service, enabling users to capitalize on a basic advantage of ATM—integrated physical and service access that promises to reduce cost. Much of native ATM service is provided to users that still demand higher rates of access speed, typically at DS-3 rates, while native Frame Relay is the popular choice for the intermediate speeds from 64 Kbps up to 20 Mbps with high-speed serial interfaces (HSSI). The ATM Forum’s Frame-based UNI (FUNI) and the Circuit Emulation Services (CES) specifications also define operation of ATM at 64 Kbps speeds. The standard support within Frame Relay, ATM, and PPP for DS1/E1 inverse multiplexing has extended Frame Relay and ATM’s benefits to users that currently employ separate TDM-based networks, as well as providing an economical network side protocol to access the local and wide area networks. The available choices don’t end there: native Ethernet user and network interfaces combined with Ethernet switches can also be utilized in this space.

Figure 10-5 illustrates one option for how ATM can provide integrated access for voice, data, and video applications to a wide range of network services over a single access line. ATM delivers equipment, bandwidth, and operational savings when supporting all data, voice, and video requirements over a *single* access line. As shown in the figure, the configuration involves an ATM-based, multiservice access device connecting voice, video, and data applications over a single access line to a multiservice edge switch in a service provider network. Multiple services sharing a single physical access circuit may allow for savings in network interface equipment, eliminating the need for multiple local loops and also reducing wide area network service costs. When used for integrated access, the provider’s edge switch may perform circuit emulation to split off the TDM traffic destined for the telephone and private-line networks, as detailed in Chapter 16. Simultaneously, customer devices access the Internet, and a public Frame Relay or ATM

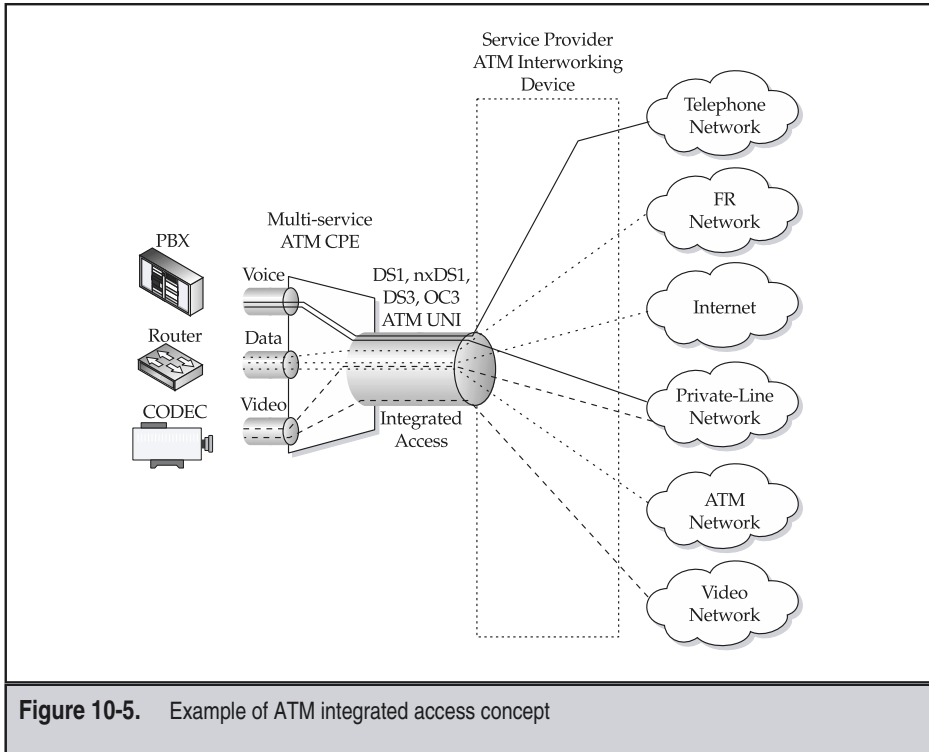


Figure 10-5. Example of ATM integrated access concept

data service, using protocol interworking as defined in Chapters 17 through 19. Furthermore, video applications may access video content networks, or interwork with legacy private line-based video services as indicated in the figure.

Of course, other combinations of integrated access connections than those shown in Figure 10-5 between customer applications or devices with public or private network services are also possible. Note that in this example we split out each service to a service-specific core network. This is typically a situation where a deployed base of service networks offers the actual customer service point. This is a crucial point that may not be immediately obvious in this scenario: the ATM integrated access is here used only as a replacement for multiple private-line connections to the customer premises. One or all of the services may be offered at the access side as well; and in Part 4, we will touch on the new distributed solutions that are being developed in this area to support both voice and data services with media gateway architectures.

ATM as an End-to-End Service

ATM handles both connection-oriented traffic (either directly [cell-based] or through adaptation layers) and connectionless traffic through the use of servers and adaptation layers. ATM virtual connections operate according to one of the following service categories:

- ▼ Constant Bit Rate (CBR)
- Variable Bit Rate (VBR), in either real-time (rt) or non-real-time (nrt) modes
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)
- ▲ Guaranteed Frame Rate (GFR)

These service categories support a wide range of applications with ATM virtual connections that operate according to a traffic contract and a specific service category that guarantees a particular Quality of Service (QoS), as discussed in Part 5. In this way, it supports applications requiring different delay and loss performance. ATM handles connection-oriented traffic directly or through adaptation layers, as detailed in Chapter 12. ATM provides either permanent or switched virtual connections (PVCs or SVCs) end to end. All cells are then transferred, in sequence, over this virtual connection. ATM standardizes on one network architecture for multiplexing and switching and operates over a multiplicity of physical layers. Thus, the synergistic vision of ATM is that of a network constructed using ATM and ATM adaptation layers (AALs) switching and multiplexing principles that supports a wide range of services; we will look at those services in Part 4.

ATM as a Scalable Infrastructure

ATM technology still has advantages over technologies such as IP, FR, or SMDS in a network infrastructure. ATM-based architectures currently offer the most mature integrated platform for voice, video, and data. ATM also can provide a highly scalable infrastructure, from the campus environment to the central office. Scalability occurs along the dimensions of interface speed, port density, switch size, network size, multiple application support, and addressing.

ATM also provides bandwidth granularity and flexibility in designing network topologies. As an illustration of this fact, consider a private-line-based network with multiple sites. If full-mesh connectivity is required, each site would require multiple access ports, local loops, and private-line circuits. On the other hand, a network using ATM-capable CPE devices, as shown in Figure 10-6, can share a single physical access port to a public ATM network. A logical virtual connection (VC) connects ATM devices at sites A, B, C, D, and E. Thus, each site saves the cost of access ports, local loops, and dedicated private-line circuits.

Depending upon the service provided by the public ATM network, each VC could burst up to its full line rate. If all VCs have this capability, then the service provider can statistically oversubscribe the VCs when multiplexing together the traffic of many

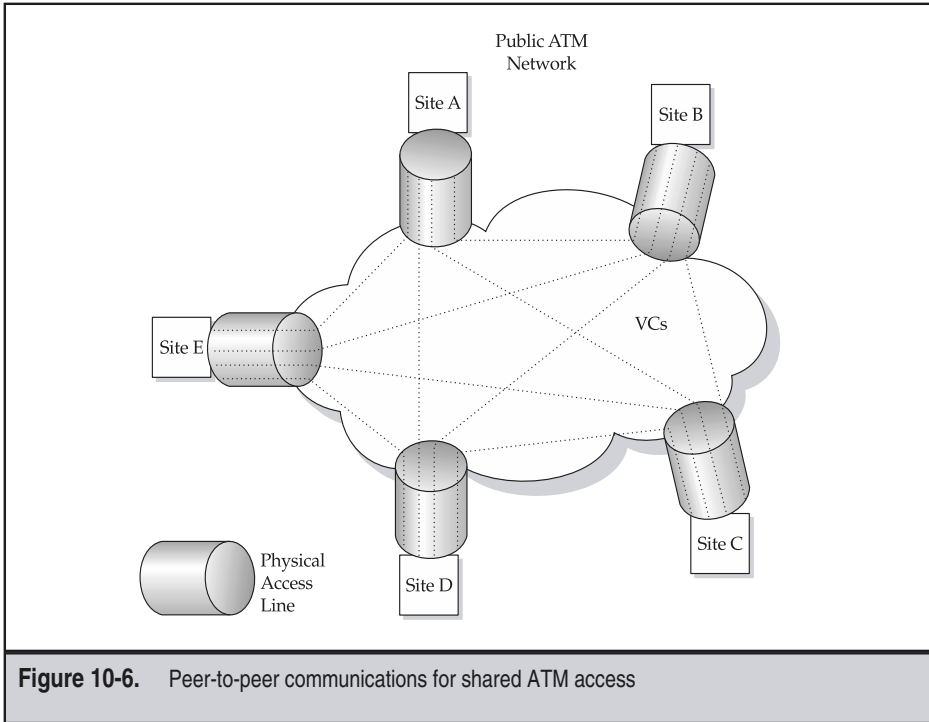


Figure 10-6. Peer-to-peer communications for shared ATM access

customers on shared facilities, since the end-user transmission rates are constrained by the access line speeds. For example, if the access port is a DS3, and the VCs connecting site A to sites B and C are both running at 20 Mbps, then the VCs to sites C and D must be idle. Public network switches use traffic control to partition bandwidth and buffer resources between customers to ensure guaranteed QoS and fair service during periods of congestion. Typically, carrier pricing for logical ATM connections in the Variable Bit Rate (VBR) service category runs less than for the equivalent dedicated-bandwidth private-line service. On the other hand, the price for a public ATM Constant Bit Rate (CBR) service should be about the same as that of an equivalent private-line service. The advantage to users with ATM over private lines in this case is that ATM CBR service has much finer bandwidth granularity than the rigid TDM hierarchy, as well as the fact that the same access line can support both CBR and VBR services simultaneously.

ATM architecture also enables dynamic and flexible adds, moves, and changes to the network topology. Once a site has a port and an access line connecting to the public ATM network, adding VCs to new remote sites is much easier than ordering and installing a new dedicated private-line circuit. Typically, carriers provision new logical ATM connections within minutes to no more than a few days. On the other hand, new private-line connections often take several weeks to provision and may require coordination between multiple service providers.

ORIGINS OF MPLS: REINVENTING IP OVER ATM

Beginning in the middle 1990s, Internet service providers constructed IP backbones using high-end enterprise routers interconnected via a network of ATM switches that provided full-mesh connectivity to avoid making multiple hops through (what were at the time) expensive router ports. This approach provided the initial infrastructure for the public Internet; but as the Internet experienced tremendous growth, the IP overlay networks began experiencing limitations in both the packet forwarding speeds and further network scaling. The full mesh of ATM connectivity was not absolutely necessary but was an economical way to construct IP backbone networks in the mid-1990s because router ports were significantly more expensive than switch ports. As the cost of router ports declined, ATM VCs could be built only between selected routers to provide a hierarchical, or a partial-mesh, topology to perform traffic engineering for only portions of the network, and also to reduce router adjacencies. These hybrid IP routing and IP over ATM traffic engineering overlay networks became complicated to manage because another important aspect of the ATM VC was collection of node-to-node traffic matrix data.

The development of MPLS targeted solutions to address the issues that arose from the IP over ATM ISP backbone experience. MPLS is a particular type of label switching specifically designed for connectionless networks, and a router in this context is defined as a label switching router (LSR). MPLS is also developing the traffic engineering and routing control benefits that ATM already possesses, although with more efficient support of native IP traffic by avoiding ATM cell overhead. The improvement in performance of an LSR over a hybrid IP/ATM architecture derives from several factors. First, frame-based MPLS can achieve a 10 to 15 percent improvement in link-level utilization over ATM due to reduced overhead, as analyzed in Part 8. Second, an IP router network overlaid on an ATM network that provides a full mesh of $N(N-1)/2$ PVCs makes every router appear to be only one hop away from every other router, as illustrated in Figure 10-7. This increases the load on routing processors due to the operation of the flooding algorithm, which inherently limits the scalability of an IP over ATM overlay network. Additionally, although ATM provides good traffic engineering support for the full mesh of virtual connections

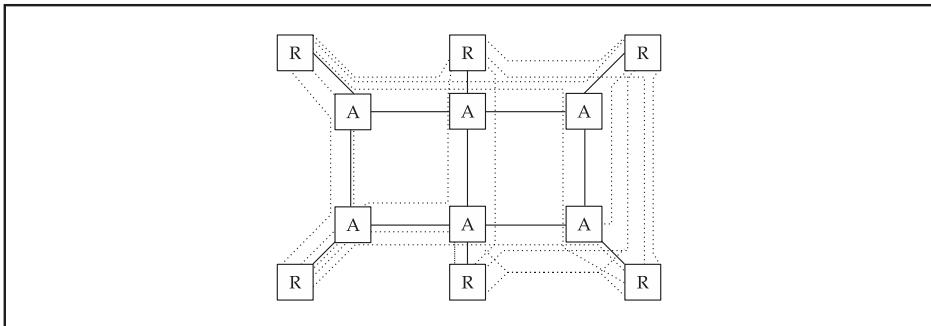


Figure 10-7. Hybrid network of IP routers connected via ATM switches

between the routers, the administration of this virtual mesh is done separately from the process of IP routing, a mode of operation sometimes called “ships in the night,” since there is no automatic coordination. On the other hand, an LSR allows label switching and routing decisions to be made in a coordinated fashion, instead of operating two independent control protocols. However, issues exist with this approach as well. MPLS merges some of the operation of the routing protocols, such as IS-IS and OSPF, with the IP control plane; whereas in an IP over ATM network, these protocols operate independent of each other. In the past, the IP protocol remained separated from the underlying infrastructure, and this enabled the famous “IP over anything” architecture. MPLS ties IP to its infrastructure, and this has raised some concerns that relying on MPLS too much may make IP less flexible if the assumption is made that IP must run over MPLS.

So where did the LSR and MPLS concepts originate? There exists today a multitude of “MPLS-like” implementations. Even after five years of work, MPLS is still an evolving set of standards; and as we shall see, there is still work to be completed before a full set of essential capabilities can be offered as standard implementations. This section traces some of the early suggestions to use ATM and Frame Relay label switching concepts for IP networks as well, but let’s first look at the background that motivates and shapes the development of MPLS solutions.

The LSR concept emerged on the scene, biting at the heels of LAN switching in the continual quest to improve networking price-performance. To forward packets and also support various services, a router usually supports a multitude of protocols. An ATM or Frame Relay switch, by contrast, forwards traffic according to relatively simple label-swapping procedures. At the time MPLS concepts emerged, routers had a higher relative price per port for the same performance in comparison with an ATM switch. A label edge router (LER) would operate at the edges of backbone networks with the goal of performing layer 3 routing and switching decisions only once, and not for every packet that transits the network to the same destination. Manufacturers were building better mousetraps in response to the tremendous growth in IP internetworking, and the general guiding principle was that practically all end-user applications would use IP in the future. Interestingly, a number of these manufacturers published at least an overview of their approach in IETF informational RFCs, in addition to issuing public proclamations that their approach would be an open, nonproprietary solution. The first company to break with the momentum of the standards bodies was a start-up company in Silicon Valley, Ipsilon Networks, whose IP Switching approach proposed placing IP over ATM on a strict protocol efficiency diet. A similar approach from Toshiba proposed cell switch routers for efficiently interconnecting classical IP over ATM and MPOA networks. Another entrant onto the scene was Cisco Systems’ Tag Switching architecture, which works with not only ATM but a number of legacy technologies as well. IBM’s Aggregate Route-Based IP Switching (ARIS) approach differs from the other approaches in offering a clever means of overlaying data transmission in the reverse direction over the spanning tree rooted in each node as determined by most routing protocols.

While making faster and cheaper routers was one motivation for changing the IP forwarding paradigms, there were more compelling reasons to look at new solutions. Some

of these we have already mentioned. Scalability problems arise as indicated in Figure 10-7, with the PVC mesh causing Interior Gateway Protocol (IGP) stress, as well as the ATM additional overhead. At the time, there were also limitations in the SAR interfaces of ATM switches. To process packets in and out of cells, an SAR process has to operate at the interface speeds; and as routers added higher-rate interfaces like OC-48 and eventually OC-192, the ATM switch development lagged behind. Routers could utilize Packet over SONET/SDH interfaces, and MPLS could replace the traffic-engineering capabilities provided by ATM. Today, interface issues like these are not relevant, as ATM interfaces operate at these speeds as well. The most powerful reason to develop MPLS is related to the need to support new services on IP networks, including real-time services.

The IETF formed a Multiprotocol Label Switching (MPLS) working group to sort this all out and come up with a common standard. For a detailed history of the IETF MPLS working group efforts, see References [Gray 01]. The following sections present a brief overview of each of the approaches that were the ancestors of MPLS.

Ipsilon's IP Switching

Beginning in 1995, Ipsilon Networks introduced a fresh idea into the industry. The company asserted that ATM had already lost the battle for the desktop to Fast Ethernet, that LANE did little more than LAN switching, and that the software burden of complex ATM protocols made proposed IP over ATM implementations cost-prohibitive. Furthermore, their publications questioned the direction of the ATM Forum and the IETF LANE, NHRP, and MPOA approaches for implementing IP over ATM. They pointed out duplication of function, scaling problems, and difficulties in multicast implementations. The answer to all of these problems was a simplified form of IP over ATM that they called IP switching.

Ipsilon published the key aspects of their protocol in Internet RFCs 1553, 1554, and 1587. This bold move made the aspects of the protocol open to all manufacturers: the algorithms were available not only to hosts and routers that would use Ipsilon's devices, but to competitors as well. The company even made source code available to the research community free of charge.

Basically, Ipsilon's approach classified traffic into either short- or long-lived flows. The new components of the IP switching approach applied only to the longer duration flows, such as FTP, long Telnet sessions, HTTP, and extended Web multimedia sessions. IP switching handled short-lived, interactive traffic, such as DNS, e-mail, and SNMP, in exactly the way IP routers handle it today. Therefore, in order for IP switching to improve performance, most of the total traffic must be in long-lived flows. If most of the traffic is in short-lived flows, then the performance of IP switching is no better than that of routers. A number of studies published by Ipsilon of corporate networks indicate that although a high percentage of flows are of short duration, these short-lived flows carry a small percentage of the packet traffic. Indeed, these studies report that the small number of long-lived flows carry most of the packet traffic in corporate networks. Other studies reported that optimizing 20 percent of the flows on a public Internet backbone optimized

50 percent of the packets traversing the backbone. These studies reinforce the common wisdom that a small portion of the overall flows make up most of the traffic.

As illustrated in Figure 10-8, the principal contribution that the Ipsilon approach made to MPLS was that a control protocol other than that based upon ITU-T standards could be used to control an ATM switch. An IP Switch Controller would provide this functionality via the Ipsilon's Flow Management Protocol (IFMP) as specified in IETF RFCs 1953 and 1954. At the time, this was a radical concept and one that allowed engineers to think outside the box in the early stages of MPLS standards development. The IP Switch Controller also implements a Generic Switch Management Protocol (GSMP) as specified in RFC 1987 to make and break ATM VCC connections through interaction with the ATM switch's controller. The IFMP protocol has not been widely utilized. For further details on these protocols, see References [McDysan 97] and [Davie 00]. On the other hand, the GSMP protocol has been extended by the IETF to handle the management and control of not only ATM switches, but Frame Relay and MPLS switches as well. Additionally, current work on the GSMP protocol is focused on control of optical switches and TDM (SONET/SDH) multiplexing equipment.

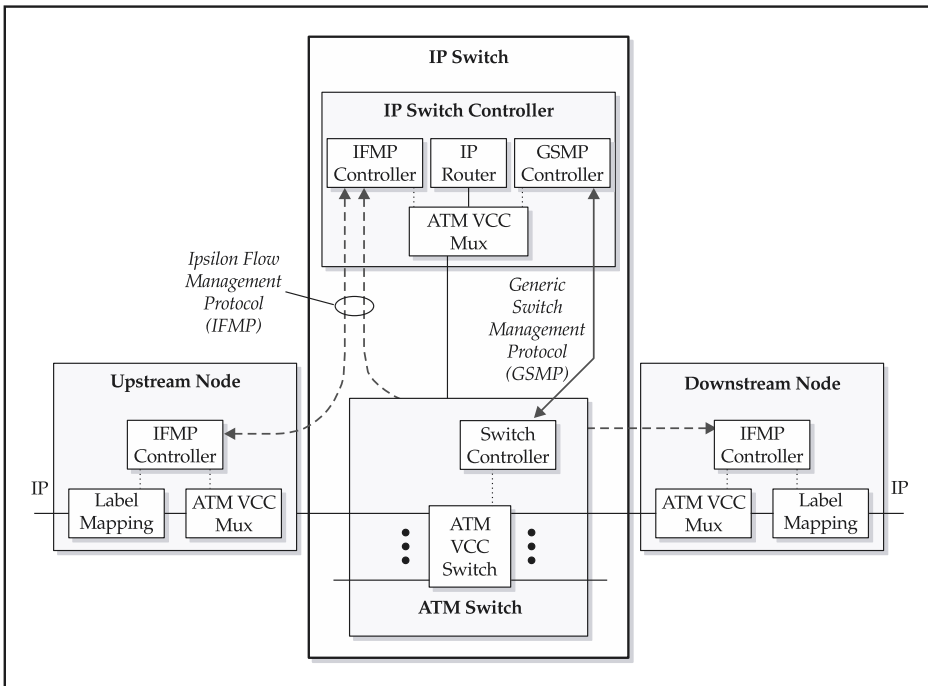


Figure 10-8. Ipsilon's IP switching architecture

Toshiba's Cell Switching Router (CSR)

RFC 2098 describes another vendor-proprietary proposal for handling IP over ATM networks. A cell switch router (CSR) has ATM cell-switching capabilities in addition to conventional IP datagram routing and forwarding, as illustrated in Figure 10-9. Note that this architecture is very similar to Ipsilon's IP Switch at this functional block diagram level, but operating with different control protocols.

The routing function in the CSR normally forwards IP datagrams along hop-by-hop paths via a routing function, exactly as in the IP switching approach. The routing function automatically recognizes long-lived flows and either assigns or establishes efficient shortcut ATM paths, similar to IP switching again. But CSR adds several new concepts: First, it proposes to handle more than the IP protocol. Second, it allows shortcut connections to be preconfigured or established via interaction with RSVP. CSR also proposes setting up shortcut routes that may bypass several routers. CSRs interact using a Flow Attribute Notification Protocol (FANP), as indicated in Figure 10-9. CSRs also implement standard IP routing protocols, the ATM Forum PNNI protocol, and ATM signaling.

Cisco's Tag Switching

Cisco announced its-tag switching architecture in September 1996. An informational RFC 2105 gives an overview of the architecture and protocols involved in tag switching.

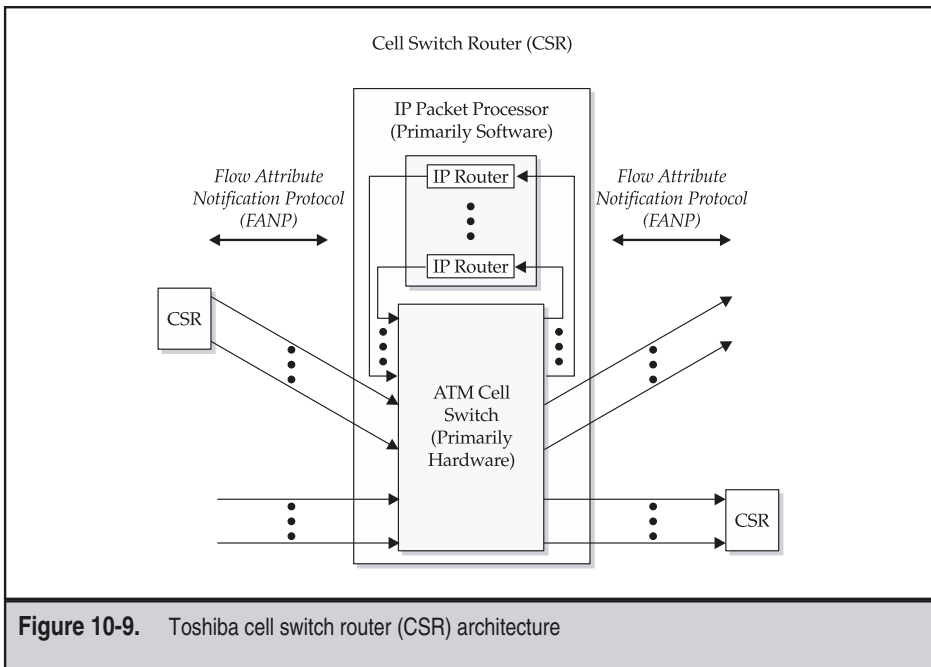


Figure 10-9. Toshiba cell switch router (CSR) architecture

Figure 10-10 illustrates the basic components and interfaces of Cisco's tag switching architecture in an ATM network environment. Tag edge routers at the boundaries of an ATM network provide network layer services and apply tags to packets. Tag switches/routers at the core of the network switch tagged packets or cells using tags determined via information piggybacked onto standard routing protocols, or via Cisco's Tag Distribution Protocol (TDP). Tag switches/routers and tag edge routers implement standard network layer routing protocols, such as OSPF and BGP, as shown in the figure. Additionally, they implement TDP in conjunction with standard network layer routing protocols to distribute tag information.

Tag switching is a high-performance packet-forwarding technique based on the concept of *label swapping*. A *label* is a generic name for a header. Swapping labels at intermediate nodes leads to an end-to-end connection. Since ATM VCC switching directly implements a special case of the general label swapping using the VPI/VCI fields in the cell header, the switches/routers know whether to switch cells or to assemble the cells and route the resulting packets based on information derived from TDP. The IETF adopted many of these concepts, replacing the word "tag" with "label," resulting in the MPLS standards terminology of label edge router (LER), label switching router (LSR), and Label Distribution Protocol (LDP) [Davie 00].

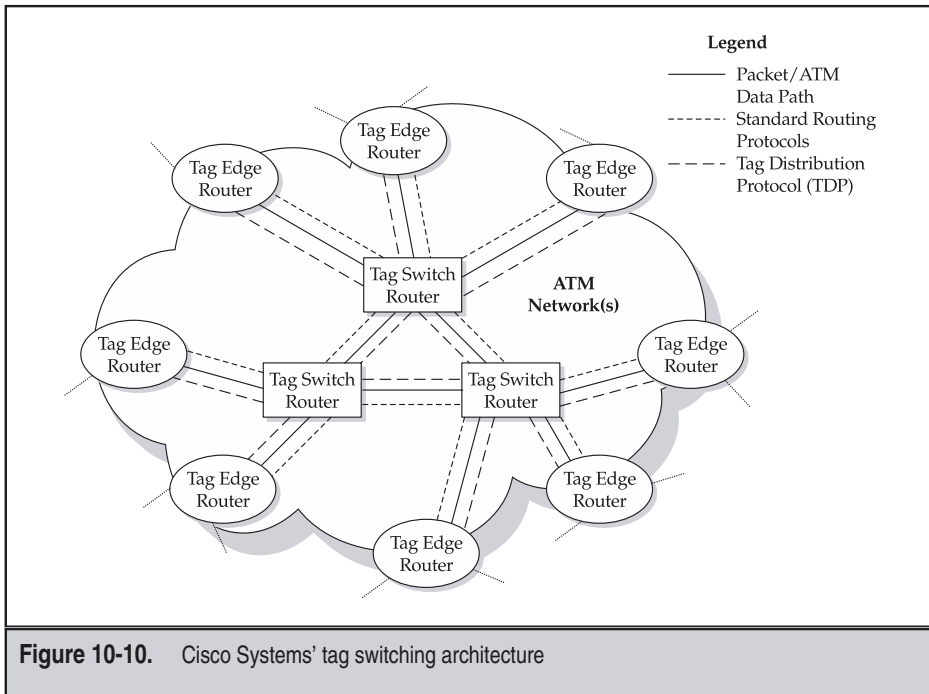


Figure 10-10. Cisco Systems' tag switching architecture

Tag edge routers run standard routing protocols and populate their next-hop tables with the most desirable routes in terms of the routing criteria, such as the destination address prefix. Tag routers and switches utilize these next-hop tables and distribute VCC tag information via TDP. Tag edge routers examine network layer headers of received packets, perform network services (e.g., filtering), select a next-hop route, and then apply a tag. In other words, they perform traditional routing. For example, a tag edge router may apply a VCC tag such that several intermediate tag switches/routers may switch the cells directly through to the destination tag edge router without performing any routing! Of course, any such tag-switched paths must first be established before this operation can occur. Thus, this design replaces the complex processing of each packet header with the simpler processing of only the label. At the destination edge router, the tag is removed—that is, the packet is reassembled and forwarded to the destination.

Figure 10-11 illustrates the operation of tag switching for a simple network. Tag edge router R1 on the left-hand side serves a Class C IP address subnet designated by C.B.A.*, while tag edge router R2 on the far right-hand side of the figure serves IP subnet A.B.C.*. As shown in the figure, each device in a tag-switched network has a tag forwarding information base (TFIB), which contains the incoming tag (In Tag), the address prefix obtained from the internetworking routing protocol, an outgoing tag (Out Tag), and an outgoing

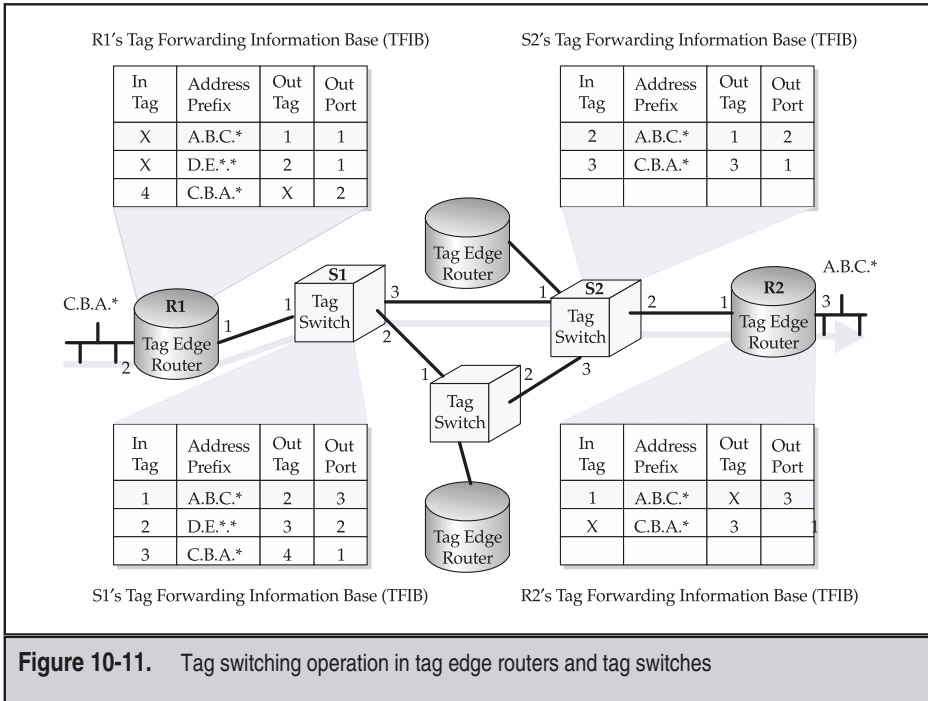


Figure 10-11. Tag switching operation in tag edge routers and tag switches

interface number (Out Port). Let's trace a packet destined for IP address A.B.C.55 from port 2 on the LAN connected to tag edge router R1 through the tag-switched network to the destination port 3 on router R2. The shaded entry in R1's TFIB instructs the tag edge router to prefix packets with IP address prefix A.B.C.* with an outgoing tag of 1, and transmits the resulting tagged packet on port 1. Tag switch S1 receives this packet and consults its TFIB. The shaded entry indicates that S1 changes the tag to a value of 2 and transmits the result on port 3. Tag switch S2 receives the packet and sees from the shaded entry in its TFIB that it should swap the tag to a value of 1 and transmit the tagged packet on port 2. Note that tag switches S1 and S2 only examined the tag values to make a switching decision. Finally, tag edge router R2 receives the packet with tag value 1, removes the tag (indicated by an X in the TFIB in Figure 10-11), and transmits it onto the destination LAN with IP addresses of the form A.B.C.*.

The TFIBs in Figure 10-11 also show the corresponding swapping of tag labels at each node in the reverse direction from tag edge router R2 to the IP subnet C.B.A.* on tag edge router R1. Note that, in this example, a tag can never be reused in the input tag or outgoing tag columns in any device's TFIB; otherwise, an ambiguity would result. In this example, tags have only local significance (i.e., per-switch); thus, a tag can be reused in an outgoing tag column as long as tags are not duplicates of values used by downstream tag switches. Since this destination-based forwarding approach is topology driven, rather than traffic driven, tag switching does not require high call setup rates, nor does it depend on the longevity of flows to achieve increased throughput, as the Ipsilon or Toshiba CSR approaches do. Furthermore, tag switching has the potential to make ATM switches peers of other routers, since they participate in standard network layer routing protocols with edge routers.

IBM's Aggregate Route-Based IP Switching (ARIS)

IBM's Aggregate Route-Based IP Switching (ARIS) defines a route as a multicast distribution tree rooted at the egress point, traversed in reverse [Feldman 97]. The egress point is specified by a unique identifier, for example, an IP address prefix, an egress router IP address, or a multicast source and group address pair. Recall from Chapter 9 how the result of the Dijkstra algorithm computation of the least-cost paths from a root node to every other node in the network results in a minimum weight-spanning tree. ARIS effectively uses this tree to determine the forwarding path as the reverse direction along such a minimum spanning tree, as shown in Figure 10-12. The thick solid line originating from the egress point is the root of the minimum spanning tree. The arrows show the data-forwarding path in the opposite direction, toward the root of the spanning tree (i.e., the egress router).

Note that flows from the leaves of the spanning tree back toward the root merge at several points. An important contribution that ARIS made to MPLS standards was this concept of label merging. ARIS also defined operation over either frame-switched networks or cell-switched networks. For use over cell-switched networks, ARIS requires ATM switches capable of VC merging in order to support larger networks. An ATM switch capable of VC merging transmits all cells from an individual AAL5 PDU received

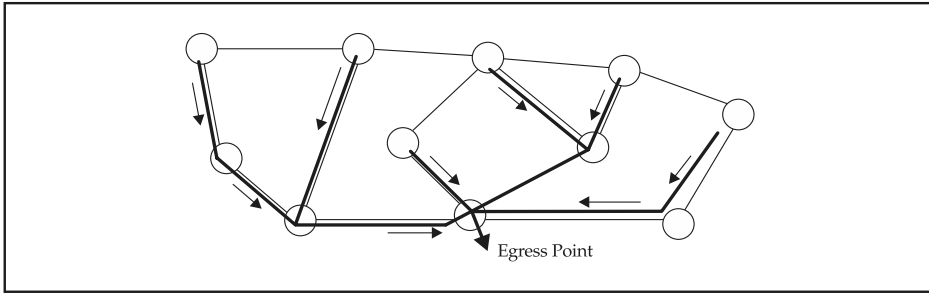


Figure 10-12. Example of IBM's Aggregate Route-Based IP Switching (ARIS)

on a particular input branch onto the merged VCC prior to transmitting cells received on another branch from another AAL5 PDU, as illustrated in Figure 10-13. Thus, VC merging allows ARIS integrated switch routers (ISRs) to group cells from individual AAL5 PDUs from different inputs and switch them onto a shared VCC on the path back to a common egress identifier. Note that the VC merge function need not reassemble the entire AAL5 PDU, but need only ensure that the sequence of cells belong to one AAL5 PDU remains intact, as shown in the figure. The standards call this configuration of forwarding in the reverse direction *multipoint-to-point*.

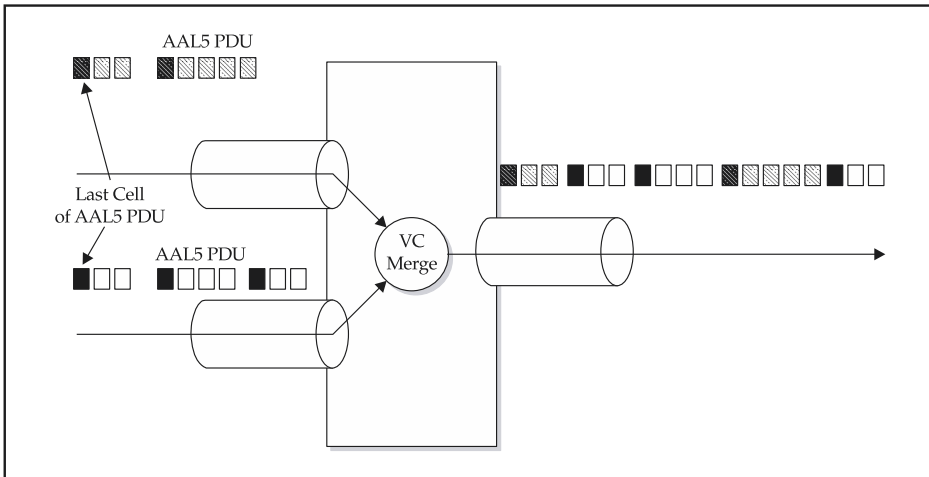


Figure 10-13. VC merging

A key consequence of this design is that every egress router in an N-node ARIS network has only $N - 1$ point-to-multipoint connections! Thus, the ARIS design significantly improves network scalability when compared with the $N(N - 1)/2$ connection required in a full-mesh network. In the absence of a VC-merging capability, ARIS defines the means to merge virtual circuits through the use of virtual paths (VPs); however, the number of bits allocated to the VPI, either 8 at the UNI or 12 at the NNI, limits the size of a network supported by this alternative. This concept, called VP merge, was adopted in MPLS. The other decision that impacts the scalability of the ARIS approach is selection of the egress identifier. Choice of an IP address prefix results in a larger number of egress end points, and hence a less scalable design. Choosing a router IP address makes the network scale in terms of the number of routers. Although not a solution to a large IP backbone in itself, ARIS is well suited to the sizes of many corporate IP networks.

ARIS defined an integrated switch router (ISR) supporting standard IP routing protocols. It also implements an IP forwarding table that includes a reference to the multipoint-to-point switched path determined by an explicitly specified egress point on the IP network. Switched paths terminate in neighboring ISRs or may traverse a number of ISRs along the best path to the egress ISR. By the nature of its design, ARIS guarantees that switched paths are loop-free. Like IP tag switches, ISRs forward datagrams at hardware speeds. The ARIS specification defined a protocol and message exchange emanating at the egress ISR to broadcast its existence and eventually establish the reverse merged paths. ARIS switched paths are soft state, maintained only as long as ARIS keep-alive messages are exchanged. A stable ARIS network has N multipoint-to-point trees rooted in each egress node. Note that only control traffic occurs in the point-to-multipoint direction; all data traffic transfer occurs in the multipoint-to-point direction.

Early IETF Multiprotocol Label Switching (MPLS)

In response to the unsolicited proposals for building scalable Internet backbones, the IETF established the Multiprotocol Label Switching (MPLS) group in 1997 to come up with one common specification. The initial work produced RFC 2702 that summarizes the requirements for traffic engineering on IP networks over MPLS. An architecture document [RFC 3031] addresses the MPLS features that support these requirements; it combines a number of the functional requirements and design decisions from the proprietary approaches described previously. The initial design discussions for MPLS focused on some basic features, including the following:

- ▼ Use of a short, fixed-length layer 2 switching label to achieve lower-cost and higher-performance packet forwarding than is attainable through traditional routing techniques
- Scalability on the order of N streams for best effort traffic, as a means to scale in support of rapid Internet growth (drawing on IBM's ARIS work)
- Mandatory support for forwarding of both unicast and multicast
- Mandatory support for RSVP and the IETF integrated services model

- Mandatory support for topology-driven protocols, such as those defined by Cisco's tag switching
- Retention of compatibility with existing and legacy IP routing protocols, operations, administration, and maintenance facilities, as well as coexistence with devices not capable of supporting MPLS
- Prevention, or rapid detection and removal, of routing loops
- ▲ Mandatory independence of any specific data link technologies. Specific optimizations for particular data link networks, such as Frame Relay and ATM as well as new data link technologies optimized for MPLS, may be considered

Table 10-1 summarizes the advantages of an MPLS solution over the prior native router or IP over ATM overlay method. Note that MPLS does not claim to support QoS better than ATM networks can. Furthermore, MPLS ended up using explicit routes to

Traditional IP Router Network	Routers Overlaid on ATM Network
Forwarding capacity: Short, fixed MPLS layer 2 labels are more efficient to process than longer, variable-length layer 3 headers.	Scalability: MPLS avoids the $N(N - 1)/2$ router adjacencies required in a full traffic-engineered mesh network.
Explicit routing: MPLS is more efficient because the entire path is transferred only once, not with every packet.	Common operation over multiple media: MPLS specifications operate over frame- or cell-based link layer networks.
Traffic engineering: MPLS is more efficient and flexible than adjusting administrative routing weights in traditional routing protocols.	Common route management and control: ATM network connections must be administered in close coordination with IP routing to achieve efficient operation.
QoS routing: MPLS supports this by setting up explicit routes.	Simplicity: MPLS eliminates the need for shortcut-based routing as used in NHRP.
Functional partitioning: MPLS allows LSRs to perform transit forwarding.	
Reusable paradigm: MPLS and ATM can operate on the same devices.	

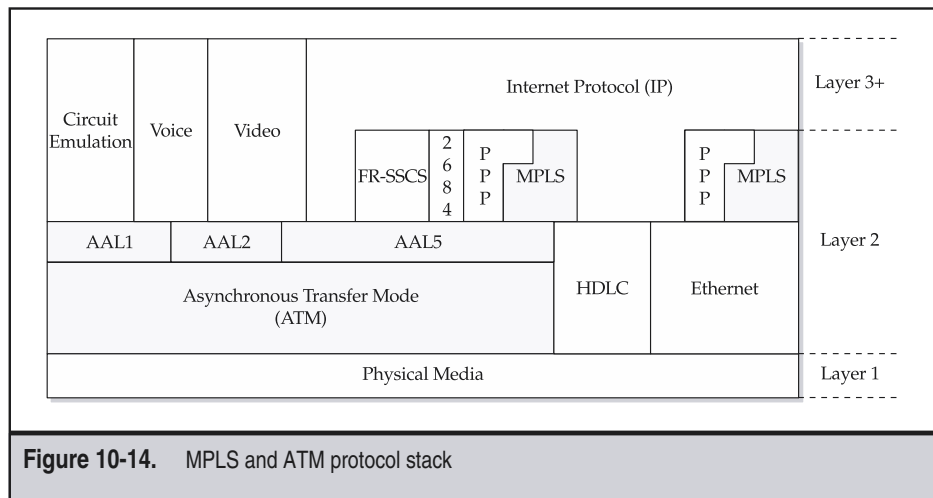
Table 10-1. Advantages of MPLS Over Traditional IP and IP/ATM Internetworks

support specific bandwidth and QoS guarantees in a manner that is very similar to ATM PNNI. In essence, MPLS is a different means for setting up connections in a layer 2 network. Much energy was placed on the topic of reinventing IP over ATM, and the result has been a set of MPLS standards developed by the IETF. We describe the functions of MPLS protocols in parallel with comparable ATM functions so that you can observe the influence of the ATM experience, as well as appreciate the innovations developed as part of the ever-expanding scope of MPLS-based solutions.

INTRODUCTION TO MPLS

So far, we have treated ATM and MPLS without really dealing with the fact that, strictly speaking, MPLS is a label-switching paradigm optimized for IP that does not encompass the range of services that ATM supports. So far in this chapter, we have discussed the many faces of ATM, as an infrastructure, a switching protocol, and an end-to-end service. Currently, MPLS supports only a part of this range of capabilities, instead focusing on a broad set of other IP-oriented functions. However, the IETF and the MPLS Forum are actively extending the services supported, and Parts 4 and 8 summarizes some of the directions that these efforts could take.

Figure 10-14 illustrates the layered relationship between IP and higher-layer protocols, and where MPLS fits into this overall layered model. Starting at the bottom of the figure, all communication protocols are supported by some layer 1 physical transmission medium. Higher-layer protocols access the cells or frames through the conventional OSI method of numbered protocol layering. Note that some services, such as circuit emulation in support of private lines, voice, and video, do not necessarily conform to the



OSI-layered model. When we look at support for multiple services over MPLS, we will see that the OSI model of successively numbered protocol layers is no longer applicable. We will see layer 1 SONET/SDH and layer 2 ATM and Frame Relay stacked upon MPLS layer 2, or even IP layer 3. Continuing to the right in Figure 10-14, observe that MPLS operates over AAL5, HDLC, and Ethernet. IP runs over almost anything, although the sometimes-cited operation of “IP over barbed wire” may be a bit exaggerated, and the figure shows that IP and MPLS can operate directly over HDLC or Ethernet as well, often using the Point-to-Point protocol introduced in Chapter 8.

MPLS does not present a user interface that offers an end-to-end service. We have, however, indicated that the kind of functionality that ATM offers in the way of traffic engineering and infrastructure support of multiple services will also be possible within the MPLS framework. The remainder of this section takes an introductory look at the main features that MPLS offers in the following areas:

- ▼ Traffic engineering for IP networks
- Network-based IP VPN operating over MPLS tunnels
- ▲ MPLS tunneling in support of multiple services

Traffic Engineering of IP Networks

Since the MPLS work draws on several important concepts from ATM, the same support is provided in an integrated MPLS/IP network that was available in an IP/ATM network. The MPLS label and the ATM cell header both have a similar semantic interpretation, with the exception of time to live (TTL) processing in MPLS. Figure 10-15 depicts the same network shown for IP overlaid over ATM earlier, except with label switching routers (LSRs) at each site instead of a separate router and ATM switch at each site. Observe that the routing protocol has fewer adjacencies, and hence will have less processing to perform on flooded topology update messages. In fact, the routing adjacencies and flooding are identical to the physical topology. A further advantage of this tighter integration is the potential to integrate the traffic engineering aspects more closely with those of network routing. As studied in Chapter 14, this is not a trivial problem, since an ideal routing

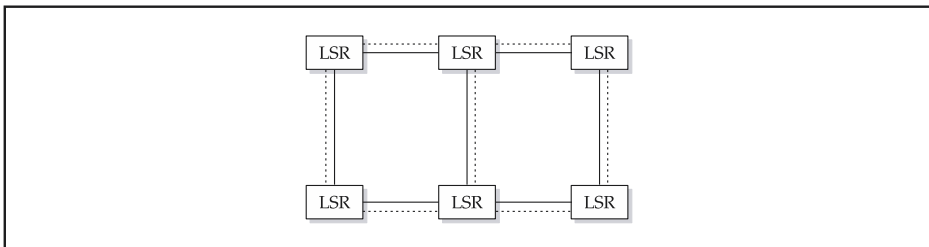


Figure 10-15. Network of MPLS label switching routers (LSRs)

algorithm must simultaneously meet constraint-based routing, traffic engineering, and restoration requirements.

Implementation of QoS and traffic management via constraint-based routing is a theoretically complex problem, for which an optimal solution is currently not known. Chapter 14 summarizes this deep subject. The implications of restoration in response to failures and reversion to a globally optimal network configuration make this topic at best an art form today. Since a standard in this area is not likely in the near future, to be able to perform constraint-based routing in a consistent manner is an important requirement.

Network-Based IP VPN using MPLS Tunneling

A virtual private network simulates the operation of a private wide area network over a shared public infrastructure. VPNs are not a new concept. Network-based voice switching and intelligence were used in the middle 1980s, followed by Frame Relay and ATM in the early 1990s to create virtual private networks [McDysan 00b]. IPsec was then used to create IP VPNs over the Internet beginning in the latter half of the 1990s. IP VPNs can be provided in several ways. A layer 3 VPN can be implemented by encapsulating layer 3 packets into IPsec, Generic Routing Encapsulation (GRE), or Layer 2 Tunneling Protocol (L2TP) tunnels over the Internet. MPLS supports what is referred to as a network-based VPN. Important requirements of VPNs are to provide user isolation, reliability, and flexibility in a scalable manner. The MPLS-based VPN solutions come in two flavors, the virtual router [RFC 2917] and the BGP/MPLS [RFC2547] models, as discussed further in Chapter 19. An important motivating factor for building IP-centric network-based VPNs is that a service provider can provide an integrated IP VPN solution to a customer in a more cost-effective, supportable manner. A customer will find that a network-based VPN has several benefits. First, peering with all other sites is no longer necessary, because each site has only a single peering relationship with the network nodes to which it is attached. Also, the service provider performs management of the VPN, making networking much simpler for a customer.

Multi-Service MPLS Tunneling

An area that is currently receiving a great deal of attention in the industry is to tunnel other services over an IP infrastructure using either MPLS or an extended L2TP. This would allow ATM, Frame Relay, Ethernet, SONET, and other protocols to use a common backbone in certain parts of a network. We first look at this in Part 4 and then examine the possible future directions in Part 8, since these protocols are still in a prestandard stage. However, some features are reasonably well defined. We will look at the encapsulation of various protocols into MPLS that is based on the IETF Pseudo-Wire Edge-to-Edge Emulation (PWE3) working group, as well as the operation of the ATM Forum's ATM/MPLS Network Interworking standard.

The concept of multiservice tunneling is simple. First, the tunneled protocol is indicated by an inner header, which is prefixed to the original protocol data unit to be transported (e.g., ATM cell, FR or Ethernet frame, or [set of] TDM time slots). Next, one or

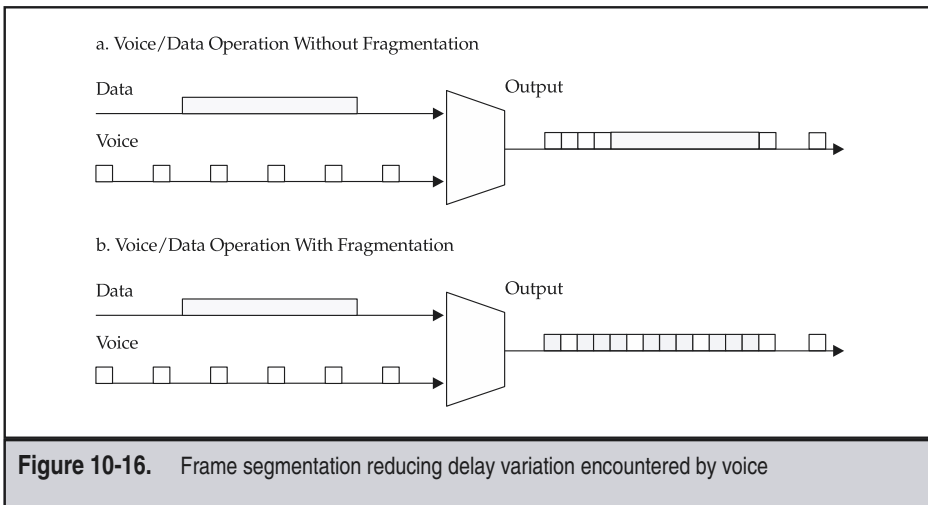
more outer headers (e.g., MPLS or IP) are added to the protocol-specific labeled content. The entire package is then conveyed by the tunnel network using either MPLS and/or IP to the destination, which reverses the previously described process. Although this process is simple in concept, some significant issues arise related to preserving the performance of the original protocol. For example, in support of protocols like SONET or SDH with stringent performance requirements, we will see that it is difficult for MPLS- or IP-based tunnels to avoid detrimental effects.

CONSIDERATIONS IN THE CHOICE OF CELLS VERSUS FRAMES

This section contains some material that highlights some trade-offs involved in the usage of cells or frames in terms of performance and engineering economics.

Effect of Link Speed on Packet Performance

A significant performance issue occurs with frame-based protocols on lower-speed links when a long frame gets ahead of a short, delay-sensitive frame [McDysan 00a]. A long, 1500-byte Ethernet packet getting ahead of a short voice packet can delay the latter by approximately 8 ms on a 1.5 Mbps link, as illustrated in Figure 10-16a. Interactive voice conversation becomes strained if the one-way delay is greater than 150 ms. What occurs when the delay becomes this large is that both people involved in the conversation can begin speaking before realizing that the other is already talking. The resulting collisions



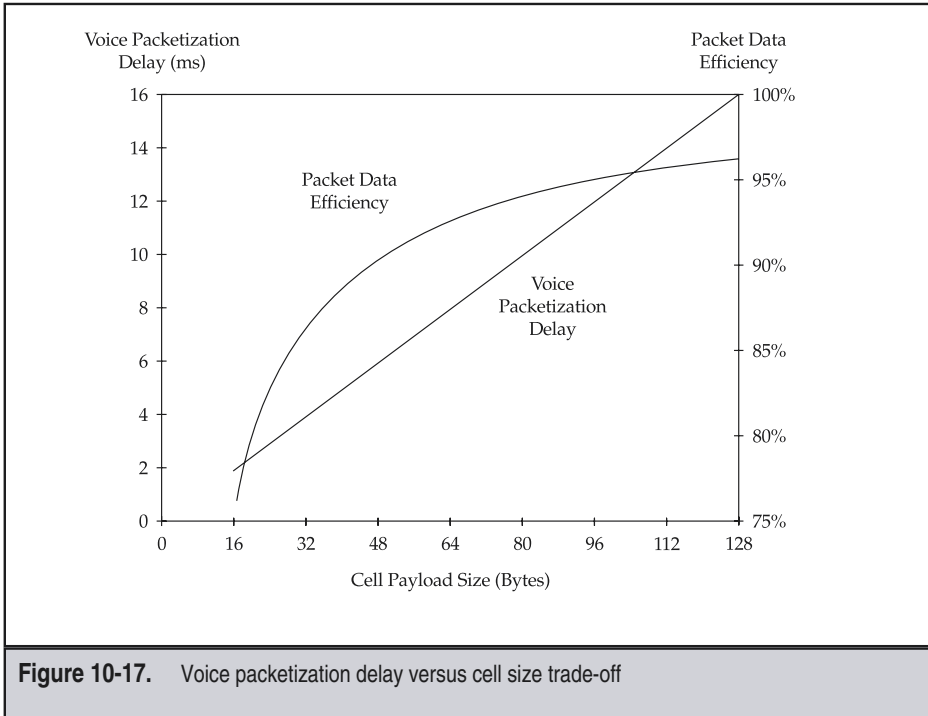
in conversation force repetition of sentences or phrases, greatly reducing the interactive nature of normal speech that we often take for granted. Therefore, packet voice and data cannot share the same link unless something is done to avoid the occurrence of long packets on low-speed links. Note that once the link speed is greater than 10 to 100 Mbps, this problem no longer is a significant issue [McDysan 02]. Therefore, in high-speed MPLS trunking applications, packet segmentation is not necessary.

The Internet Protocol defines a means for fragmenting packets; however, that method utilizes significant processor resources. Several standards solve this problem using what is called nonpreemptive prioritized queuing at the link layer, meaning that the transmission of the current frame is completed before the next frame can be sent. Some proprietary implementations implement preemptive prioritized queuing, meaning that the long frame can be interrupted to send the urgent, higher-priority frame. However, no standards have been adopted for this type of implementation. Frame Relay and other HDLC-based standards defined a simpler method where the user device and the network work together to fragment the large packets into shorter segments. A higher-priority packet, say, a voice packet, then has an opportunity to interrupt a longer data packet, as shown in Figure 10-16b. For example, if the fragment size is 100 bytes, then the maximum waiting time reduces to less than a millisecond on a 1.5 Mbps link. The Frame Relay Forum FRF.12 implementation agreement [FRF.12] specifies a protocol that breaks up long data frames into smaller segments and reassembles them at the destination. The FRF.12 specification defines procedures for performing fragmentation across an interface between a user and a network, or on an end-to-end basis. IETF RFC 2686 specifies a similar means to implement this function using the Point-to-Point Protocol (PPP) for use in IP networks. This avoids the problem of a long frame delaying shorter, urgent voice frames.

Rationale for the Choice of ATM Cell Size

When it came to deciding on a standard cell size in the CCITT, a debate raged between a 32-octet versus a 64-octet payload size for valid technical reasons. The final decision on the 48-byte payload size was actually a compromise between these two positions. The choice of the 5-octet header size was a separate trade-off between a 3-octet header and an 8-octet header, between increased function delivered by a larger header versus the improvement in efficiency of a smaller header.

The debate centered over the basic trade-off between packet data transport efficiency and voice packetization delay versus cell payload size, as illustrated in Figure 10-17. The figure shows packet data efficiency for a 5-octet cell header. Voice packetization delay is the amount of time required to fill the cell payload at a rate of 64 Kbps, that is, the delay waiting to fill the cell with digitized voice samples. Ideally, high efficiency and low packetization delay are both desirable but cannot be achieved simultaneously, as seen from the figure. Better efficiency occurs at large cell sizes at the expense of increased packetization delay. In order to carry voice over ATM and interwork with two-wire analog telephone sets, if the total round-trip delay exceeds 50 ms, then the network must employ echo cancellation. During this time, the echo cancellation delay objective was more stringently set at 15 ms. Hence, a cell size of 32 octets avoided the need for echo cancellation.



Looking at Figure 10-17, a cell payload size of 32 octets results in a best-case data packet efficiency (for a very long packet) of approximately 86 percent. Increasing the cell payload size to 64 octets would have increased the data efficiency to 93 percent. Thus, the ITU-T adopted the fixed-length 48-octet cell payload as a compromise between a long cell size for more efficient transfer of delay-insensitive data traffic (64 octets) and smaller cell sizes for delay-sensitive voice traffic (32 octets). However, as studied in Part 8, the actual efficiency of the 48-byte payload in support of the actual packet length distribution of IP traffic is only about 80 percent. A consequence of this compromise is that voice over ATM connections in a local geographic area require echo cancellation, whereas voice over TDM connections do not. Thus, voice over ATM in local geographic areas starts at an economic disadvantage with respect to traditional digitized voice over TDM. Chapter 17 explores the operation of voice over ATM and the role of echo cancellation in more depth.

Hardware Price-Performance Trade-offs

ATM and MPLS implementations addressed the performance problems of software-based routers by switching in hardware. ATM focused on making the data unit

a standardized fixed size, since hardware is able to handle fixed-length data blocks with less complexity than blocks of variable length. The reason for this is that with a fixed-length data block, the hardware does not need an additional counter to track boundaries of variable-length data blocks. ATM exposes a standard fixed-length data block, called a *cell*, used by a switching machine on external interfaces. Actually, most ATM switches use an internal cell that is larger than 53 bytes to ease implementation and provide additional internal switching functions like multicast and traffic control.

For a router or switch with frame-based interfaces, transforming between a variable-length frame on an external interface to and from the internal fixed-length data block is a relatively simple operation. Many hardware implementations that support variable-length frames on external interfaces actually employ a fixed-length data block for switching and routing within the machine. However, router technology evolved in parallel with that of switching, and now many routers are able to perform IP address-based forwarding at line rate in hardware. Therefore, there is currently little performance advantage of MPLS in terms of line rate forwarding, and the principal benefits are traffic engineering and support of other services, as described earlier.

REVIEW

This chapter introduced you to the foundational elements of ATM and MPLS. These include ATM's multifaceted nature, acting as an interface, a protocol, a technology, integrated access, a scalable infrastructure, and a service. MPLS does not have all of these facets, but primarily provides a technology and scalable infrastructure for traffic-engineering IP backbone networks, as well as an interface enabling other services. We summarized each of these aspects to set the stage for the rest of the book. The chapter then looked at the background and early development of MPLS protocols, and, finally, described the main applications for MPLS: traffic engineering, network-based VPNs, and support of multiple services using tunneling. We concluded with a discussion of topics where cells versus frames makes a difference in performance and economics.

CHAPTER 11



ATM and MPLS: Physical Layer and Label Switching Functions

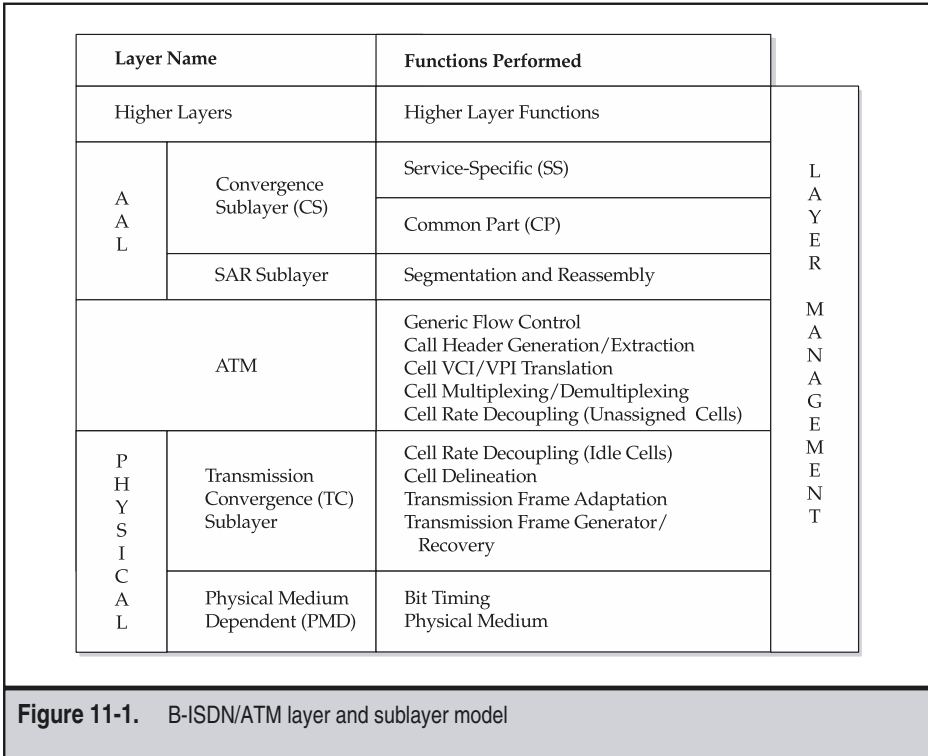
This chapter begins by exploring the foundation of the B-ISDN protocol stack: the physical and ATM layers. We start at the bottom with the physical (PHY) layer and then move to the ATM layer, which defines virtual connections and other functions. The ATM Adaptation Layer (AAL) provides support for higher-layer services such as signaling, circuit emulation, and Frame Relay, as defined in the next chapter. The description begins with the broad range of physical interfaces and media currently specified for transmission of ATM cells. Next, the discussion moves to definitions and concepts for the ATM layer, including the cell structure, meanings of the cell header fields, payload types, and generic functions used later in the book.

The chapter continues with the related MPLS functions. These are the formats of the generic MPLS label (called a “shim header”), as well as support for MPLS labels using Frame Relay and ATM “labels.” We then introduce other MPLS-specific terminology, concepts, and examples, primarily focusing on the standardized MPLS support for IP.

OVERVIEW OF PHYSICAL, ATM, AND AAL LAYER FUNCTIONS

This section looks at the next level of detail in the B-ISDN model of physical, ATM, and AAL layers from several points of view. This overview provides an outline for the ATM descriptions in this chapter, as well as the coverage of the AAL layer in Chapter 12. Unfolding the front and right sides of the B-ISDN protocol cube described in Chapter 10 yields the two-dimensional layered model shown in Figure 11-1, which lists the functions of the four B-ISDN/ATM layers along with the sublayer structure of the AAL and physical (PHY) layer, as defined by ITU-T Recommendation I.321.

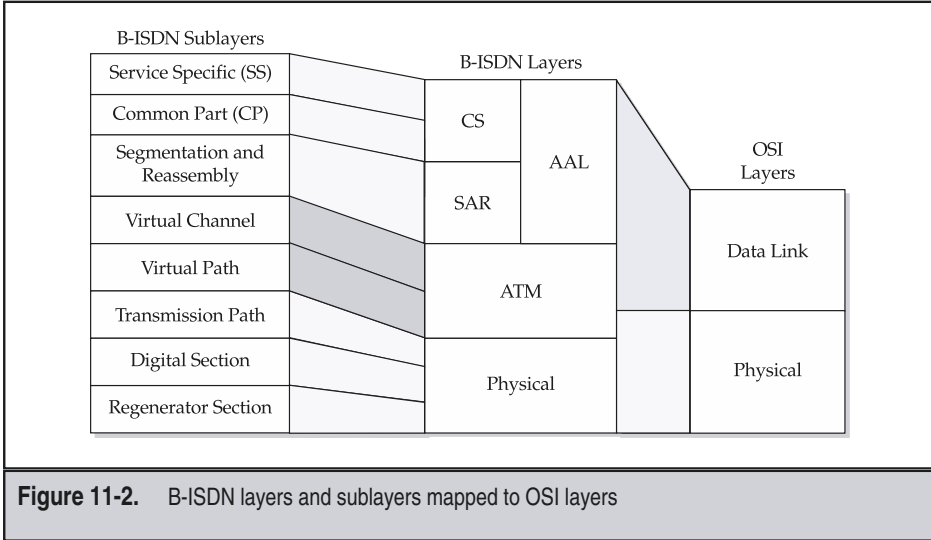
Starting from the bottom, the physical layer has two sublayers: Transmission Convergence (TC) and Physical Medium Dependent (PMD). The PMD sublayer interfaces with the actual electrical or optical transmission medium, detecting the signals, transferring bit timing, and passing the bit stream to and from the TC sublayer. The TC sublayer extracts and inserts ATM cells within either a Plesiochronous or Synchronous (PDH or SDH) Time Division Multiplexing (TDM) frame and passes these to and from the ATM layer, respectively. The ATM layer performs multiplexing, switching, and control actions based upon information in the ATM cell header and passes cells to, and accepts cells from, the AAL. The generic AAL has two sublayers: Segmentation and Reassembly (SAR) and Convergence Sublayer (CS). The CS is further broken down into Common Part (CP) and Service-Specific (SS) components. Not all AALs follow this model; for example, AAL2 does not have CS and SAR sublayers. Instead, AAL2 efficiently multiplexes short packets from multiple sources into a single cell. AALs pass Protocol Data Units (PDUs) to and accept PDUs from higher layers. These PDUs may be of either variable or fixed length. Chapter 12 details the AALs currently standardized for operation over the common ATM layer.



B-ISDN Protocol Layer Structure

The physical layer corresponds to layer 1 in the OSI model. Most experts concede that the ATM layer and AAL correspond to parts of OSI layer 2, but other experts assert that the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) fields of the ATM cell header have a network-wide connotation similar to OSI layer 3 [Tannenbaum 96]. Precise alignment with the conceptual OSI layers is not necessary: use any model that best suits your networking point of view. As we shall see, B-ISDN and ATM protocols and interfaces make extensive use of the concepts of layering and sublayering.

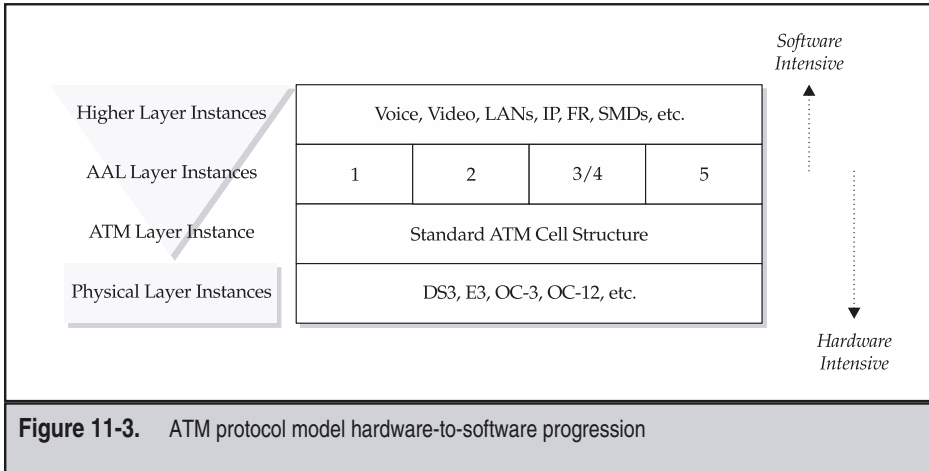
Figure 11-2 illustrates the mapping of the B-ISDN PHY, ATM, and AAL sublayers to the OSI layers employed in this text. This book uses a protocol model consisting of multiple planes, instead of attempting to fit all of the ATM-related protocols into the monolithic OSI Reference Model, as some texts attempt to do. Of course, the I.321 B-ISDN protocol model defines the physical and ATM layers that are the foundation for the user, control, and management planes.



Hardware and Software Implementations of B-ISDN Layers

The number of standardized protocols for each layer, and whether their target implementation is in hardware or software, tells us a great deal about ATM. Figure 11-3 depicts the number of instances of standard protocols at each layer by rectangles in the center of the figure. The arrows on the right-hand side illustrate the fact that ATM implementations move from being hardware-intensive at the lower layers (PHY and ATM layers) to software intensive at the higher layers (AALs and higher layers). Figure 11-3 shows the single ATM layer at the center as the pivotal protocol, shown at the tip of the inverse pyramid on the left in the figure. The singular instance of the ATM cell structure operates over a large number of physical media.

Atop the ATM layer, only four AALs support an ever-expanding set of higher-layer functions. Part 4 provides the detailed coverage of this suite of higher-layer protocols. In summary, ATM allows machines with different physical interfaces to transport data, independent of the higher-layer protocols, using a common, well-defined protocol amenable to a high-performance and cost-effective hardware implementation to support guaranteed bandwidth and service quality. This flexibility of a single, multipurpose protocol is a key objective of ATM-based equipment and service architectures. Now, in our journey up through the layers of the B-ISDN/ATM protocol model, we start with the physical layer.



ATM PHYSICAL LAYER

This section covers the key aspects of the physical (PHY) layer. The PHY layer provides for transmission of ATM cells over an electrical or optical physical transmission medium connecting ATM devices. The PHY layer has two sublayers: the Physical Medium Dependent (PMD) sublayer and the Transmission Convergence (TC) sublayer. The PMD sublayer provides for the actual transmission of the bits in the ATM cells. The TC sublayer transforms the flow of cells into a steady flow of bits and bytes for transmission over the physical medium, such as a DS1/E1, DS3/E3, or OC3/STM-1 private-line access circuit into the WAN, or twisted pair cabling within an office or residence.

Physical Medium–Dependent Sublayer

The PMD sublayer interfaces with the TC sublayer via a serial bit stream. Table 11-1 summarizes some of the popular standardized interfaces in terms of the name, physical medium, interface speed, user bit rate, and standardizing group(s) of each. The PMD sublayer clocks the bits transmitted over a variety of physical media at the line rate indicated in the table. Multimode fiber allows transmitters to use inexpensive light emitting diodes (LEDs), while single mode requires use of more expensive lasers. As seen from the table, shielded twisted pair physical media generally support higher bit rates than unshielded twisted pair. The User Bit Rate (Mbps) column indicates the actual bandwidth available to transmit 53-byte ATM cells after removing physical layer overhead. Multiple standards bodies define the physical layer in support of ATM: ANSI, ITU-T, ETSI, and the ATM Forum, as indicated in the Table 11-1.

Interface Description	Physical Medium	Line Rate (Mbps)	User Bit Rate (Mbps)	Standardizing Group(s)
$n \times$ DS0	DS1, E1	$n \times 0.064$	$n \times 0.064$	ATMF
DS1	Twisted pair	1.544	1.536	ITU, ANSI, ATMF
E1	Coaxial cable	2.048	1.920	ITU, ETSI, ATMF
$n \times$ DS1 IMA	Twisted pair	$n \times 1.544$	$n \times 1.488^*$	ATMF, ITU
$n \times$ E1 IMA	Coaxial cable	$n \times 2.048$	$n \times 1.860^*$	ATMF, ITU
J2	Coaxial cable	6.312	6.144	ITU, ATMF
Token Ring based	(Un)shielded twisted pair	32	25.6	ATMF, ITU
E3	Coaxial cable	34.368	33.92	ITU, ETSI, ATMF
DS3	Coaxial cable	44.736	40.704, 44.21 [†]	ITU, ANSI, ATMF
Midrange PHY	Unshielded twisted pair	51.84, 25.92, 12.96	49.536, 24.768, 12.384 [‡]	ATMF
STS-1	Single/multi-mode fiber	51.84	49.536	ANSI
FDDI based	Multimode fiber	125	98.15	ATMF
E4	Fiber, coaxial cable	139.264	138.24	ITU, ETSI
STS-3c	Single/multi-mode fiber	155.52	149.76	ITU, ANSI, ATMF
STM-1	Fiber, coaxial cable	155.52	149.76	ITU, ETSI, ATMF
155.52	Unshielded twisted pair	155.52	149.76	ATMF
Fiber channel based	Multimode fiber, shielded twisted pair	194.4	155.52	ATMF

Table 11-1. ATM Physical Layer Interfaces, Media, and Bit Rates

Interface Description	Physical Medium	Line Rate (Mbps)	User Bit Rate (Mbps)	Standardizing Group(s)
STS-12c	Single/multi-mode fiber	622.08	599.04	ITU, ANSI, ATMF
STM-4	Fiber, coax	622.08	599.04	ITU, ETSI
1,000 Mbps	Single/multi-mode fiber, twisted pair	1,250	1,000	ATMF, IEEE
STS-48c, STM-16	Single-mode fiber	2,488.32	2,377.728	ATMF, ITU, ETSI

[†]User bit rate for $n \times DS1$ and $n \times E1$ inverse multiplexing over ATM (IMA) physical interfaces specified by the ATM Forum assumes default value of one overhead cell for every 32 cells.

[‡]The two user bit rates for DS3 are for the older method called Physical Layer Convergence Protocol (PLCP), taken from the 802.6 Distributed Queue Dual Bus (DQDB) standard, and the new method that employs cell delineation.

[§]The lower-bit-rate values for the midrange PHY are for longer cable runs.

Table 11-1. ATM Physical Layer Interfaces, Media, and Bit Rates (*continued*)

Transmission Convergence (TC) Sublayer

The TC sublayer maps ATM cells to and from the TDM bit stream provided by the PMD sublayer. The TC sublayer delivers cells, including the 5-byte cell header, to the ATM layer at speeds up to the user bit rate indicated in Table 11-1. The user bit rate is the cell rate times the cell size of 424 bits (53 bytes). The difference between the interface speed and the user bit rate is due to physical layer overhead. On transmit, TC maps the cells into the physical layer frame format. On reception, it delineates ATM cells in the received bit stream. Generating the HEC on transmit and using it to correct and detect errors on receive are also important TC functions.

The following sections cover two examples of TC mapping of ATM cells: direct mapping to a DS1 payload and direct mapping to an STS-3c payload. The section then covers the use of the Header Error Check (HEC) and why it is so important. Another important function that TC performs is cell rate decoupling by sending idle (or unassigned) cells when the ATM layer has not provided a cell. This critical function allows the ATM layer to operate with a wide range of different speed physical interfaces. An example later in this section illustrates cell rate decoupling using unassigned or idle cells.

The remainder of this section gives examples of direct mapping by the Transmission Convergence (TC) sublayer to and from the physical layer bit stream provided by the PMD sublayer.

DS1 Direct Mapping

Figure 11-4 illustrates direct cell delineation mapping by the TC sublayer for the DS1 physical interface. Note that the cell boundaries need not align with octet boundaries defined for DS1, for example, as defined in ISDN. Most TC layer specifications are similar in form to this standard. For further details, see the ATM Forum DS1 UNI physical layer specification [ATMF PHY-0016] or the ANSI T1.646 standard.

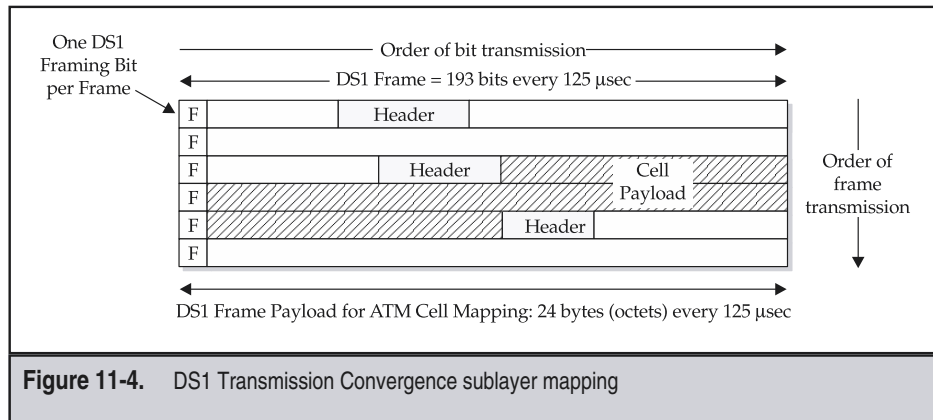
SONET STS-3c Direct Mapping

Figure 11-5 illustrates the direct mapping of ATM cells into the SONET STS-3c (155.52 Mbps) Synchronous Payload Envelope (SPE) defined in Chapter 6. Note that ATM cells continuously fill the STS-3c payload, since an integer number of 53-octet cells do not fit in an STS-3c frame.

The ATM User-Network Interface (UNI) 3.1 specification eliminates a number of SONET overhead functions to reduce the complexity, and hence cost, of SONET-based ATM interfaces. For example, the UNI eliminates the requirement for the processing-intensive SONET Data Communications Channel (DCC). The TC sublayer uses the HEC field to delineate cells from within the SONET payload. The user data rate is computed as 9 rows times 260 columns of bytes at 8000 SONET frames per second, or 149.76 Mbps. For further details, see the ATM Forum UNI 3.1 specification or ANSI standard T1.646. For a more readable discussion of ATM mapping into the SONET payload, see [Goralski 95]. The mapping over STS-12c is similar in nature. The difference between the North American SONET format and the international SDH format exists in the TDM overhead bytes.

TC Header Error Check (HEC) Functions

The Header Error Check (HEC) is a 1-byte code applied to the 5-byte ATM cell header capable of correcting any single-bit error in the header. It also detects many patterns of



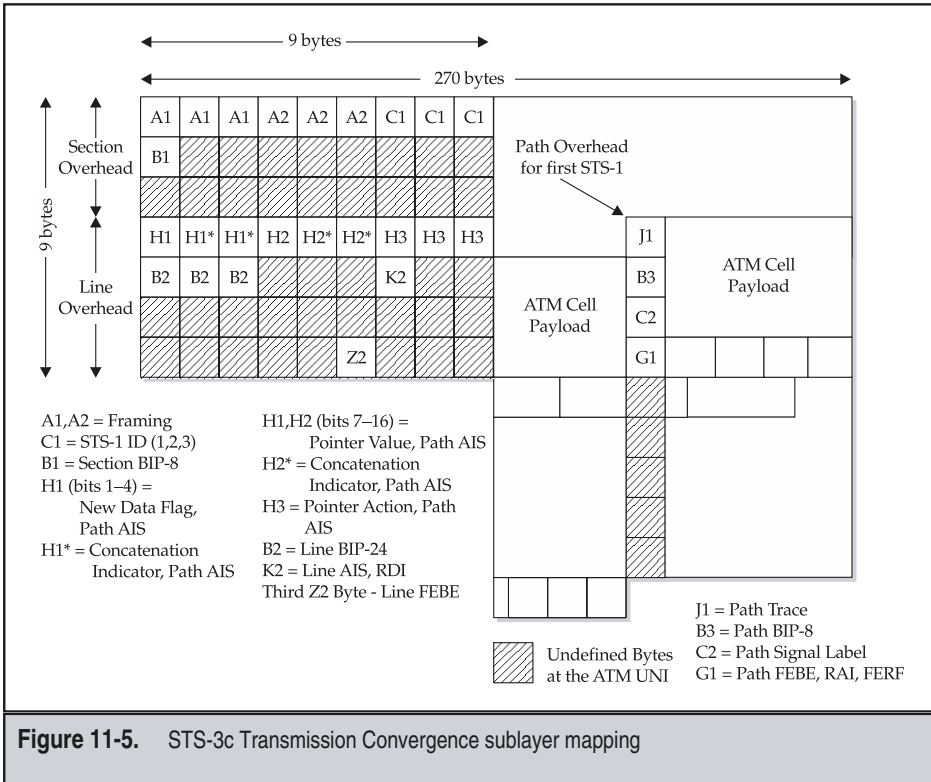


Figure 11-5. STS-3c Transmission Convergence sublayer mapping

multiple-bit errors. The TC sublayer generates the HEC on transmit and uses the received HEC field to determine if the received header has any errors. The receiver may either correct or discard the cell whether HEC detects a single bit error, but it must discard the cell if HEC detects more than one error. Chapter 23 presents an analysis of the undetected error rate to help users decide on the HEC correction or detection option for their particular application. Since the header tells the ATM layer what to do with the cell, it is very important that it not have errors; if it did, the cell might be delivered to the wrong user or inadvertently invoke a function in the ATM layer.

The TC also uses the HEC to locate cell boundaries when they are directly mapped into a TDM payload, for example, as in the DS1 and STS-3c mappings described in earlier sections. During startup, the receiver looks for a valid HEC (i.e., one without any errors identified); and once a valid value is found, it looks 53 bytes later for the next one. Once several valid cell headers have been identified, the receiver now knows where to look in the sequence of received bytes for the next cell. This method works because HEC infrequently

matches random data in the cell payloads when the 5 bytes being checked are not part of a valid cell header. Thus, almost all ATM standards employ the HEC to locate cell boundaries in the received bit stream. One notable exception is the North American standard for DS3, which uses a separate Physical Layer Convergence Protocol (PLCP) for cell delineation. Once the TC sublayer locates several consecutive cell headers in the bit stream received from the PMD sublayer through the use of the HEC, then the TC knows to expect the next cell 53 bytes later. Standards call this process *HEC-based cell delineation*. Thus, the fixed length of ATM cells aids in detecting valid cells reliably.

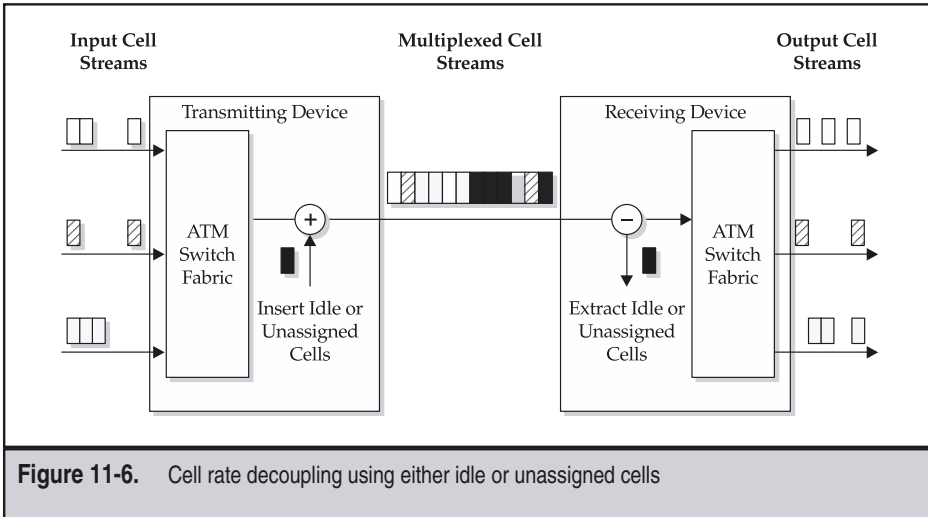
TC Cell Rate Decoupling

The TC sublayer performs a cell rate decoupling, or speed matching, function, as well. Physical media that have synchronous cell time slots (e.g., DS3, SONET, SDH, STP, and the Fiber Channel–based method) require this function, while asynchronous media such as the FDDI PMD do not. As we shall see in the next section, special codings of the ATM cell header indicate whether a cell is either *idle* or *unassigned*. All other cells are *assigned* and correspond to cells generated by the ATM layer. Figure 11-6 illustrates an example of cell rate decoupling between a transmitting and receiving ATM device. Starting at the left-hand side of the figure, the transmitting ATM device multiplexes multiple cell streams together, queuing them if a time slot is not immediately available on the physical medium. If the queue is empty when the time to fill the next cell time slot arrives as determined by the physical layer, then the TC sublayer in the transmitter inserts either an unassigned cell or an idle cell, indicated by the solid black filled cells in the figure. The receiving device extracts unassigned or idle cells and distributes the other, assigned cells to the destinations determined from the VPI and VCI values, shown on the right-hand side of the figure. Also, note how the act of multiplexing and switching changes the intercell spacing in the output cell streams, as the reader can see from the example.

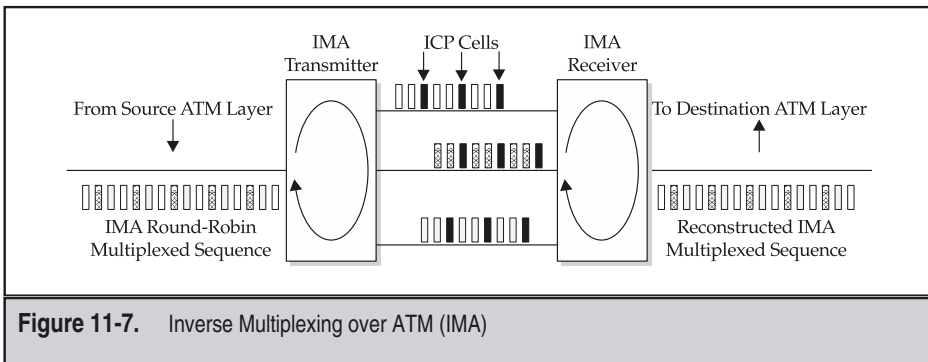
ITU-T Recommendation I.321 originally placed the rate decoupling function in the TC sublayer of the PHY layer using idle cells, while the ATM Forum placed it in the ATM layer using unassigned cells. The ITU-T model viewed the ATM layer as independent of whether or not the physical medium has synchronous time slots. Since the ITU-T initially standardized on idle cells, while the ATM Forum chose unassigned cells for cell-rate decoupling, these methods were incompatible because the coding of the cell header to indicate idle or unassigned cells differed. In 1996, ITU-T Recommendation I.361 aligned itself with the ATM Forum's usage of the unassigned cell for rate decoupling. Therefore, the difference in usage of idle versus unassigned cells occurs only in older equipment.

Inverse Multiplexing over ATM

The Inverse Multiplexing over ATM (IMA) specification [AF-PHY-86.1, ITU I.761] standardizes access at $n \times DS1$ and $n \times E1$ rates. IMA combines multiple DS1 or E1 access circuits into a single aggregate transmission path with approximately n times the bandwidth of a single DS1 or E1 circuit. Hence, IMA provides a capability similar to proprietary $n \times DS1$ or $n \times E1$ TDM multiplexers, but in a standard manner. Figure 11-7 illustrates the operation of



an IMA system. Cells arriving from the sending ATM layer originate at the upper left-hand side of the figure. The IMA transmitter multiplexes these cells in a round-robin manner onto the three physical links in the IMA group, interleaving IMA Control Protocol (ICP) cells for control and synchronization purposes. IMA cells (ICP or filler) are specially coded OAM cells that IMA does not convey to the ATM layer. IMA transparently carries standard ATM OAM cells as detailed in Part 7. The shading of cells in the figure indicates the link selected by the IMA transmitter. At the receiver, the cells on each of the physical links may arrive after different delays on each of the physical connections, for example, due to different circuit lengths. The maximum differential delay required by the



IMA specification is 25 ms, or approximately a difference of 2,500 miles in the physical circuit propagation delay when operating over optical fiber.

On the right-hand side of the figure, the IMA receiver employs the ICP cells to realign the user cells before multiplexing them back into an accurately reproduced version of the original high-speed ATM cell stream, which it then delivers to the destination ATM layer. The net effect of IMA is that the end equipment sees an ATM cell stream operating at approximately an $n \times \text{DS1}$ or $n \times \text{E1}$ aggregate rate. At the default rate of one ICP cell every 32 cell times, the IMA consumes approximately three percent overhead.

Several ATM switch, access multiplexer, and concentrator vendors now offer support for IMA. IMA offers a cost-effective way to garner greater than DS1 or E1 bandwidth across the WAN, without having to lease a full DS3 or E3 access line. Furthermore, IMA allows users to add bandwidth in DS1 or E1 increments, typically up to 8 DS1s or E1s, at which point a user can cost-justify the purchase of a more expensive DS3/E3 or SONET/SDH access line. Carriers began offering $n \times \text{DS1}$ IMA services in the United States in 1997. This service is especially useful for international connectivity where bandwidth is even more expensive than in the United States.

xDSL Physical Layer for ATM

Digital Subscriber Line (or DSL for short) technology is an important access method for consumers and small businesses. This section summarizes reference configuration and important technical parameters.

xDSL Explained

The term xDSL refers to a family of communication technologies based around the DSL technology, where “x” corresponds to standard upstream and downstream data rates, defined from the perspective of the end user. The term *upstream* refers to transmission from the user to the service provider; while the term *downstream* refers to transmission from the service provider to the end user. Several groups of manufacturers, carriers, and entrepreneurs are actively standardizing and promoting this technology in the DSL Forum. They’ve coined a set of acronyms, explained in Table 11-2. The table gives the upstream and downstream rates, the number of twisted pairs required, and a list of representative applications. In symmetric DSLs, the upstream rate is the same as the downstream rate. In asymmetric DSLs, the downstream rate is typically much higher than the upstream rate, which is suitable for Internet access—for example, Web browsing.

Digital Subscriber Line (DSL) technology harnesses the power of state-of-the-art modem design to “supercharge” existing twisted pair telephone lines into information superhighway on-ramps. For example, ADSL technology enables downstream transmission speeds of over 1 Mbps to a subscriber, while simultaneously supporting transmissions of at least 64 Kbps in both directions. xDSL achieves bandwidths orders of magnitude over legacy access technologies, such as ISDN and analog modems, using existing cabling, albeit over shorter distances. The basic concept of xDSL is to utilize frequency spectrum on the twisted pair beyond the 4 kHz traditional POTS channel. It divides the frequency

Acronym	Full Name	Pairs	Upstream Rate	Downstream Rate	Example Applications
DSL	Digital Subscriber Line	1	160 Kbps	160 Kbps	ISDN service for voice and ISDN modem data
HDSL (DS1)	High data rate Digital Subscriber Line	2	1.544 Mbps	1.544 Mbps	North American T1 service
HDSL (E1)	High data rate Digital Subscriber Line	3	2.048 Mbps	2.048 Mbps	European and International E1
SDSL (DS1)	Single line Digital Subscriber Line	1	1.544 Mbps	1.544 Mbps	North American T1 service
SDSL (E1)	Single line Digital Subscriber Line	1	2.048 Mbps	2.048 Mbps	European and International E1
CDSL	Consumer Digital Subscriber Line	1	64 to 384 Kbps	1.5 Mbps	"Splitterless" operation to the home (g.Lite)
ADSL	Asymmetric Digital Subscriber Line	1	16 to 640 Kbps	1.5 to 9 Mbps	Video on demand, ATM, LAN, and Internet Access
VDSL	Very high data rate Digital Subscriber Line	1	1.5 to 2.3 Mbps	13 to 52 Mbps	High-quality video and high-performance Internet/LAN

Table 11-2. Digital Subscriber Line (xDSL) Acronyms Explained

spectrum (up to several MHz) into many frequency channels. Since each channel has different quality in terms of SNR, different bit rate can be achieved by each channel. Higher bit rate xDSL technologies either utilize more bandwidth or increase the bit rate of each channel. Another important parameter for xDSL is the distance of the twisted pairs. For example, ADSL at a lower rate usually requires a wiring distance of less than 12,000 feet, while vDSL at a higher rate could be achieved only when the wiring distance is less than 500 feet.

DSL competes with cable modem in this marketplace, and, at the time of this writing, cable modem had more customers than xDSL. Several factors are in play here. One is the operational difficulty of finding twisted pairs over which the high-speed xDSL signals can operate. Another set of issues revolve around attempts to empower competition and regulate incumbent carriers. For example, in the United States, many of the competing xDSL carriers have gone out of business, while the incumbent local exchange carriers seek regulatory relief in order to better compete with the cable modem-based competition. However, many experts expect xDSL to play a crucial role over the coming decade as small business and consumer demand for video and multimedia information increases because it uses existing unshielded twisted pair cabling, while the installation of new, higher performance cabling (e.g., optical fiber) will take many years.

Figure 11-8 illustrates a typical xDSL configuration. The equipment on the left-hand side is located on the customer premises. Devices that the end user accesses—such as a personal computer, television, and N-ISDN video phone—connect to a premises distribution network via service modules. These service modules employ STM (i.e., TDM), ATM, or packet transport modes, as depicted by the arrows at the bottom of the figure. An existing twisted pair telephone line connects the user's xDSL modem to a corresponding modem in the public network. The xDSL modem creates three information channels—a high-speed downstream channel ranging from 1.5 to 52 Mbps, a medium-speed duplex channel ranging from 16 Kbps to 2.3 Mbps, and a POTS (Plain Old Telephone Service) channel.

Many xDSL schemes require analog splitters at the user's site to separate the POTS channel from the digital modem. Basically, a splitter is a filter that passes only the lower 4 kHz of spectrum to the local telephone devices. This design guarantees uninterrupted telephone service, even if the xDSL system fails. However, if the telephone company must send a technician to install such a device, the cost of the service increases markedly. The CDSL method targets this problem explicitly by eliminating the need for telephone company-installed splitters. In CDSL, data traffic is also transmitted over the voice band when there is no voice traffic to be transmitted. Thus, there is no need for a splitter. Splitter-less xDSL modems are much simpler and can be customer. Other service providers give users easily installed micro-splitters that attach to each user telephone, fax machine, or other legacy telephone-compatible devices.

Moving to the center of Figure 11-8, multiple xDSL modems connect to an access node within the public network. Frequently, these are called DSL Access Multiplexers (DSLAMs). Splitters in the telephone company Central Office (CO) carve off analog voice signals, multiplex them together, digitize them, and deliver the aggregate voice traffic to the Public Switched Telephone Network (PSTN). There is also another approach to

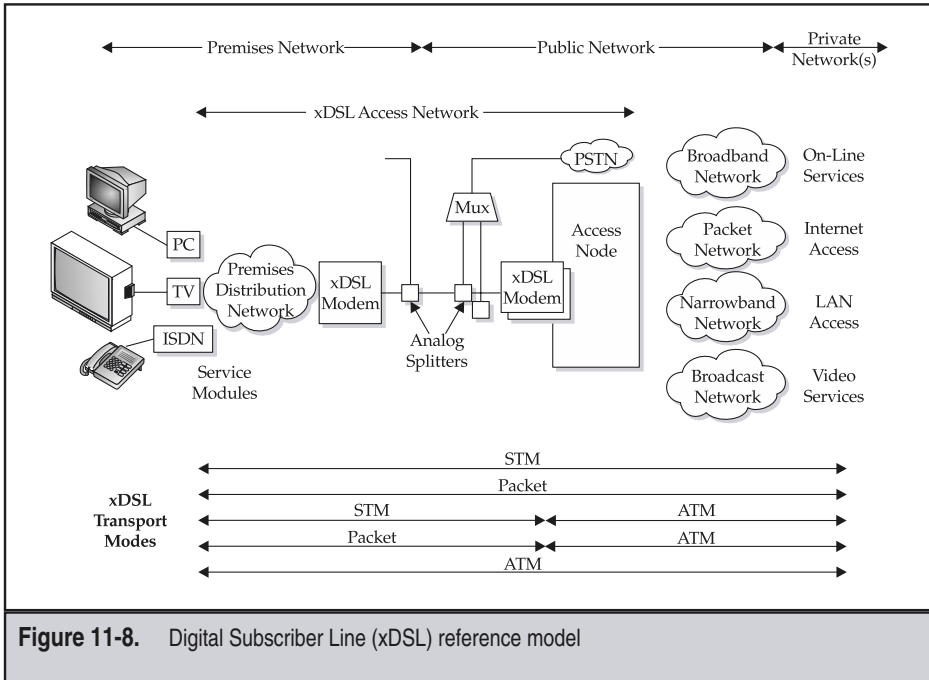


Figure 11-8. Digital Subscriber Line (xDSL) reference model

efficiently transporting voice, voice-band data, fax traffic, and ISDN over DSL (or wireless physical layers, IEEE 802.11 or 802.16) as specified in the ATM Forum's loop emulation service specifications. In this case, voice would be carried over ATM by using AAL2 [AF VMOA 145]. Voice transport can then include support for non-compressed voice or any number of methods for compressing voice, together with silence removal. This basic approach provides an ATM transport layer for the voice traffic, and refinements in this area follow the current industry trend of distributing functionality to the edges of the network. What started with an effort to support voice by extending traditional voice functions to the DSL modem with digital carrier loop protocols [GR-303], effectively enables implementation of a remote voice switch port. This work is now actively adding distributed control with media gateway and media control protocols [ITU H.248] for both voice and other media at the edge, eliminating the need to connect to a centrally located voice switch, since all voice services can be performed at the edge. We will come back to this later in Chapter 16, when we briefly look at the protocols for the next-generation distributed voice services architecture. The access node interfaces to packet, broadband, video, or service networks, as shown on the right-hand side of Figure 11-8. Depending upon the transport mode employed by the customer, conversion to/from ATM may be performed as indicated by the lines at the bottom of the figure.

ATM LAYER

Now we move up one more layer to the focal point of the B-ISDN protocol model: the Asynchronous Transfer Mode (ATM) layer. This section details the key context and definitions of the ATM layer, which include

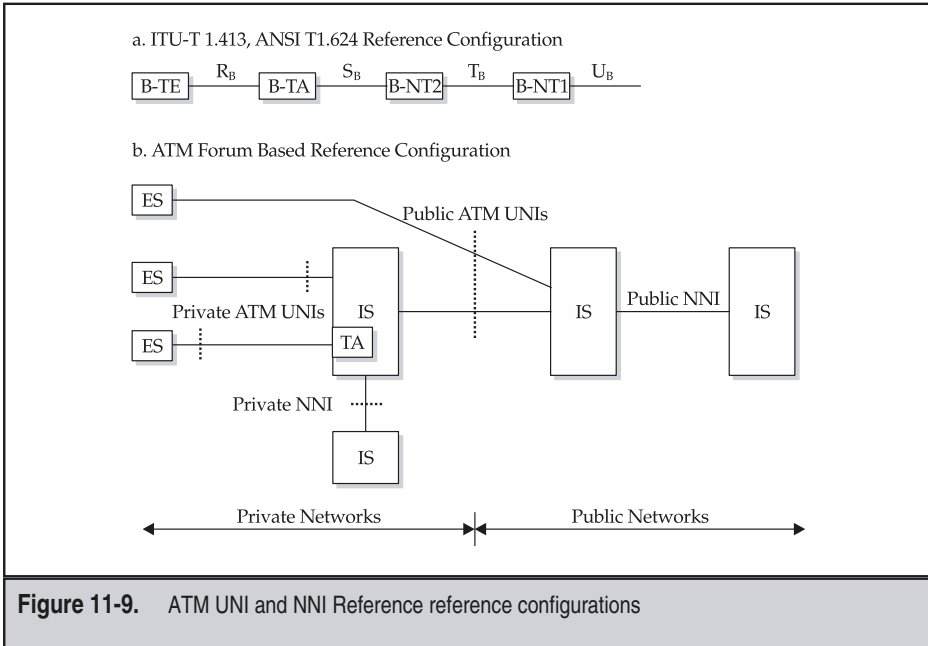
- ▼ Cell Construction
- Cell Reception and Header Validation
- Cell Relaying, Forwarding, and Copying Using virtual connections
- Cell Multiplexing and Demultiplexing Using virtual connections
- Cell Payload Type Discrimination
- Interpretation of Pre-defined Reserved Header Values
- Cell Loss Priority (CLP) bit Processing
- Support for Multiple QoS Classes
- Usage Parameter Control (UPC)
- Explicit Forward Congestion Indication (EFCI)
- Generic Flow Control (GFC)
- ▲ Connection Assignment and Removal

Part 5 covers the key topics of support for multiple QoS classes, UPC, EFCI, GFC, and connection assignment and removal. We begin our description of ATM with a summary of its networking terminology followed by discussion of the relationship of the ATM layer to the physical layer, and its further division into Virtual Path (VP) and Virtual Channel (VC) connections.

ATM UNI and NNI Defined

Figure 11-9a gives the ITU-T and ANSI-oriented B-ISDN view defined in I.413 for the User-Network Interface (UNI) and the Network Node Interface (NNI). ATM UNIs interconnect user equipment, called Broadband Terminal Equipment (B-TE), to either a Terminal Adapter (TA) or Network Termination (NT) device prior to interfacing to an ATM network. These standards assign letters to the interfaces between each of these functional blocks. Similar to the ISDN reference model described in Chapter 6, the ANSI standards define the U reference point, while the international standards do not.

Figure 11-9b shows the ATM Forum terminology of private and public UNIs corresponding to the ITU-T reference point terminology above it. The ATM UNI is either a private ATM UNI, which occurs at the R or S reference point in ITU-T Recommendation I.413 and ANSI T1.624, or a public ATM UNI, which occurs at reference points T or U, as shown in the figure. Normally, we refer to the Network-Node Interface (NNI) defined in ITU-T Recommendation I.113 as the standard interface between networks; however, it



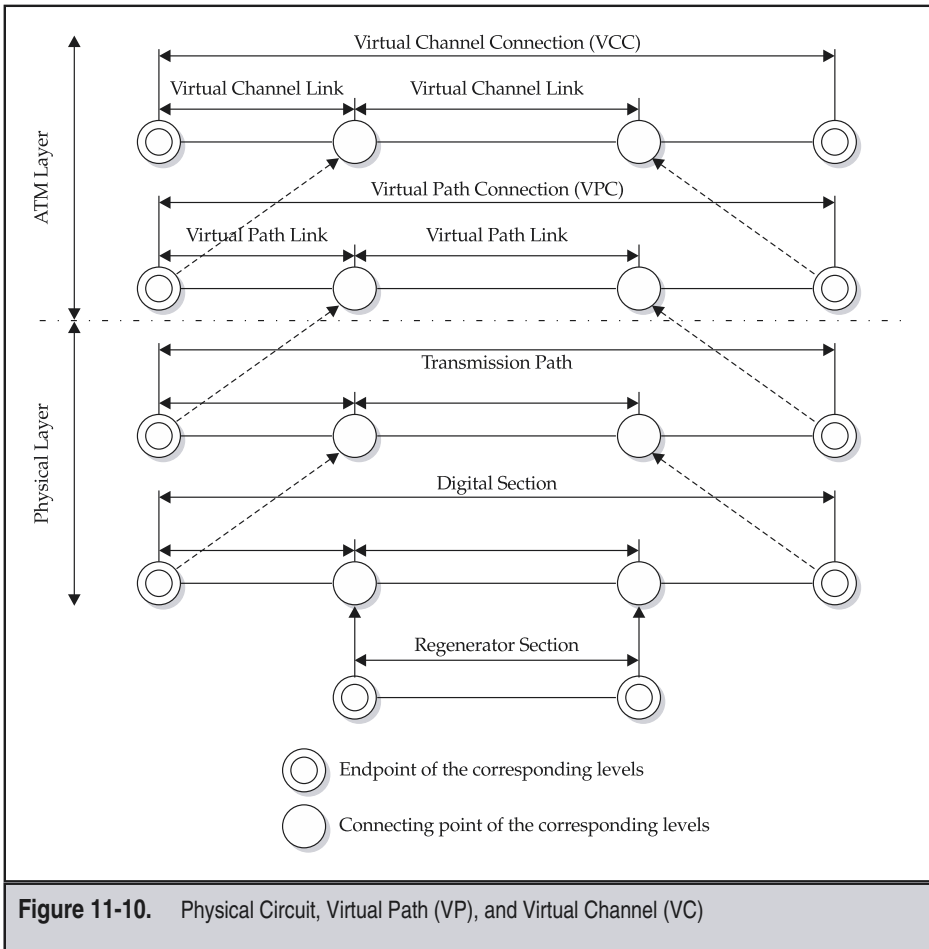
will most likely also be the interface used between nodes within a network. The ATM Forum distinguishes between an NNI used for private networks and one for public networks, as shown in Figure 11-9. This book uses primarily the ATM Forum terminology.

Two standardized coding schemes exist for cell structure: the User-Network Interface (UNI) and the Network Node, or Network to Network Interface (NNI). The UNI is the interface between the user, or CPE, and the network switch. The NNI is the interface between switches or between networks. We introduce the UNI and NNI coding schemes, and then detail the format and meaning of each field as defined in this section. ITU-T Recommendation I.361 is the basis of these definitions, with further clarifications and implementation details given in the ANSI T1.627 standard and the ATM Forum UNI and Broadband Inter-Carrier Interface (B-ICI) specifications.

ATM Virtual Paths and Channels (VPs and VCs)

As shown in Figure 11-10 from ITU-T Recommendation I.311, a key concept is the construction of end-to-end ATM Virtual Path and Virtual Channel Connections (VPCs and VCCs) from one or more VP or VC links. The physical layer has three levels: regenerator section, digital section, and transmission path, as shown in the figure. The ATM layer uses

only the transmission path of the physical layer, which is effectively the TDM payload that connects ATM devices. Generically, an ATM device may be either an endpoint or a connecting point for a VP or VC. A VPC or a VCC exists only between endpoints, as shown in Figure 11-10. A VP link or a VC link exists between an endpoint and a connecting point or between connecting points, also indicated in the figure. A VPC or VCC is an ordered list of VP or VC links, respectively. The control plane establishes VPCs and VCCs by provisioning, in the case of a Permanent Virtual Connection (PVC), or by signaling, in the case of Switched Virtual Connections (SVCs), as detailed in Chapter 13. The following sections define these terms more precisely.



Virtual Channels (VCs)

The *Virtual Channel Identifier (VCI)* in the cell header identifies a single VC on a particular Virtual Path (VP). A VC connecting point switches based upon the combination of the VPI and VCI. A *VC link* is a unidirectional flow of ATM cells with the same VPI and VCI between a VC connecting point and either a VC endpoint or another VC connecting point. A *Virtual Channel Connection (VCC)* is a concatenated list of VC links traversing adjacent VC switching ATM nodes. A VCC defines a *unidirectional* flow of ATM cells from one user to one or more other users. A point-to-point bidirectional VCC is actually a pair of point-to-point, unidirectional VCCs relaying cells in opposite directions between the same endpoints.

Intermediate ATM VC switches may modify both the VPI and VCI values at connecting points to forward cells along VC links that comprise an end-to-end VCC. Note that VPI and VCI values must be unique on any particular physical interface. ATM VC switches use VPIs and VCIs independently on each interface. In other words, the VPI and VCI have local significance only for a VCC.

A network must preserve cell sequence integrity for a VCC; that is, the cells must be delivered in the same order in which they were sent. This means that ATM devices deliver cells to intermediate connecting points and the destination endpoint in the same order transmitted by the originating endpoint. Each VCC has an associated Quality of Service (QoS).

Virtual Paths (VPs)

Virtual Paths (VPs) define an aggregate bundle of VCs between VP endpoints. A *Virtual Path Identifier (VPI)* in the cell header identifies a bundle of one or more VCs. A *VP link* provides unidirectional transfer of cells with the same VPI between VP endpoints or VP connecting points. A VP connecting point switches based upon the VPI only—it ignores the VCI field. A *Virtual Path Connection (VPC)* is a concatenated list of VP links between adjacent VP switching nodes. A VPC defines a unidirectional flow of ATM cells from one user to one or more other users. A point-to-point bidirectional VPC is actually a pair of point-to-point unidirectional VPCs relaying cells in opposite directions between the same endpoints.

The users of the VPC may assign the VCCs within that VPI transparently, since they follow the same route. Therefore, a VP switch cannot modify the VCI values. Intermediate ATM VP switches may modify only the VPI values at connecting points to forward cells along VP links that comprise an end-to-end VPC. Note that VPI values must be unique on any particular physical interface. In other words, the VPI has local significance only for a VPC, while the VCI has end-to-end significance.

Standards do not require a network to preserve cell sequence integrity for a VPC; however, the cell sequence integrity requirement of a VCC still applies. Each VPC has an associated Quality of Service (QoS). If a VPC contains VCCs with different QoS classes, then the VPC assumes the QoS of the VCC with the highest QoS class.

Usage of VPI and VCIs in ATM Networks

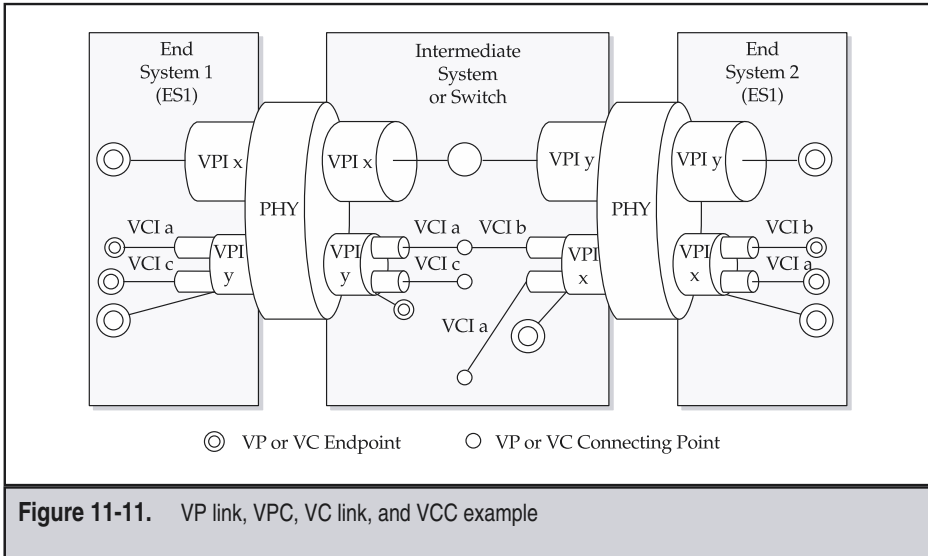
Standards reserve VPI = 0 for VCCs. Therefore, the number of bits allocated in the ATM cell header to the VPI limit each physical UNI to no more than $2^8 - 1 = 255$ virtual paths and each physical NNI to no more than $2^{12} - 1 = 4095$ virtual paths. Each virtual path can support no more than $2^{16} = 65,536$ virtual channels on the UNI or the NNI. Even though each ATM access circuit can contain a combination of up to 255 VPCs and 65,536 VCCs per VP in theory, service providers and equipment manufacturers typically support less. Check with your equipment vendor or service provider regarding the specific VPI and VCI values supported. Generally, higher-speed circuits support far more VPCs and VCCs than lower-speed ones do.

Although the UNI and NNI cell formats specify 8 and 12 bits for the VPI, respectively, and 16 bits for the VCI on both interfaces, real ATM systems typically support a smaller number of the lower-order bits in the VPI and VCI. Ranges of VPI/VCI bits supported by interconnected devices must be identical for interoperability. One way to handle VPI/VCI interoperability is to use the ATM Forum's Integrated Local Management Interface (ILMI), which allows each system to query the other about the number of bits supported, thus guaranteeing interoperability.

VP and VC Switching and Cross-Connection

This section provides a specific example of VP and VC endpoints and connecting points in intermediate and end systems for VP links, VPCs, VC links, and VCCs. Figure 11-11 depicts two end systems (ES or CPE) and an intermediate system (IS or switch). The endpoint and connecting points use the terminology and notation from Figure 11-10. The physical interface, Virtual Path, and Virtual Channel are shown as a nested set of *pipes* using the convention from ITU-T Recommendation I.311. The transmission path PHY layer carries VPs and VCs that are either unidirectional or bidirectional. This example shows end systems (or CPEs) with both VP and VC endpoints. The left-hand-side end system, or CPE, originates a VP with VPI x and two VCs with VCI values equal to a and c .

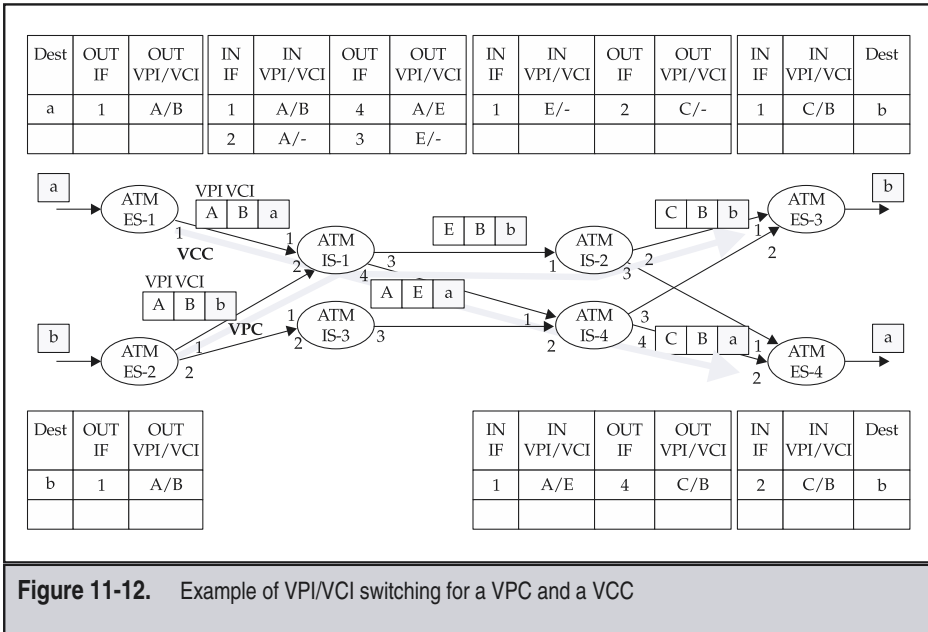
The intermediate system (or switch) contains VP and VC switching functions, as shown in Figure 11-11. The intermediate system VP switching function translates the VPI from x to y , since VPI x is already in use on the physical interface to the destination end system. The VP switching function automatically connects all VCs within VPI x to VPI y for this pair of VPI values. This simultaneous switching of a large number of VCs within a VP is the principal reason for the standardization of VPs. If ATM employed only a single level of interface-level addressing in the cell header, this function would be impossible. VC switching operates within VPs as illustrated by the other VC connection terminating within the switch. Now looking at the VCs on VPI y , the VC switching function in the switch translates the received VCI a to an outgoing VCI b on VPI x for delivery to the destination. VCI c from VPI y is switched to some other destination (not shown in the figure) using a connecting point. Similarly, the switch takes VCI a coming from another physical interface and/or VPI and places it within VPI x for delivery to End System 2.



Example of ATM Virtual Path and Channel Cell Switching

This section presents a detailed example of a VPC and VCC in terms of the VPI and VCI switching actions taken by a source and destination end system, as well as the actions taken by the intermediate switches along the path from a source to destination with reference to Figure 11-12. The figure shows ATM end systems (ES) at the left and right with intermediate systems (IS) in the middle, all connected via physical interfaces (IF) identified by a number that is unique to each node. The ATM layer-related function performed by each node is summarized in a table either immediately above or below each ES and IS. The example for a VPC originates at ES-2, traverses IS-1 and IS-2, and terminates at ES-3. At source ES-2, a payload with a destination (Dest) b is mapped to outgoing interface (OUT IF) 1 and labeled with an outgoing VPI/VCI (OUT VPI/VCI) value of A/B, as shown in the figure. IS-1 recognizes cells with incoming VPI = A arriving on interface 2 as part of a VPC, indicated in the IN VPI/VCI column in the table where the VCI value is a dash, indicating a “don’t care” condition. IS-1 determines from the switching lookup table that the outgoing interface, and the outgoing VPI, is E. Next, IS-2 receives the cell with VPI/VCI E/B on incoming interface 2, and replaces the VPI value with C. Note that the VCI is unchanged by the VP switching operation performed by IS-1 and IS-2. Finally, the cell arrives at the destination ES-3, which looks at both the VPI and VCI to determine that the contents of the cell(s) should be delivered to destination b.

Figure 11-12 also contains an example of a VCC, originating at ES-1 and then switched by IS-1 and IS-4 through to destination ES-4. At source ES-1, a payload with logical destination a is mapped to an outgoing VPI/VCI = A/B on interface 1. IS-1 receives this cell, and



from its switching table determines that the outgoing interface is 4 and that the outgoing VPI/VCI is A/E. Here the VCI changed, but the VPI did not. Continuing with the next node in the VCC, IS-4 consults its switching table for incoming interface 1 with incoming VPI/VCI = A/E and determines that the cell should be switched to outgoing interface 4, with outgoing VPI/VCI = C/B. At this node, both the VPI and VCI are changed. Finally, at destination ES-4, the switching table maps the cells coming in on interface 2 with VPI/VCI = C/B to logical destination a.

We now cover the detailed coding of the ATM cell, including the VPI and VCI values introduced previously, along with all of the other values in the ATM cell header.

The ATM Cell

The unit of transmission, multiplexing, and switching in ATM is the fixed-length *cell*. Standards bodies chose a fixed-length ATM cell to simplify hardware design. In fact, for this reason, some packet switching devices allocate hardware buffers for each packet equal to the size of the maximum packet to simplify the hardware design. As the price of memory declines and the density of logic increases, the advantage of a short fixed-length cell in simplifying hardware design decreases.

ATM standards define a fixed-size cell with a length of 53 octets (or bytes) made up of a 5-octet header (H) and a 48-octet payload (P), as shown in Figure 11-13. An ATM device

transmits the bits from the cells over the physical transmission path in a continuous stream. ATM networks switch and multiplex all information using these fixed-length cells. The cell header identifies the destination port through the label swapping technique, as well as the payload type and loss priority. The VPI and VCI have significance only on a single interface, since each switch translates the VPI/VCI values from input port to output port along the path of an end-to-end connection. Therefore, in general, the VPI/VCI values on the input and output ports of the same switch differ. The sequence of VPI/VCI mappings in the switches along the path makes up the end-to-end connection. The ITU-T and the ATM Forum reserve VCI values 0 through 31 for specific functions, as detailed later. The Generic Flow Control (GFC) field allows a multiplexer to control the rate of an ATM terminal. While B-ISDN standards summarized in Chapter 22 define GFC, only a few implementations actually use the standard. The format of the ATM cell at the Network-Node Interface (NNI) eliminates the GFC field and instead uses the 4 bits to increase the VPI field to 12 bits, as compared to 8 bits at the User-Network Interface (UNI).

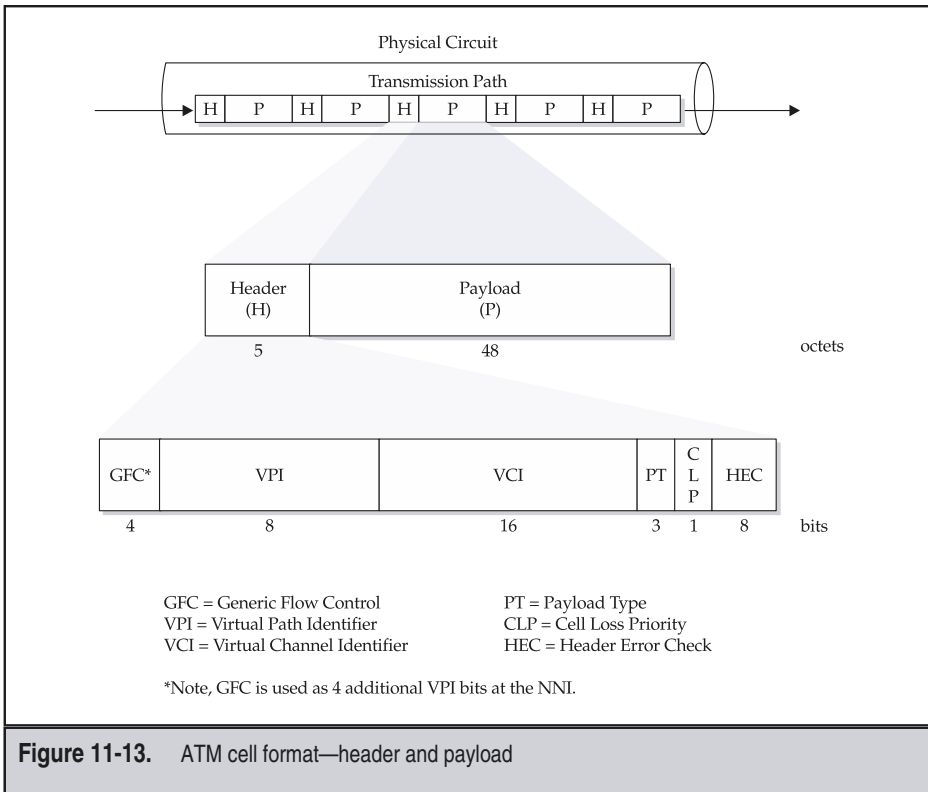


Figure 11-13. ATM cell format—header and payload

The Cell Loss Priority (CLP) bit indicates the relative priority of the cell similar to the Discard Eligible (DE) bit in Frame Relay service. Lower-priority cells may be discarded before higher-priority cells by the Usage Parameter Control (UPC) at the ingress to the ATM network, if cells violate the predetermined user contract, or within the network during periods of congestion.

The 53-byte ATM cell at the User-Network Interface (UNI) has an address significant only to the local interface in two parts: an 8-bit Virtual Path Identifier (VPI) and a 16-bit Virtual Channel Identifier (VCI). The cell header also contains a 4-bit Generic Flow Control (GFC), a 3-bit Payload Type (PT), and a 1-bit Cell Loss Priority (CLP) indicator. An 8-bit Header Error Check (HEC) field protects the entire header from errors. Later in this section, more detailed definitions are given for the meaning of each header field. A fundamental concept of ATM is that switching occurs based upon the VPI/VCI fields of *each* cell. Switching done on the VPI only is called a Virtual Path Connection (VPC), while switching done on both the VPI/VCI values is called a Virtual Channel Connection (VCC).

The Network Node Interface (NNI) format is identical to the UNI format with two exceptions. First, there is no Generic Flow Control (GFC) field. Second, the NNI uses the 4 bits used for the GFC at the UNI to increase the VPI field to 12 bits at the NNI, as compared to 8 bits at the UNI.

We now further describe the meaning of each field in the ATM cell header from ITU-T Recommendations, ANSI standards, and ATM Forum specifications.

Meaning of Preassigned Reserved Header Values

A key function of the ATM layer is the identification and processing of preassigned, reserved header values. Table 11-3 shows the preassigned (also called predefined) header field values for the UNI defined in ITU-T Recommendation I.361. The 4-bit GFC field applies to all of these values. The ITU-T reserves the first 16 VCIs for future assignment as preassigned, reserved header value functions. Other portions of the book cover the use of these specific header values as indicated in Table 11-3.

Usage	VPI	VCI	PT	CLP
Unassigned Cell	0	0	XXX	0
Invalid	Non-zero	0	XXX	X
Meta-signaling (I.311)	Any value	1	0AA	C
General broadcast signaling	Any value	2	0AA	C
Segment OAM F4 Cell	Any value	3	0A0	A
End-to-end OAM F4 Cell	Any value	4	0A0	A

Table 11-3. Preassigned, Reserved ATM Cell Header Values

Usage	VPI	VCI	PT	CLP
Point-point signaling	Any value	5	0AA	C
VP RMCCell	Any value	6	110	A
Future VP Functions	Any value	7	0AA	A
Segment OAM F5 Cell	Any value	Any value other than 0, 3, 4, 6, or 7	100	A
End-to-End OAM F5 Cell	Any value	Any value other than 0, 3, 4, 6, or 7	101	A
VC RM Cell	Any value	Any value other than 0, 3, 4, 6, or 7	110	A
Reserved for future VC functions	Any value	Any value other than 0, 3, 4, 6, or 7	111	A

X = "Don't Care" A = Used by appropriate function C = Originator-set CLP

Table 11-3. Preassigned, Reserved ATM Cell Header Values (*continued*)

Chapter 13 describes the general broadcast and point-to-point signaling functions. Chapter 28 detail the OAM cell flow formats and functions. Chapter 22 details the use of the Resource Management (RM) cell in the Available Bit Rate (ABR) service category. VC RM cells are invalid on VCI values reserved for other functions, such as signaling or OAM. We described the use of the unassigned and idle cell types earlier in this chapter. The NNI has an additional 4 bits in the VPI field.

The VCI values in Table 11-3 identify the currently defined VCCs out of the range 0–15 reserved by the ITU-T. The ATM Forum has assigned some of the next 16 VCI values (16–31) on every VPI in support of other protocols. Table 11-4 summarizes these currently reserved VCI values.

Meaning of the Cell Loss Priority (CLP) Field

A value of 0 in the Cell Loss Priority (CLP) field means that the cell is of the highest priority—or, in other words, the network is least likely to discard CLP = 0 cells in the event of congestion. A value of 1 in the CLP field means that this cell has low priority—or, in other words, the network may selectively discard CLP = 1 cells during congested intervals in order

VCI Value	Purpose
1	Meta-Signaling
2	General broadcast signaling
3	Segment F4 (i.e., VP) OAM Cell
4	End-to-end F4 (i.e., VP) OAM Cell
5	Point-to-point Signaling Channel
6	VP Resource Management Cell
7	Reserved for Future VP Functions
16	Integrated Local Management Interface (ILMI)
17	LAN Emulation Configuration Server (LECS)
18	Private Network-Network Interface (PNNI) routing channel

Table 11-4. VCI Values Reserved by the ITU-T and ATM Forum

to maintain a low loss rate for the high-priority $CLP = 0$ cells. The value of CLP may be set by the user or by the network as a result of a policing action. Part 5 details the uses of the CLP bit in traffic and congestion control.

Meaning of the Payload Type (PT) Field

Table 11-5 depicts Payload Type (PT) encoding from ITU-T Recommendation I.361. Observe that the rightmost bit is an AAL indication bit (currently used by AAL5 to identify the last cell in a packet). The middle bit indicates upstream congestion, and the first bit discriminates between data and operations cells. Payload types carrying user information may indicate congestion by the Explicit Forward Congestion Indication (EFCI) bit. Also, user cells may indicate whether the cell contains an indication to the AAL protocol. OAM and resource management cells cannot indicate congestion or AAL function. The management information payload type for the F5 flow indicates whether the cell is a segment or an end-to-end Operations Administration and Maintenance (OAM) cell for a VCC. A specific PT coding indicates the presence of a Resource Management (RM) cell. Chapter 12 covers the usage of the AAL_indicate bit by AAL5. Part 5 covers the use of EFCI and resource management cells. Chapter 28 details OAM cell usage.

ATM-LAYER QOS AND SERVICE CATEGORIES

As you read about ATM, you will encounter the claim that ATM is the best communications networking technology to guarantee Quality of Service (QoS) and reserve bandwidth. The

PT Coding	Payload Type (PT) Meaning
000	User Data Cell, EFCI = 0, AAL_indicate = 0
001	User Data Cell, EFCI = 0, AAL_indicate = 1
010	User Data Cell, EFCI = 1, AAL_indicate = 0
011	User Data Cell, EFCI = 1, AAL_indicate = 1
100	OAM F5 segment associated cell
101	OAM F5 end-to-end associated cell
110	Resource Management cell
111	Reserved for future VC functions

EFCI = Explicit Forward Congestion Indication
AAL_indicate = ATM-layer-user-to-ATM-layer-user indication

Table 11-5. ATM Payload Type (PT) Encoding and Meaning

next few sections briefly introduce the terminology of ATM QoS parameters, service categories, and traffic parameters, leaving the details to Part 5.

Quality of Service (QoS) Parameters

The most commonly used ATM QoS parameters are

- ▼ Cell Transfer Delay (CTD)
- Cell Delay Variation (CDV)
- Cell Loss Ratio (CLR) (defined for CLP = 0 and CLP = 1 cells)
- Cell Error Ratio (CER)
- ▲ Cell Misinsertion Rate (CMR)

For all applications, the CER and the CMR must be extremely small, on the order of one in a billion or less. Therefore, the principal QoS parameters are delay (CTD), variation in delay (CDV), and loss ratio (CLR). To a large extent, human sensory perceptions determine the acceptable values of these major QoS parameters, while data communication protocol dynamics define the rest, as detailed in Chapter 20.

Service Categories and Traffic Parameters

We will use the term “ATM service category” many times, so now we briefly introduce these acronyms. They are an essential part of ATM parlance, and therefore we briefly

introduce them now, and Chapter 20 provides the rigorous definition of each service category. Each service category definition includes terms that define the traffic contract parameters and QoS characteristics. The ATM Forum Traffic Management 4.1 specification (abbreviated TM 4.1) defines the following ATM layer service categories:

- ▼ Constant Bit Rate (CBR)
- Real-time Variable Bit Rate (rt-VBR)
- Non-real-time Variable Bit Rate (nrt-VBR)
- Unspecified Bit Rate (UBR), optionally with a Minimum Desired Cell Rate (MDCR)
- Available Bit Rate (ABR)
- ▲ Guaranteed Frame Rate (GFR)

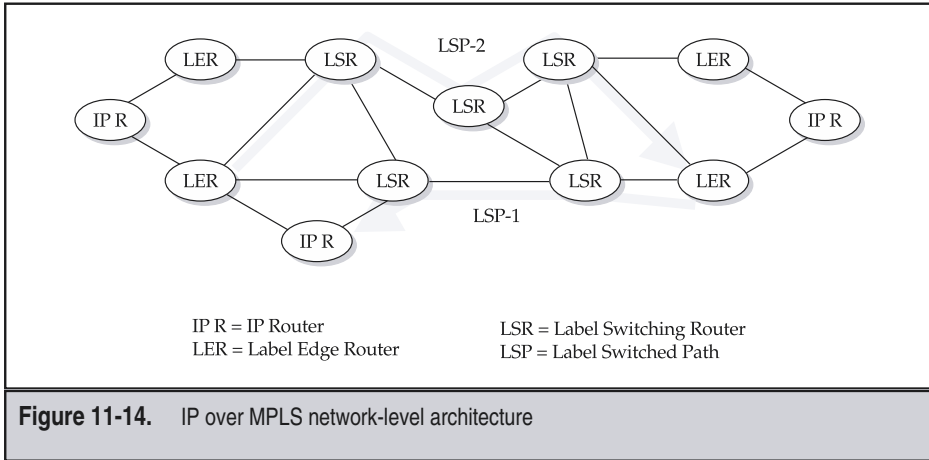
ITU-T Recommendation I.371 defines a similar set of set of attributes as ATM transfer capabilities, such as deterministic bit rate (DBR), statistical bit rate (SBR). Each of these service categories uses one or more traffic parameters, such as Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS). Chapter 20 defines these terms, while Chapter 21 describes how switches and end systems use them to shape and police traffic to meet these parameters.

MULTIPROTOCOL LABEL SWITCHING (MPLS)

As described in Chapter 10, Multiprotocol Label Switching (MPLS) began out of the motivation to improve traffic engineering in IP backbone networks and was the merger of several ideas. As stated in RFC 3031, however, the architecture of MPLS is extensible to support protocols other than IP. This section highlights some of the important aspects of the architecture and defines the major terminology and acronyms unique to MPLS. In this chapter, we study the forwarding functions. Chapter 14 covers the IETF standardized control functions of signaling and routing that establishes MPLS forwarding.

IP over MPLS Architecture and Terminology

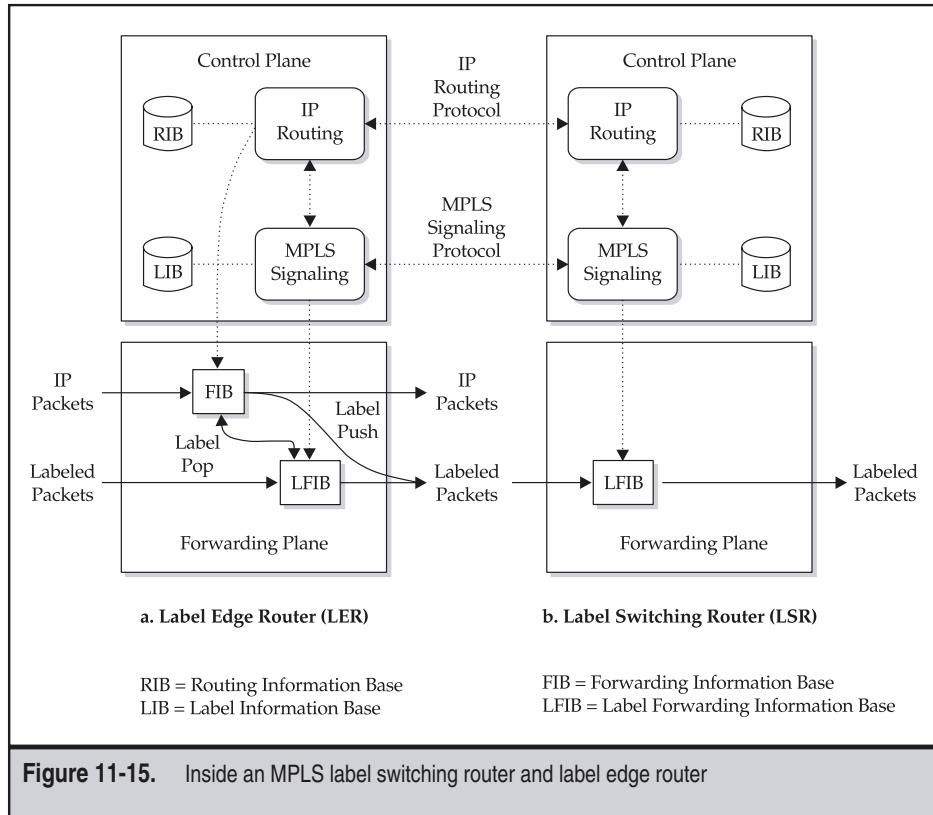
MPLS was designed from the ground up to support IP. Figure 11-14 illustrates a hybrid network containing traditional IP routers, along with MPLS-capable label edge routers and label switching routers, abbreviated LER and LSR. A label switched path (LSP) is a unidirectional connection that begins at an ingress LER and terminates at either an egress LER or a traditional IP router that is not MPLS capable. An LSP may follow the same path that a traditional IP router would choose (e.g., LSP-1), or it may be directed along an explicitly routed path (e.g., LSP-2) via a control signaling protocol. The ingress LER determines which IP packets are directed onto an LSP. An LSR only switches based upon only the label value; that is, it does not look at the IP header when making a forwarding decision. MPLS operates over essentially any possible link layer protocol that interconnects an LER or LSR. The next section further describes the functions of an LER and LSR.



MPLS Forwarding Operations

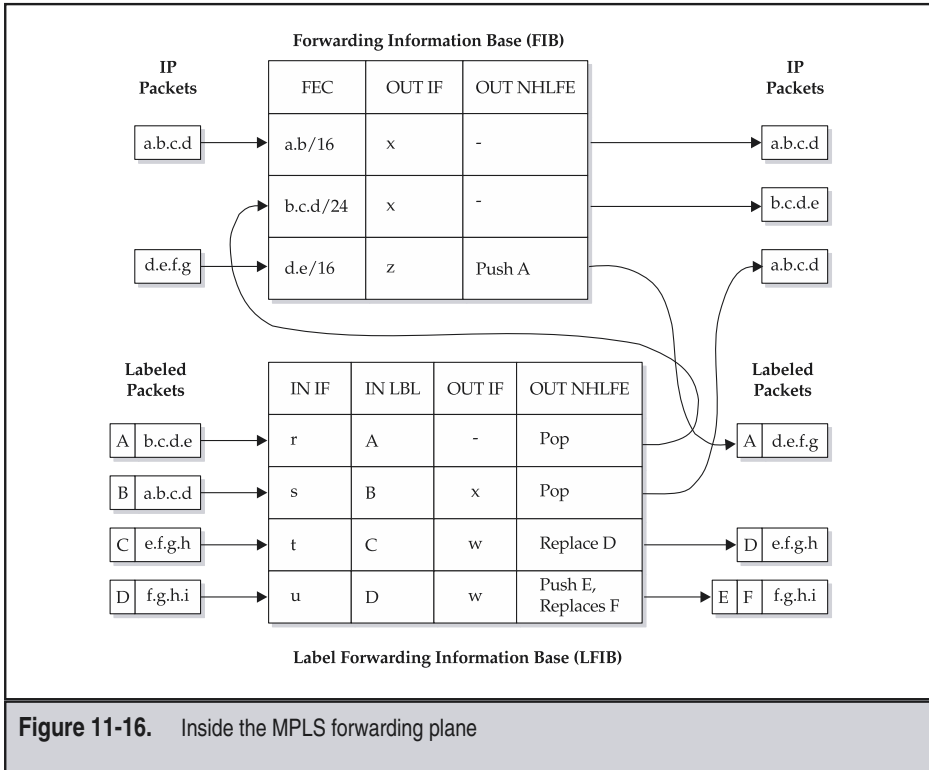
We now take a look at the inside of an LSR and a LER to better understand the operation of MPLS. Figure 11-15 presents a view of an LER and LSR in terms of control and forwarding plane components [Peplnjak01, MSF ARCH 1.0]. We briefly introduce the control plane here, and details provide in Chapter 14. The control plane and contains an IP routing function that interfaces with other LERs, LSRs, and traditional IP routers using a number of IP routing protocols. The result is a *routing information base (RIB)* populated with information describing the potential ways to reach particular IP address prefixes. An LER uses this information to populate a *forwarding information base (FIB)* in the forwarding plane. In a similar manner, the control plane also contains an MPLS signaling component that interfaces using an MPLS signaling protocol with LSRs. The result here is a *label information base (LIB)* that contains information about the association of label bindings negotiated with other MPLS-capable routers. The MPLS signaling component receives information from the IP routing function and the LIB to populate a *label forwarding information base (LFIB)* in the forwarding plane. An LER may forward IP packets, add a label to an IP packet (label push), or remove a label from an IP packet (label pop), while an LSR is only capable of forwarding labeled packets and pushing or popping labels (not shown in the figure). We now examine the forwarding plane of MPLS in greater detail, introducing the acronyms defined in MPLS standards.

RFC 3031 defines a number terms used in the description of IP over MPLS. We summarize the definitions for these terms and put them into the context of the forwarding plane concept introduced previously with reference to the exploded view of the FIB and LFIB shown in Figure 11-16. A *forwarding equivalence class (FEC)* is a subset of packets based upon some of the information in the IP header used by the FIB. A commonly used FEC is that of a longest prefix match on an IP destination address, denoted in this book



using the classless interdomain routing (CIDR) prefix style defined in Chapter 8. For example, the high-order 16 bits matching dotted-decimal IP addresses of the form "a.b.*.*" (where * indicates any possible legal value) is denoted as "a.b/16" for the first FEC entry in the FIB. Note that FECs can be based upon additional or other IP header fields, for example, TOS/Diffserv byte values. The FIB uses the FEC to determine the next hop outgoing interface for IP packets, much as a traditional router does (e.g., as done for IP address prefix "a.b/16" in the example).

The *next hop label forwarding entry (NHLFE)* is used by the LFIB when forwarding a labeled packet or by the FIB in an LER when pushing a label onto an IP packet. The NHLFE may also contain information about the link layer encapsulation, label stack encoding, and other link-level-specific details. The NHLFE must contain the next hop outgoing



interface along with an indication that one of the following operations is to be performed on the labeled packet:

- ▼ Replace the topmost label with a new label
- Pop the topmost label
- ▲ Replace the topmost label with a new label and push another label onto the label stack

Examples of each of these operations are given in the LFIB in Figure 11-16. The *incoming label map (ILM)* part of the LFIB operates on a labeled packet and maps an incoming label to a set of NHLFEs. This is represented in the figure by the columns labeled IN IF and IN LBL, but could also be a separate table per interface in an implementation. The *FEC-to-NHLFE (FTN)* of a FIB maps the FEC derived from an incoming IP packet to a set of one or more NHLFEs. As an example in the figure, the label A is pushed onto IP packets

with the FEC d.e/16. Note that an ILM or FTN may map to more than one NHLFE—for example, for use in load balancing across multiple equal-cost LSPs [RFC 3031].

An important optimization of MPLS support of IP is avoiding the label lookup processing at the egress LER in the case of a packet arriving on an LSP for that requires a subsequent IP lookup. This is shown in Figure 11-16 for a packet arriving with label A that is popped and then directed to the FIB for a lookup based upon the IP header. To avoid this additional processing, MPLS defines a process called *penultimate hop popping*, where the penultimate (i.e., next to last) router in the LSP pops the label instead of requiring the last router to do so. This reduces the processing required at the last router and results in the same forwarding behavior, as the example in the next section demonstrates.

Example of MPLS Forwarding of IP Packets

The example in this section puts the concepts of label edge and switching routers (i.e., LER and LSR) along with the forwarding plane FIB and LFIB components detailed in the previous section in a network context with reference to Figure 11-17. The example contains LERs on the left and right along with LSRs in the center of the figure. The example shows the path and forwarding actions taken at each node for two LSPs: LSP-1 and LSP-2. We trace the operation of LSP-1 in the following discussion, beginning at LER E1 in the upper left-hand corner of the figure where an IP packet with destination address (DA) a.b.c.d arrives. LER E1 consults its FIB and determines that this packet belongs to FEC a.b.c/24, and outputs it on interface 2 after pushing on label A. Next, LSR S1 sees the packet with label A arrive on interface 1, and its LFIB indicates that the packet should be output on interface 4 and the label should be replaced with label D. Interface 4 on LSR S1 connects to interface 1 on LSR S4, which carries the labeled packet. Since LSR S4 is the penultimate hop for LSP-1, the operation specified in its LFIB is to pop the label and send the packet on outgoing interface 4. Finally, at the destination LER E4, the FIB entry operates on the FEC a.b.c/24 and delivers the packet to outgoing next hop interface 3. The forwarding plane entries are also shown in the example for LSP-2.

MPLS ENCAPSULATION STANDARDS

Because MPLS uses the principle pioneered in ISDN of separating control from forwarding (see Chapter 6), it is capable of operation over a number of different link layer protocols. This means that MPLS can be implemented as an integrated router plus switch, or it can be an existing switch supporting a technology like Frame Relay, ATM, or even Ethernet. This section describes the generic MPLS shim header encapsulation and the MPLS over ATM and Frame Relay standards to provide detail on the operation of MPLS.

MPLS Shim Header

What the heck is a label, you ask? RFC 3032 from the IETF's MPLS working group defines the contents of a four-byte MPLS shim header, as shown in Figure 11-18. The MPLS

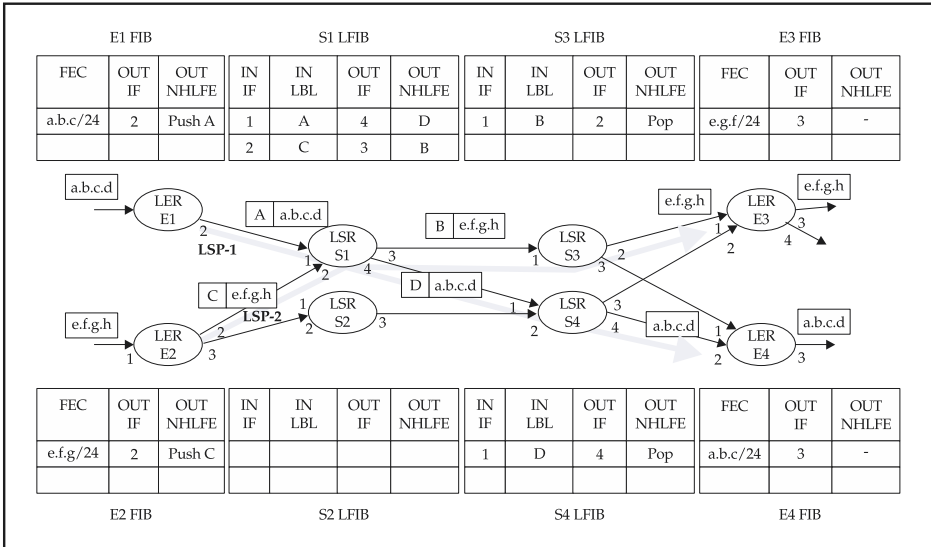


Figure 11-17. Example of MPLS forwarding of IP packets

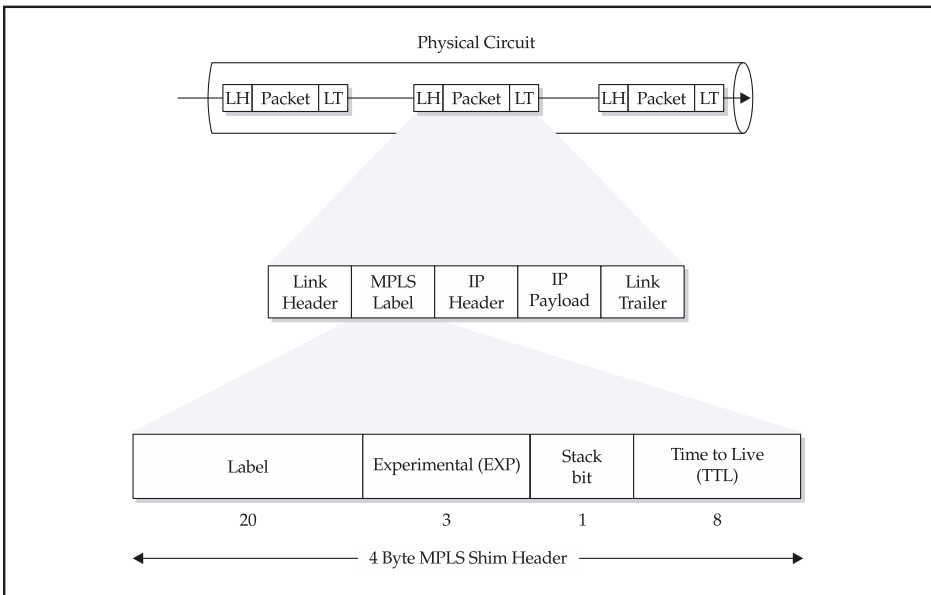


Figure 11-18. Multiprotocol Label Switching (MPLS) shim header

header is “shimmed” between the link layer protocol, (e.g., Point-to-Point protocol [PPP] or Ethernet) and the network layer (e.g., IP), and hence is the basis for the name “shim header.” When a link layer has an identifier, such as the ATM VPI/VCI or the Frame Relay DLCI, the MPLS shim header may not be necessary. This section gives more details on these MPLS encapsulations later. Since the efficiency in transporting is an important business criterion in selection of networking technology, this chapter began a tally of the overhead involved when MPLS is transmitted over a number of popular link layer encapsulations. After describing other related information, Part 8 concludes with a comparison of protocol efficiency in support of IP.

The contents of the label give some insight into the capabilities of MPLS. The 20-bit label value defines the index used in the forwarding table. Earlier drafts of the MPLS work defined the 3-bit experimental (EXP) field as a Class of Service (COS) field, as it was called in Cisco tag switching, and also defined an indication of congestion. The EXP field may be a function of the IP Type Of Service (TOS) or Diffserv field, as described in Chapter 20. It may also be a function of the packet’s input interface, the flow type, or support other traffic management functions, like conformance marking, discard priority indication, or congestion indication.

MPLS shows its support of IP by using the same 8-bit time to live (TTL) field as defined in the IP packet header in Chapter 8. The TTL field provides a means for loop detection and limiting the amount of time that a packet can spend in the network. In most cases, an LSR along the LSP should decrement the TTL for each hop so that, if a routing loop is encountered, the TTL will eventually expire and looping packets will be discarded. Since some link layer networks do not support TTL (e.g., Frame Relay and ATM), the LSR at the ingress to such a network should decrement the TTL by the number of hops across that network. If the TTL is less than the number of hops across the TTL-unaware link layer network, then the ingress LSR should discard the packet. There are other cases where an LSR may not decrement TTL at each hop; for example, a service provider may wish to hide information about parts of its topology.

The MPLS architecture allows a hierarchy of labels, which is similar to ATM’s use of the VPI and VCI to create the hierarchy of virtual channels within virtual paths. Since ATM has only two levels of label, it can only support two levels of hierarchy; but in MPLS, the number of label stack entries in the shim header can be significantly greater, and therefore the levels of hierarchy can also be greater. The practical limitation to the number of stacked labels is the Maximum Transfer Unit (MTU) of the link layer protocols used along an LSP. If the packet becomes too long through the addition of too many labels, then it must be fragmented at the IP layer. Since IP fragmentation is resource intensive, RFC 3032 describes a procedure that strives to perform fragmentation only once.

The MPLS label stack operates in a last-in, first-out manner. That is, the last label pushed onto a stack with $m - 1$ entries becomes the topmost, or level $- m$ label, and any other labels decrease in level by one. The bottom label in the stack is called level $- 1$. When a packet has multiple label stack entries, LSRs use only the topmost label in forwarding decisions. When an LSR pops the top-level label, and that label is not the bottom of the stack, another label is exposed, which then becomes the new top-level label. Because a packet may have multiple labels, the stack (S) bit indicates the last label prior to the packet header.

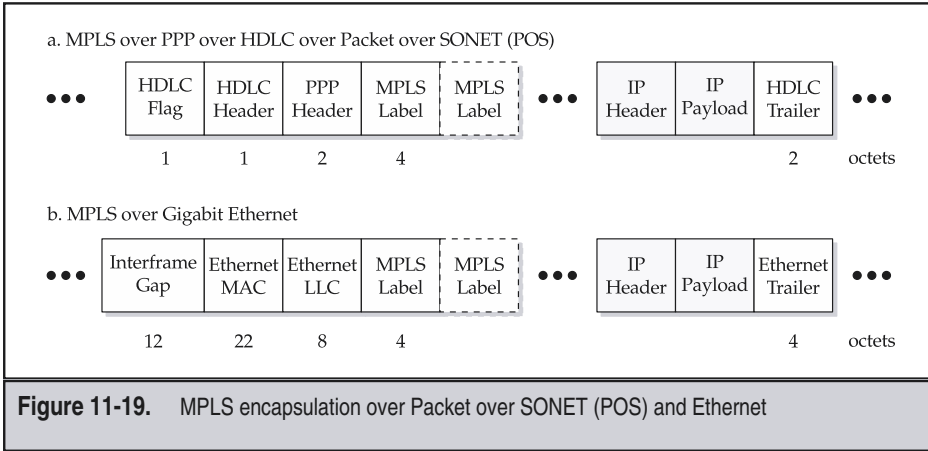
The concept of label stacking currently has at least three unique applications, and certainly more are possible. These applications are detailed later in the book, but we briefly introduce them with pointers to further detail. One application is in traffic engineering when multiple smaller flows are aggregated into a single larger flow. It is sometimes simpler to traffic-engineer a smaller number of such larger aggregated flows than a large number of small flows, as described in Chapter 21. Another application is to use the topmost label to identify a destination router and use the bottom label to identify a particular IP virtual private network (VPN) customer routing table, an approach described in Chapter 19. Furthermore, the topmost label can be used for forwarding and the bottom label can be used to indicate another service that is being carried. We introduce this concept in Chapter 12, and summarize the potential use and adoption of this approach in Part 4. Finally, as described in RFC 3031, another potential application of label stacking could allow service providers to tunnel traffic across each other's networks.

We now give a few examples of MPLS encapsulation over popular link layer protocols. Figure 11-19a illustrates the frame layout for MPLS operation over the point-to-point protocol (PPP) over HDLC over Packet over SONET (POS). Each HDLC frame has at least an opening flag, a short header and a 2-byte trailer that performs error detection, as described in Chapter 7. Chapter 8 described the operation of PPP over POS. The figure illustrates the 2-byte PPP header, which indicates the protocol type, that has a value of 0x281 and 0x283 for MPLS unicast and multicast packets, respectively. There is also a few percent of overhead present due to the octet stuffing method employed by PPP used in POS to prevent occurrence of an HDLC-flag with the HDLC frame. Since HDLC and PPP/POS comprise the link layer, one or more label stack entries in MPLS shim headers precede the IP header. As can be seen from the figure, the minimum overhead for MPLS/PPP/POS for each IP packet is 10 bytes.

Figure 11-19b illustrates the frame layout for MPLS operation over gigabit Ethernet (see Chapter 9). First, each gigabit Ethernet has a 96-bit (i.e., 12-byte) minimum interframe gap. The 22-byte Ethernet Medium Access Control (MAC) contains an 8-byte preamble, along with the 6-byte Ethernet source and destination addresses and a 2-byte length field. The logical link control (LLC) field contains the 1-byte source and destination Service Access Points (SAPs) and the 3-byte subnetwork access point (SNAP) field, followed by a 1-byte control field and then a 2-byte Ethertype field that supports multiple protocols between the same pair of SAPs. The Ethertype field is 0x8847 and 0x8848 for MPLS unicast and multicast packets, respectively. Finally, Ethernet has a 4-byte trailer used for the purpose of error detection. Since the Ethernet interframe gap, MAC, and LLC comprise the link layer, one or more MPLS label stack entries are added to the shim header and precede the IP header. As can be seen from the figure, the minimum overhead for MPLS over Gigabit Ethernet for each IP packet is 48 octets.

MPLS over ATM

RFC 3035 defines the operation of MPLS over native ATM switches. One of the initial uses of MPLS was within ISP backbones, replacing ATM as a more efficient, IP-aware, traffic-engineering-capable underlay [RFC 3031]. ATM provides all MPLS capabilities



except for the TTL field with two levels of label stacking. However, since ATM is connection oriented, routing loops don't occur in proper implementations. RFC 3035 does specify an optional path vector procedure to detect loops for MPLS over ATM networks. Furthermore, several efforts in standards bodies are also underway to make MPLS-powered networks interwork with ATM-based networks .

Figure 11-20 shows the support for MPLS over ATM adaptation layer 5 (AAL5). Starting at the bottom of the figure, the AAL5 protocol data unit (PDU) is followed by zero or more MPLS label stack entries in the shim header. As described in RFC 3035, the shim header is not necessary if there is knowledge that label stacking will never be used, and that TTL and EXP processing are not needed. This is an unusual case, and, normally, a null MPLS shim header is present. A shim header with a null value as the top-level label is necessary when label stacking is in use, because it is necessary to determine whether the stack is empty. In this case, the ATM VPI/VCI contains the label value, and the shim header label field contains a value of 0 (indicating null), along with the EXP field, the stack bit, and the TTL. After the IP header and payload, there is a padding field to fill out the payload of the last ATM cell of the segmented AAL5 PDU, followed by an 8-byte AAL5 trailer that supports error detection and some other functions.

As shown at the top of Figure 11-20, typically several ATM cells are needed to carry the contents of an AAL5 PDU. Note that ATM switches operate only on the cell headers, and not on the AAL5 PDU. Since most ATM switches do not support the VC merge function, described in Chapter 10, necessary to support a multipoint-to-point configuration, ATM-based LSPs typically only support a point-to-point configuration. The VPI and VCI in the ATM cell header make up the topmost label. If the ATM layer provides a virtual path connection between LSRs, then only the VCI is the top-level label. Finally, as detailed in the next chapter, the AAL indication bit in the ATM header indicates the last cell in the AAL5 PDU to indicate to the receiver that it can begin the reassembly operation.

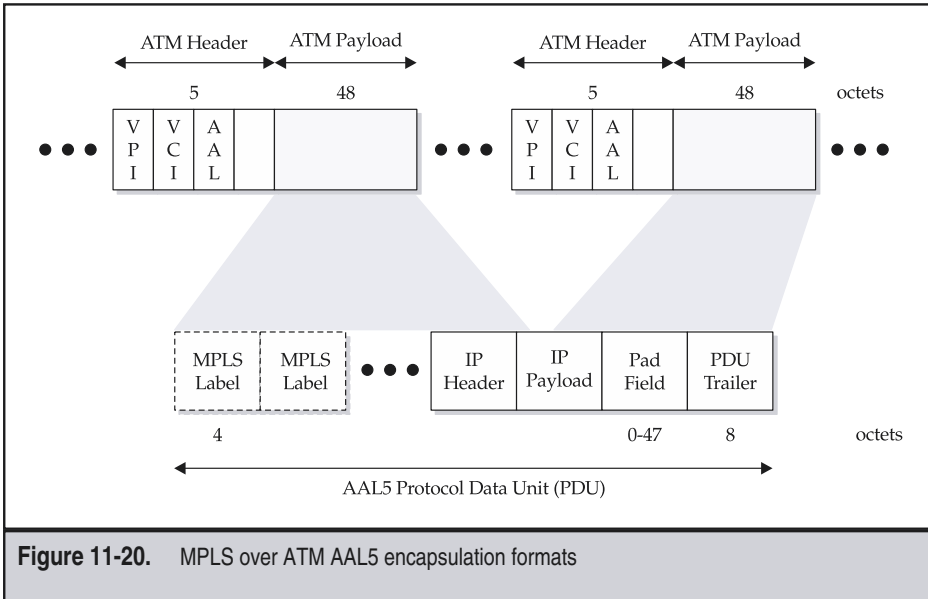


Figure 11-20. MPLS over ATM AAL5 encapsulation formats

Note that control protocol information between ATM-based LSRs is normally sent on a separate VCC, as described in RFC 3035.

The precise analysis of the overhead involved in transporting variable-length packets over AAL5 is somewhat complicated because of the factor for round up to fill the last cell. We defer the details to Part 8, after detailing the components of AAL5 in Chapter 12 and multiprotocol encapsulation in Chapter 18. The principal result there is that MPLS/PPP/POS is approximately 15 percent more efficient than ATM and AAL5 when carrying typical IP traffic. However, in networks with an in-place ATM infrastructure that also supports multiple other services, IP over ATM can still make good business sense.

MPLS over Frame Relay

RFC 3034 defines the operation of MPLS over native Frame Relay (FR) networks, which are used by many enterprises and some service providers to carry IP traffic. Like ATM, FR supports the label switching function of MPLS capabilities except for the TTL field and the stack bit. If FR operates over ATM, or uses some other form of connection-oriented implementation, then routing loops should not occur in proper implementations. Since FR continues to be a popular service (despite many premature reports of its demise), this encapsulation is important.

Figure 11-21 shows the support for MPLS by FR when using a 10-bit data link connection identifier (DLCI) in the FR header. As described in Chapter 7, FR operates over HDLC, so

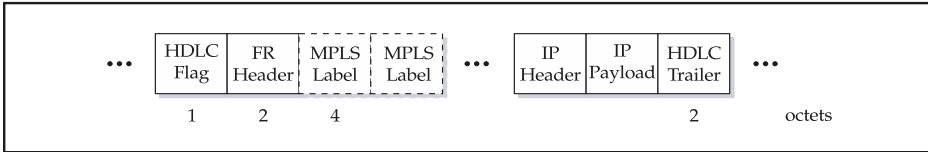


Figure 11-21. MPLS over FR encapsulation formats

the header and trailer are identical to that of PPP. As in ATM, if the FR LSRs are configured such that label stacking never occurs and TTL and EXP processing are not needed, then an MPLS shim header may not be necessary; otherwise, it must be present to implement these functions. In a manner similar to ATM, the FR DLCI carries the top-level label on which the FR switches operate, while the topmost entry in the MPLS shim header carries the values of the EXP, S, and TTL fields. Also, like ATM, a separate DLCI is necessary for operation of the control protocol between LSRs.

However, unlike ATM, some FR LSRs can support LSP merging, since the protocol is frame based. As seen from Figure 11-21, the operation of MPLS over FR is quite efficient, requiring as little as 9 bytes of overhead for carriage of IP over MPLS/FR. However, the internal network functions of traffic management implemented in ATM are not as widely supported in FR switching networks. In fact, many FR networks are implemented over ATM and AAL5, as described in Chapter 16, and therefore the operation is less efficient than that for a pure FR-only implementation.

REVIEW

This chapter first summarized the logical functions of the physical and ATM layers and how these relate to higher layers. It then introduced the reader to the basic building blocks of ATM: the physical transmission interfaces, the Virtual Path (VP), and the Virtual Channel (VC), including examples of end system and switching functions. We then looked at the ATM cell structure in detail, since the information contained in the ATM header is the basis for all ATM-related functions. The chapter then described MPLS-specific terminology and the general architecture of a separate control and forwarding plane. It then provided a description of the MPLS forwarding plane, with the companion control plane discussion contained in Chapter 14. We then described the operation of MPLS over various physical and logical media. Some of these approaches built upon tried-and-true link layer protocols, such as HDLC, PPP, and Ethernet, described in Chapters 7, 8, and 9, respectively.

CHAPTER 12



ATM Adaptation and MPLS Tunneling Protocols

This chapter covers the set of generic protocols between the ATM or MPLS link layer model and the transport or application layer. As introduced in Chapter 10, B-ISDN has a formal layered model for this function, called the ATM Adaptation Layer (AAL). The ITU-T has defined four AALs, and we cover all of them here. At the time of writing, the support for layered protocols over MPLS was an emerging and active area in the standards bodies. Since these functions are not yet fully standardized, we provide only a higher-level summary of the basic concepts and functions, providing pointers to the standards groups actively engaged in further defining these aspects.

ATM ADAPTATION LAYER (AAL)

The ATM Adaptation Layer (AAL) provides support for higher-layer services such as signaling, circuit emulation, voice, and video. AALs also support packet-based services, such as IP, LANs, and Frame Relay. First, this chapter introduces the initial ITU-T notion of AAL service classes and applies this to real-world applications. Next, the text covers the AAL generic layered structure. The chapter then delves into each ATM Adaptation Layer, describing the basic characteristics, formats, and procedures. The text gives several examples to further clarify the operation of each AAL. The remaining chapters in this part, as well as the coverage in Part 4 regarding the higher layers of the user and control planes, rest upon this foundation of AALs.

ATM Adaptation Layer (AAL)—Protocol Model

The I.363 series of ITU-T Recommendations define the next higher layer of the B-ISDN protocol stack—the AAL. Next, the text describes the generic AAL protocol model, which consists of a Segmentation and Reassembly (SAR) sublayer, along with Common Part and Service-Specific Convergence Sublayers (CPCS and SSCS). The chapter then describes the Common Part (CP) format and protocol for each standardized AAL, illustrated by example applications.

In 1993, ITU-T Recommendation I.362 [ITU I.362] defined the basic principles and classification of AAL functions. Although it is no longer in force, some concepts and terminology initially defined in I.362 are still used in the ATM Forum documents and ITU-T standards. The attributes of the service class are the timing relationship required between the source and destination, whether the bit rate is constant or variable, and whether the connection mode is connection oriented or connectionless. The AAL service class is a separate concept from the ATM layer's service category and Quality of Service (QoS) introduced in Chapter 11 and detailed in Part 5. Chapter 15 describes how the (AAL) service class (or bearer capability), service category, and QoS class (or, optionally, explicit QoS parameters) can all be signaled separately in an SVC call setup message. I.362 labeled the AAL service class terminology as A through D and envisioned that these would map to four distinct AALs. What has survived is the definition of following three bearer class definitions in the broadband bearer capability information element used in signaling messages, as follows:

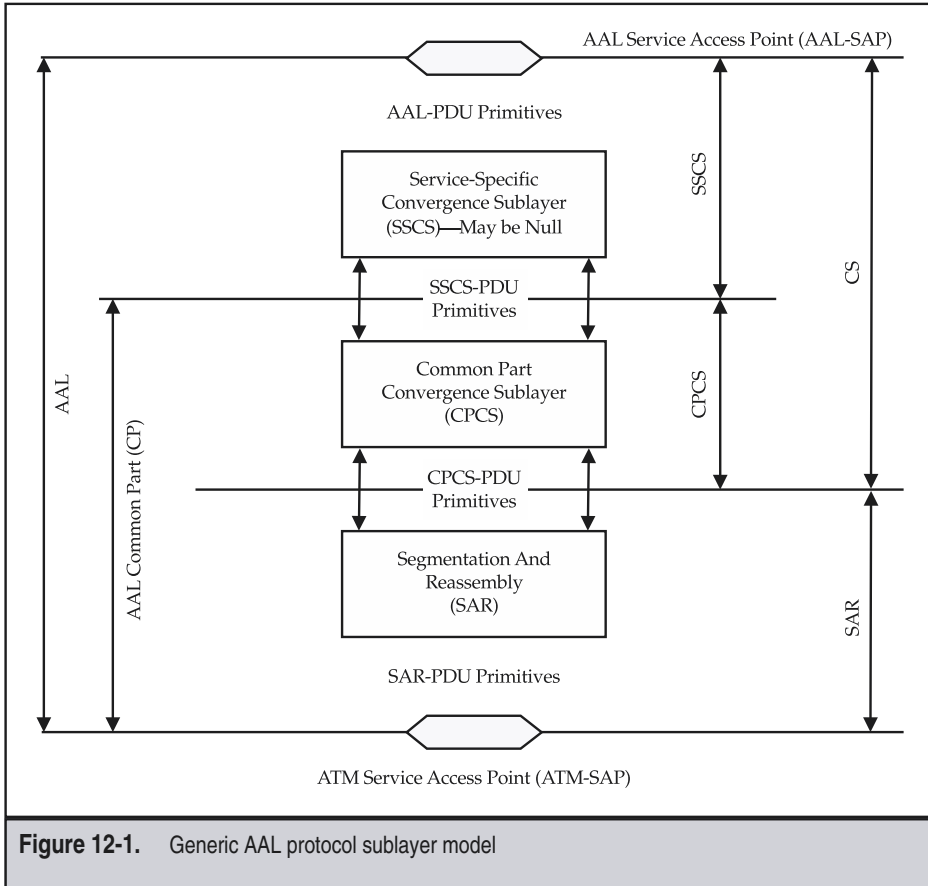
- ▼ **Bearer Class A** is a connection-oriented, constant bit rate service, where interworking based upon AAL information may be performed.
- **Bearer Class C** is a connection-oriented, variable bit rate service, where interworking based upon AAL information may be performed.
- ▲ **Bearer Class X** is an ATM service where AAL, traffic type, and timing requirements are transparent to the network.

The history of AAL development altered the direct approach of a separate AAL for each bearer class. Initially, the ITU-T targeted AAL3 for connection-oriented services and AAL4 for connectionless services. However, experts realized that AAL3 and AAL4 had much in common and merged them into AAL3/4, which the IEEE 802.6 standard adopted and the industry applied to SMDS, as described in Chapter 8. Soon thereafter, the computer industry conceived of AAL5 in response to perceived complexity and implementation difficulties in AAL3/4. When initially proposed in 1991, AAL5 was called the Simple Efficient Adaptation Layer (SEAL) for these reasons [Lyons 91]. The ATM Forum, ANSI, and the ITU-T adopted AAL5 in a relatively short time; and since then, AAL5 has become the AAL of choice for data communications. For example, as detailed in Part 4, AAL5 carries IP, Ethernet, Frame Relay, and video. The ITU-T initially defined AAL1 to interwork directly with legacy TDM networks and N-ISDN. The standards bodies completed the base AAL2 standard in 1997 to support variable bit rate voice and video applications more efficiently than the constant bit rate AAL1.

AAL Protocol Structure Defined

The B-ISDN protocol model adapts the services provided by the ATM layer to those required by the higher layers through the AAL. Figure 12-1 depicts the structure and logical interfaces for AAL1, AAL3/4, and AAL5, as defined in the I.363.x series of recommendations. AAL2 has a different structure, as described later. At the top of the figure, an AAL Service Access Point (SAP) provides services to higher layers by passing primitives (e.g., request, indicate, response, and confirm) concerning the AAL Protocol Data Units (AAL-PDUs). Subsequent sections summarize the resulting transfer of PDUs between sublayers and across the AAL-SAP from a functional point of view. See the standards referenced in the following sections for details on the protocol primitives for each AAL.

Standards further subdivide the Common Part (CP) for AAL1, AAL3/4, and AAL5 into the Convergence Sublayer (CS) and the Segmentation and Reassembly (SAR) sublayer as shown at the bottom of Figure 12-1. The CS layer contains Service-Specific (SS) and Common Part (CP) sublayers, as indicated in the figure. The SSSS may be null, which means it does nothing; or, as the name implies, when it is not null, it provides particular services to the higher-layer AAL user. Chapter 13 covers the signaling SSSS, while Chapter 17 covers the FR-SSCS. The CPCS must always be implemented along with the SAR sublayer. These layers pass primitives regarding their respective PDUs among themselves as labeled in Figure 12-1, resulting in the passing of SAR-PDU primitives (which is the ATM cell payload) to and from the ATM layer via the ATM-SAP.



Key AAL Attributes

Why did the standards bodies define so many AALs when everything ends up packed into the 48-byte payload of ATM cells anyway? The answer lies in differences between key attributes supported by each AAL as summarized in Table 12-1. First, the various AALs have vastly different PDU lengths as seen at the AAL-SAP. AAL1 and AAL2 have shorter PDU lengths to support real-time services such as circuit emulation, voice, and video. On the other hand, AAL3/4 and AAL5 support traditional packet data services by carrying anything ranging from a minimal 1-octet size up to a jumbo size of 65,535 octets. Another

Attribute	AAL1	AAL2	AAL3/4	AAL5
AAL-PDU size range (octets)	46–47	1–64	1–65,535	1–65,535
Multiple logical channels per VCC	No	Yes	Yes	No
User-to-User Indication (UUI)	No	Yes	No	Yes

Table 12-1. ATM Adaptation Layer (AAL) Attributes

key AAL attribute is support for multiple logical channels over a single ATM VCC. AAL2 does this to support packetized voice and video, and to reduce packetization delay and improve packet fill efficiency. AAL3/4 does this to make more efficient use of VCCs and reduce delay variation encountered by individual packets. Finally, some AALs support a User-to-User Indication (UUI) information transfer capability. Currently, standards don't exist for these fields; however, they offer an extensibility that other AALs don't.

The following sections define the sublayers and characteristics for each of the currently standardized AALs:

- ▼ **AAL1** Constant bit rate, real-time traffic
- **AAL2** Variable bit rate, real-time traffic
- **AAL3/4** Variable bit rate data traffic
- ▲ **AAL5** Lightweight variable bit rate traffic

Each section then provides one or more examples illustrating the application of the ATM Adaptation Layer to a real world application.

ATM ADAPTATION LAYER 1 (AAL1)

As stated in ITU-T Recommendation I.363.1, the AAL1 protocol specifies the means to:

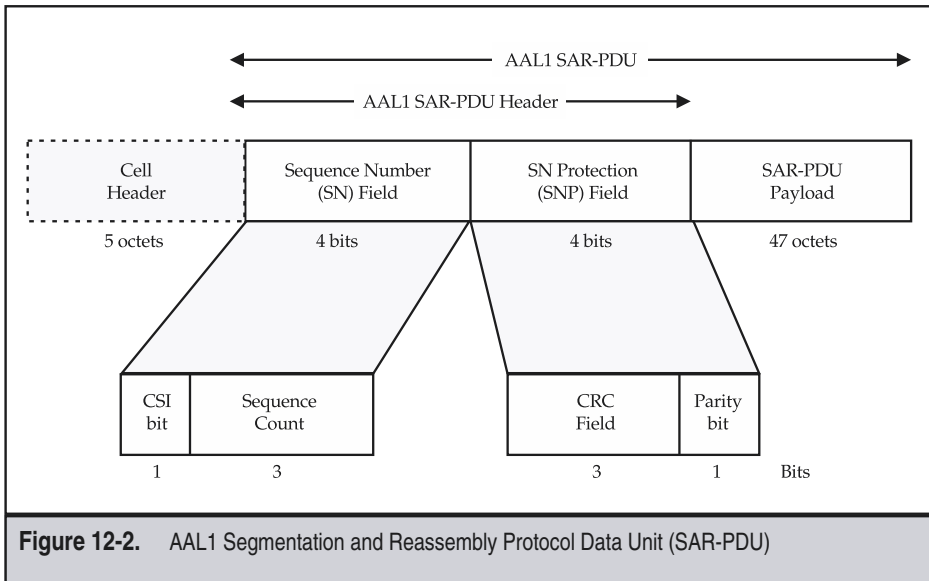
- ▼ Transfer service data units received from a source at a constant source bit rate and then deliver them at the same bit rate to the destination
- Optionally transfer timing information between source and destination
- Optionally transfer TDM structure information between source and destination
- Optionally perform Forward Error Correction (FEC) on the transferred data
- ▲ Optionally indicate the status of lost or erroneous information

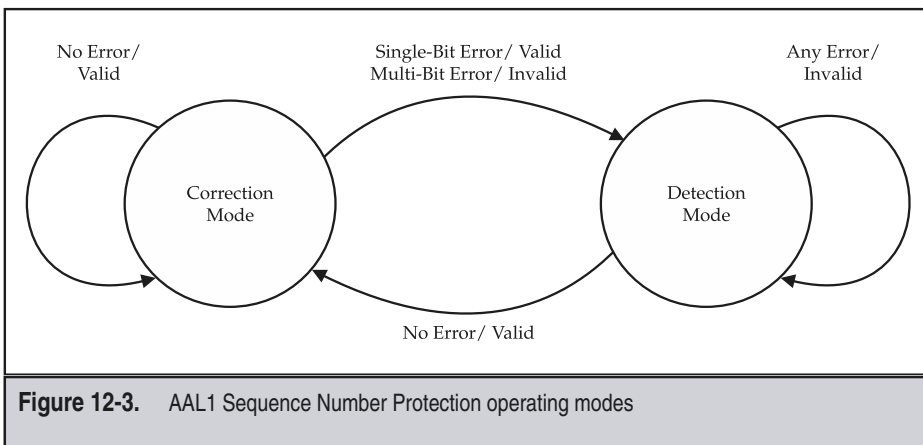
AAL1 Segmentation and Reassembly (SAR) Sublayer

The AAL1 SAR sublayer provides the following services:

- ▼ Maps between the 47-octet CS-PDU and the 48-octet SAR-PDU using a 1-octet SAR-PDU header
- Indicates the existence of CS function using a bit in the SAR-PDU header
- Generates sequence numbering for SAR-PDUs at the source and validate received sequence numbers at the destination before passing them to the CS sublayer
- ▲ Performs error detection and correction on the Sequence Number (SN) field

Figure 12-2 depicts the SAR-PDU for AAL1. Since the AAL 1 SAR uses 1 octet, 47 octets remain for user data in the SAR-PDU payload. The AAL1 SAR header has two major fields: the Sequence Number (SN) field and the Sequence Number Protection (SNP) field, as indicated in the figure. Within the SN field, the origin increments the 3-bit sequence count sequentially. The receiver checks for missing or out-of-sequence SAR-PDUs, generating a signal alarm when this occurs. AAL1 CS protocols utilize the Convergence Sublayer Indication (CSI) bit for specific functions, described later in this section. The 3-bit CRC field computes a checksum across the 4-bit SN field. As further protection against errors, the parity bit represents even parity across the first 7 bits in the 1-octet SAR-PDU header.





The sequence number is critical to proper operation of AAL1, since an out-of-sequence or missing SAR-PDU disrupts at least 47 octets of the emulated circuit's bit stream. Standards define a detailed procedure to correct many problems due to bit errors in the Sequence Number field or to accurately detect uncorrected errors. The state machine of Figure 12-3 illustrates operation at the receiver. While in the correction mode, the receiver corrects single-bit errors in the SAR header using the CRC. If after CRC correction the parity check fails, then the receiver switches to detection mode, since an uncorrectable multiple-bit error has occurred. The receiver stays in detection mode until no error is detected and the sequence number is sequential again (i.e., valid).

AAL1 Convergence Sublayer Functions

The AAL1 Convergence Sublayer (CS) defines the following functions in support of the transport of TDM circuits, video signals, voice band signals, and high-quality audio signals:

- ▼ Blocking and deblocking of user information to and from 47-octet SAR-PDU payloads
- Handling cell delay variation for delivery of AAL-SDUs at a constant bit rate
- Partial fill of the SAR-PDU payload to reduce packetization delay
- Optional processing of the sequence count and its error check status provided by the SAR sublayer to detect lost and misinserted cells
- Synchronous recovery of source clock frequency at the destination end using the Synchronous Residual Time Stamp (SRTS) method
- Transfer of TDM structure information between source and destination using Structured Data Transfer (SDT)

- Usage of the CS Indication (CSI) bit provided by the SAR sublayer to support specific CS functions, such as SRTS and SDT
- Asynchronous recovery of the TDM clock at the destination using only the received cell stream interarrival times or playback buffer fill
- Optional forward error correction combined with interleaving of the AAL user bit stream to protect against bit errors
- ▲ Optional generation of reports on the end-to-end performance deduced by the AAL based on lost and misinserted cells, buffer underflow and overflow, and bit error events

These AAL1 CS functions provide a menu that higher-layer applications use to provide required service features (e.g., timing recovery), deliver end-to-end performance (e.g., loss and delay), and account for anticipated network impairments (e.g., cell loss and delay variation). Table 12-2 illustrates some examples of this selection of AAL1 CS menu items by higher-layer services [ITU I.363.1], such as Circuit Emulation Service (CES), voice, and video. The following sections give details for use of the SDT pointer, the unstructured mode, and the clock recovery CS functions.

AAL1 CS Function	Structured TDM CES	Unstructured TDM CES	Video Signals	Voice Band Signals
CBR rate	nx64 Kbps	PDH rate [*]	MPEG2 rate	64 Kbps
Clock recovery	Synch	Synch, asynch	Asynch	Synch
Error correction	Not used	Not used	Used	Not used
Error status	Not used	Not used	Used	Not used
SDT pointer	Used [†]	Not used	Not used	Not used
Partial cell fill	Optional	Not used	Not used	Not used

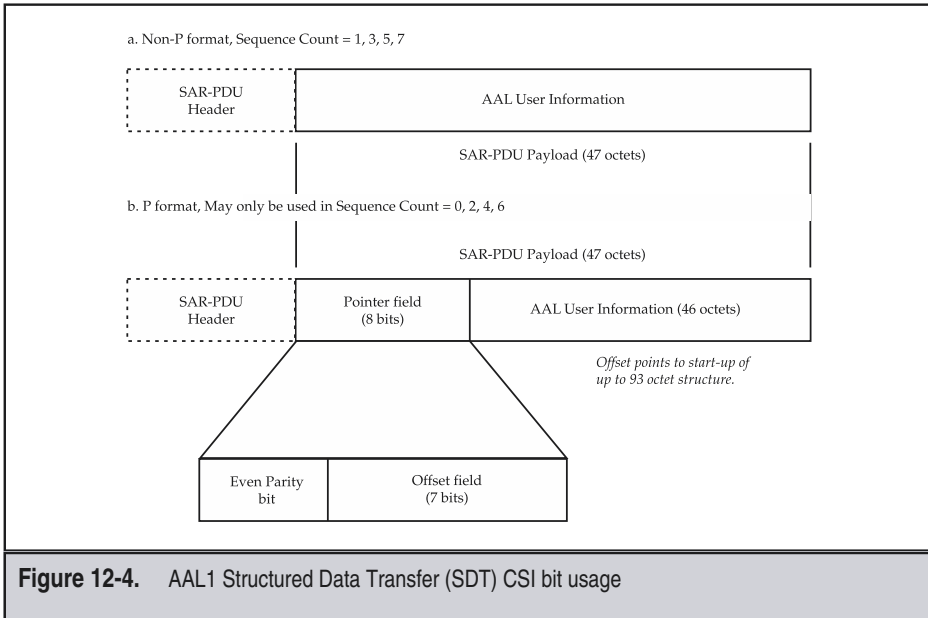
Notes:
^{*} For example, 1.544, 2.048, 6.312, 8.448, 34.368, or 44.736 Mbps.
[†]Pointer not necessary for n = 1 (64 Kbps).

Table 12-2. Application of AAL1 CS Functions to Transport of Specific Services

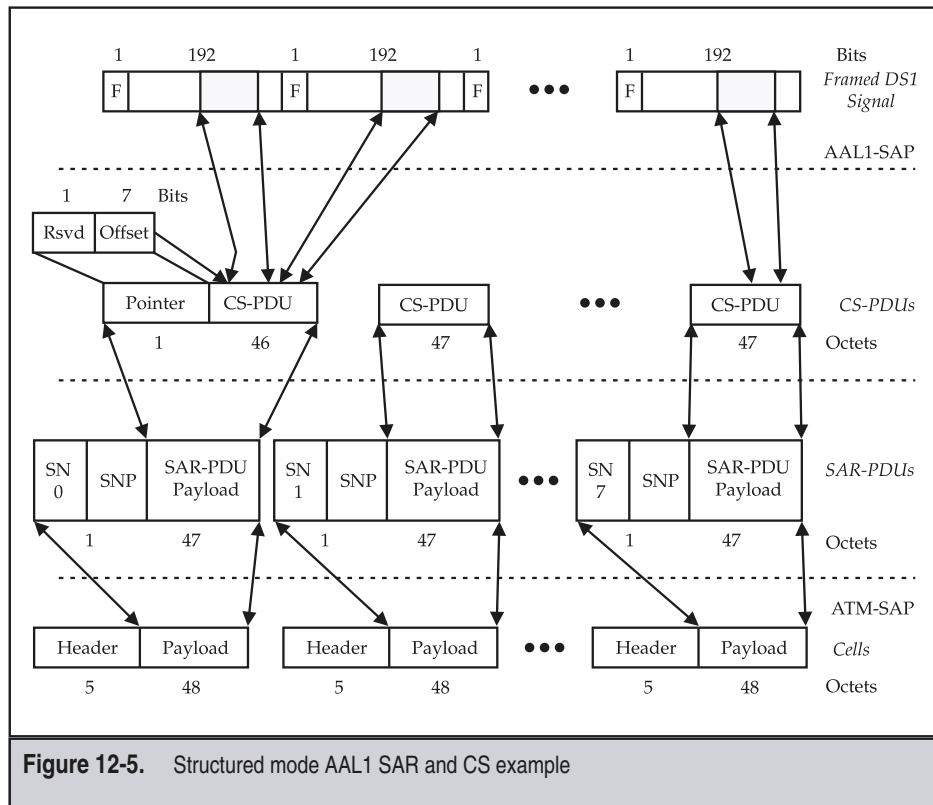
Structured Data Transfer (SDT) Convergence Sublayer

The SDT CS utilizes a pointer field in even-numbered SAR-PDUs once every eight SAR-PDUs (i.e., once within the wraparound interval of the 3-bit sequence number field) to communicate the beginning of the structure boundary. Most SAR-PDUs use the non-P format, as shown in Figure 12-4a.

The SDT CS supports octet structures ranging from 2 to 93 octets at 8 kHz. The CSI bit in the AAL1 SAR PDU indicates when the pointer field is present in an even-sequence-numbered SAR-PDU, called a P-format SAR-PDU, as illustrated in Figure 12-4b. This means that CS-PDUs carry only 46 octets of user data when the pointer is present (p-Format), as compared with the 47 octets when the pointer is absent (non-P format). According to ITU-T Recommendation I.363.1, the pointer field should be used at the first opportunity—that is, sequence number 0, as shown in the example that follows. The SDT CS uses the pointer field to reconstruct the precise 64 Kbps time slot alignment at the destination, since the time slot octets generally do not coincide with the CS-PDU boundaries. The 7-bit offset within the pointer field in the CS performs this function by pointing to the first octet of the structure. The octets prior to this octet are filled with data from the structure, from the previous sequence of eight CS-PDUs.



The Structured Data Transfer (SDT) convergence sublayer uses the pointer field to support transfer of $n \times 64$ Kbps signals, as illustrated in Figure 12-5. Starting at the top of the figure, the SDT CS accepts specific 64 Kbps time slots from the AAL1 user (e.g., a N-ISDN video teleconference), as indicated by the shaded portions of the framed DS1 in the figure. Note that the SDT CS does not convey TDM framing information indicated by the fields labeled "F" at the top of the figure. The figure illustrates how the pointer field occurs in only one out of seven SAR-PDUs, as described previously. The SAR sublayer adds the sequence number, inserts the data from the CS, and computes the CRC and parity over the SAR header, passing the 48-octet SAR-PDU to the ATM layer. The ATM layer adds the 5-byte ATM header and outputs the sequence of 53-byte cells, as shown at the bottom of the figure. The process at the receiver is analogous to that described previously, except the steps are reversed. The receiver then employs its local clock to take the contents from the received CS PDUs and clock out the resulting bit stream to the receiving device, placing it in the appropriate position in the DS1 TDM framing structure.



Unstructured Mode Convergence Sublayer

Figure 12-6 illustrates the unstructured method, which takes data from the AAL1 source, performs CS functions (e.g., asynchronous clocking information or FEC), and passes these data units to the SAR sublayer. The SAR sublayer takes 47-octet CS PDUs, prefixes these segments with a 1-octet SAR header, and passes them to the ATM layer for transmission as the cell payload. At the destination, the receiver's SAR sublayer takes the 48-octet ATM cell payload and examines the SAR-PDU header for errors. If the SAR-PDU header contains errors, the SAR layer then passes an indication to the CS, which may, in turn, inform the end application. The SAR sublayer passes correctly received SAR-PDU

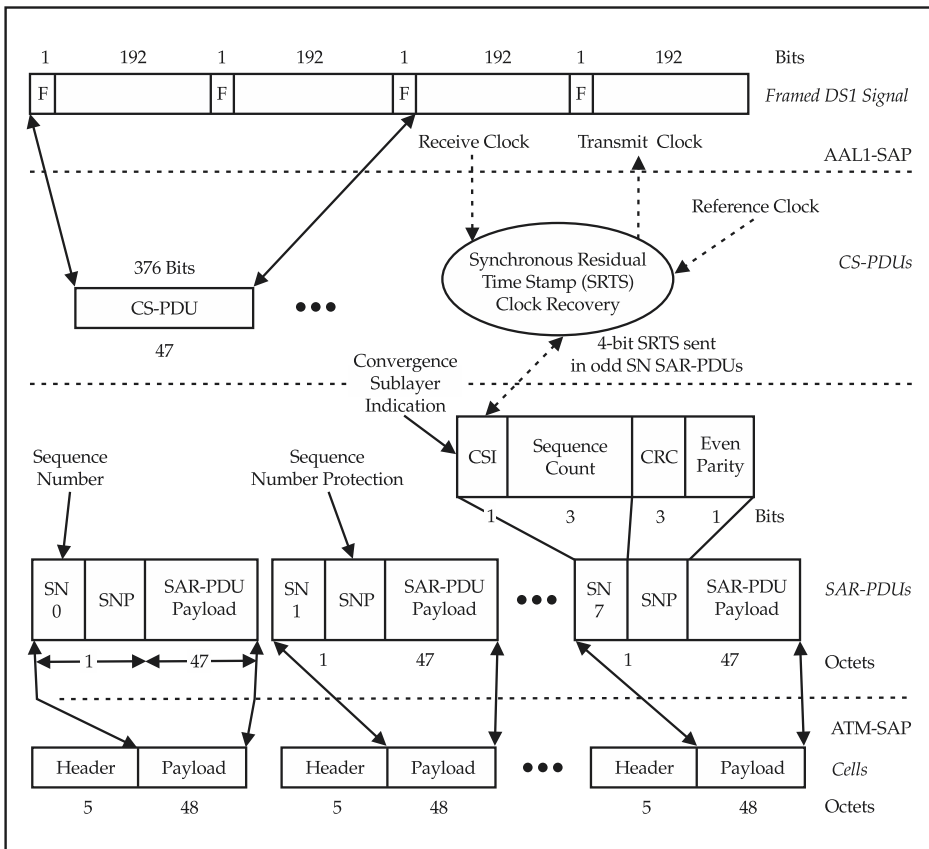


Figure 12-6. Unstructured mode AAL1 SAR and SRTS CS example

payloads to the CS, if present. The CS then reclocks the received bit stream to the destination AAL user, transparently passing any framing information in the original source signal as indicated at the top of the figure by the fields marked “F” in the AAL1 user bit stream. Unstructured mode may use either a synchronous or asynchronous clock recovery method, as described in the next section.

Figure 12-6 also illustrates the operation of unstructured mode using the Synchronous Residual Time Stamp (SRTS) method, which employs the CSI bit in SAR-PDUs with odd sequence numbers. The next section details the operation of the SRTS and adaptive clock recovery methods.

AAL1 Clock Recovery Methods

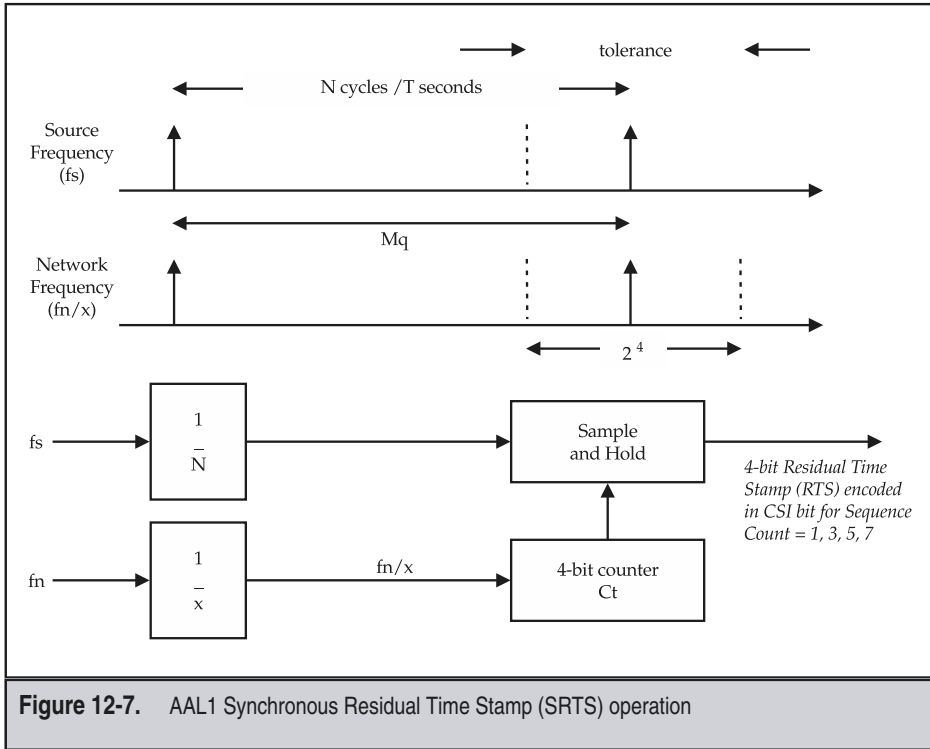
ITU-T Recommendation I.363.1 defines two methods for recovery of the source clock at the destination: synchronous and asynchronous. The synchronous method requires that the source and destination have access to the same accurate reference clock, which the standards call synchronous mode. Since N-ISDN is a synchronous TDM network, accurate clocks are usually available in most modern private and public digital telephone systems. Note that SDT CS requires synchronous clock sources at each device that convert between TDM and ATM. The destination SDT CS utilizes the structure pointer along with the structure parameters and its accurate clock to place the received 64 Kbps time slots in the framed TDM structure for delivery to the end user. The SDT mode is best suited to network designs that do not require an entire DS1/E1 between sites. For example, SDT works well in designs requiring the interconnection of PBXs in a corporate network. Accurate timing can be obtained from carrier interface circuits connected to the PBXs, from a SONET/SDH ATM UNI, or from the DS3 PLCP layer. The asynchronous method uses either an adaptive method or the Synchronous Residual Time Stamp (SRTS) method. Asynchronous clock recovery is essential to the support of legacy TDM networks—for example, T1 multiplexers—over an ATM network. The adaptive clock recovery method requires no reference clock at all, and hence it offers a plug-and-play approach to circuit emulation. Since standards don’t exist for adaptive clock recovery, many implementations offer a range of tuning parameters to control clock recovery performance. In adaptive clock recovery, the destination recovers an estimate of the source clock frequency from the intercell spacing of the received cell stream or the playback buffer fill level [Li 96a]. Adaptive clock recovery effectively controls jitter of the recovered clock signal; however, the long-term clock frequency may wander outside of normal TDM clock tolerances. A disadvantage of adaptive clock recovery is sensitivity to time-varying cell interarrival times caused by congestion or dynamically established and released CBR connections. In real-world networks, adaptive clock recovery may work acceptably for some applications that are not sensitive to jitter and wander.

But some applications require highly accurate timing. Using ATM CES for traditional private line replacement is one example where the use of an inappropriate clock synchronization method can result in excessive jitter and wander of the clock associated with the CES data outbound from an ATM network. Clock imperfections here can lead to observable defects on the end service: wow and flutter in high-fidelity acoustic services; visible

imperfections such as color impairments on video services; bit errors due to alignment jitter when interworking with the PDH or SONET/SDH network; or frame slips when interworking with existing PSTN, N-ISDN, and PBX networks. The limit for long-term phase wander for a 2 Mbps PDH CBR stream, as defined by ITU-T G.823 and G.824 [ITU G.823, ITU G.824], is 36.9 UI (unit intervals) at 1.2×10^{-5} Hz (a daily wander limit) down to 18 UI between 0.01 Hz and 1.667 Hz, and the jitter limit for a 2 Mbps PDH signal is 1.5 UI peak-to-peak between 20 Hz and 100 kHz. Note that the amount of jitter above 20 Hz is limited by the rate of change allowed by the adaptive clock's phase-locked loop. Given changes in the network load with a periodicity greater than the time constant of the adaptive clock's phase-locked loop—for example, in a network composed of many ATM switches passing through busy hours and idle periods—jitter and wander performance need to be carefully examined.

On the other hand, SRTS requires an accurate reference clock at both the source and the destination devices. Unlike the synchronous method, SRTS aims to accurately transfer the user's clock from source to destination, as required, for example, in many legacy T1 multiplexer networks. To do this, SRTS measures the difference between the source data rate and the local accurate clock reference, and transmits the difference to the destination in odd-numbered sequence number fields, as will be detailed. The destination uses this information to compute the clock rate at the destination prior to clocking out the received bit stream to the destination AAL1 user. SRTS works well in a single-carrier network or a private network where accurate clocks tied to the same reference are available. In single-carrier ATM networks, the physical access circuit often provides an accurate timing source. SRTS may result in degraded performance in connections that traverse multiple carrier networks, since an accurate time reference may not be available at both the source and the destination. Although the SRTS method delivers less jitter and receive frequency wander than the adaptive method, it requires an accurate reference clock at each end of the connection. Carefully investigate the timing source capabilities of your application to select the clock recovery method best suited to your network.

Figure 12-7 depicts more details on the operation of SRTS CS asynchronous clock recovery. As stated previously, SRTS assumes that both the origin and destination have access to a common clock of frequency f_n . The signal (e.g., DS1) has a service clock frequency f_s . The objective is to pass sufficient information via the AAL for the destination to accurately reproduce this clock frequency. At the bottom part of the figure, the network reference clock f_n is divided by x such that $1 \leq f_n/x/f_s \leq 2$. The source clock f_s divided by N samples the 4-bit counter C_t driven by the network clock f_n/x once every $N = 3008 = 47 \times 8 \times 8$ bits generated by the source. The SRTS CS uses the CSI bit in the SAR-PDU to send this sampled 4-bit Residual Time Stamp (RTS) in odd-sequence-numbered SAR-PDUs. ITU-T Recommendation I.363.1, ANSI T1.630, and Bellcore TA-NWT 1113 show how the SRTS method accepts a frequency tolerance for a difference between the source frequency and the reference clock frequency of 200 parts per million (ppm). The SRTS method is therefore capable of meeting the jitter and wander specifications of the ITU G.702, 1.544 Mbps- and 2.048 Mbps-based hierarchies specified in ITU G.823 and ITU G.824, respectively. Note that higher rates than this can be supported with the SRTS method but are not explicitly specified in standards.



ATM ADAPTATION LAYER 2 (AAL2)

AAL2 supports ATM transport of connection-oriented variable bit rate packetized voice and video. Setting a new record for the standards process, the ITU-T approved the basic definition of the AAL2 protocol in Recommendation I.363.2 in September 1997, after only 9 months. Close cooperation between the ATM Forum's Voice and Telephony over ATM (VTOA) working group and the ITU-T facilitated this remarkable accomplishment. This section provides an overview of the key methods, which the AAL2 protocol employs to minimize delay and improve efficiency for real-time voice and video. Services provided by AAL2 include providing a means for identifying and multiplexing multiple users over a common ATM layer connection, transferring service data units at a variable bit rate, and indication of lost or erroneous information. AAL2 has a number of advantages when transporting voice over ATM, including more efficient bandwidth usage due to silence detection and suppression, as well as idle voice channel deletion.

AAL2 Protocol Structure and PDU Formats

Unlike the other AALs covered in this section, AAL2 has no Segmentation and Reassembly Sublayer, but instead employs the structure illustrated in Figure 12-8.

Like all other AALs, AAL2 has an interface to the ATM layer at the ATM-SAP and an interface to the higher-layer user at the AAL2-SAP. The Common Part Sublayer (CPS) has two components as indicated in the figure: a CPS Packet Header and a CPS Protocol Data Unit (CPS-PDU). As described in Chapter 16, two Service-Specific Convergence Sublayers (SSCS) have been built atop the CPS foundation of AAL2. AAL2-oriented SSCS protocols defined are ITU-T Recommendation I.366.2, which defines trunking of narrowband ISDN, voice, and facsimile traffic over ATM. ITU-T Recommendation I.366.1 defines support for a frame mode service over AAL2. The more challenging problem for voice band

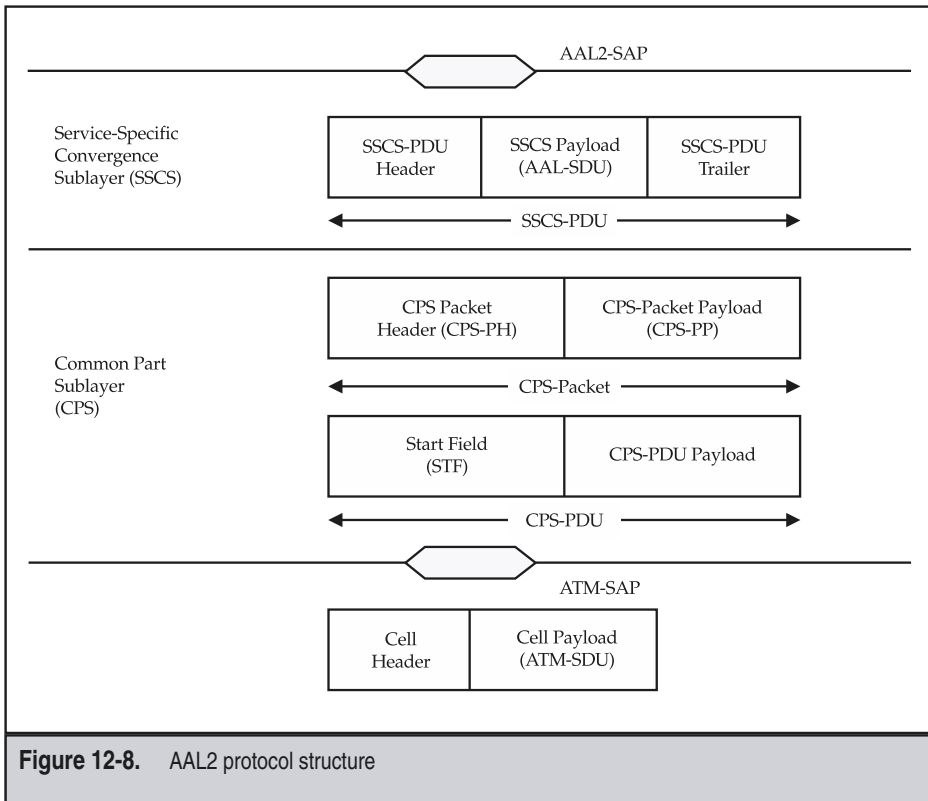


Figure 12-8. AAL2 protocol structure

support over AAL2 is to avoid delay variation in playback caused by statistical fluctuations in voice activity. Such variations in phase could cause disruption to modem or fax transmissions carried over the same AAL2 VCC.

The CPS provides the means to identify AAL2 users multiplexed over a single ATM VCC, manage assembly and disassembly of the variable-length payloads for each user, and interface to the SSCS. The CPS provides an end-to-end service via concatenating a sequence of bidirectional AAL2 channels operating over an ATM Virtual Channel Connection (VCC). Each AAL2 user generates CPS packets with a 3-octet packet header and a variable-length payload, as illustrated in Figure 12-9.

AAL2 uses the 8-bit Channel ID (CID) in the CPS Packet Header (CPS-PH) to multiplex multiple AAL2 users onto a single VCC. The CID field supports up to 248 individual users per VCC, with eight CID values reserved for management procedures and future functions. Next, the 6-bit Length Indicator (LI) field specifies the number of octets (minus one) in the variable-length user payload. The maximum length of the user payload is selected as either 45 or 64 octets. Note that selecting the 45-octet value means that exactly one CPS packet fits inside the 48-octet ATM cell payload. The 5-bit User-to-User Indication (UUI) field provides a means for identifying the particular SSCS layer along with support for OAM functions. A 5-bit Header Error Control (HEC) field provides error detection and correction for the CPS-PH. A rationale similar to that described in the preceding chapter regarding the use of HEC in the ATM cell header drives the need to protect the AAL2 CPS packet header, since undetected errors may affect more than one connection. The CPS packet payload may be up to 64 octets in length.

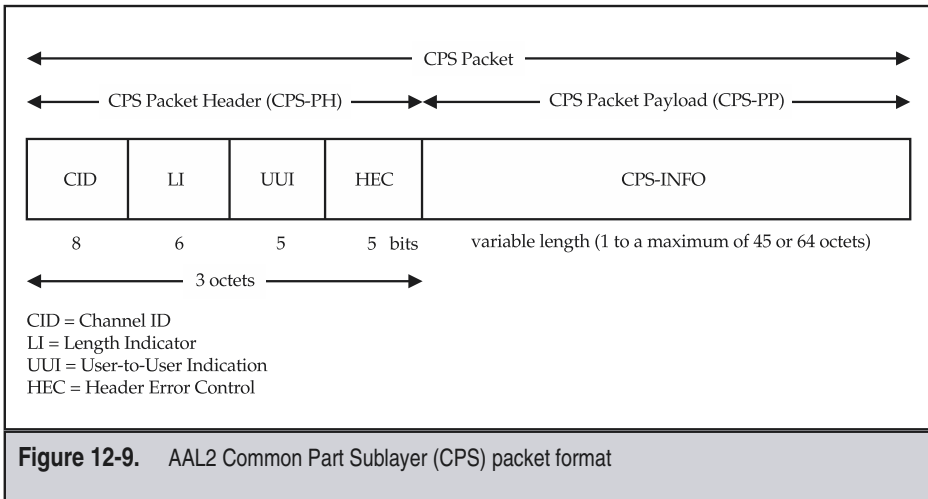


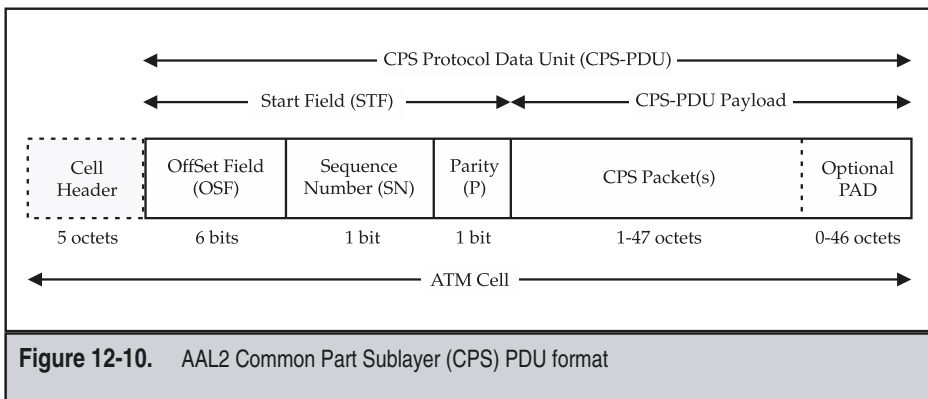
Figure 12-9. AAL2 Common Part Sublayer (CPS) packet format

The CPS sublayer collects CPS packets from the AAL2 users multiplexed onto the same VCC over a specified interval of time, forming CPS-PDUs consisting of 48 octets' worth of CPS packets. Figure 12-10 illustrates the CPS-PDU format employed by AAL2. The CPS-PDU employs a 1-octet Start Field (STF) followed by a 47-octet payload. A 6-bit Offset Field (OSF) in the Start Field (STF) identifies the starting point of the next CPS packet header within the cell. Note that if more than one CPS packet are present in a cell, then AAL2 uses the Length Indicator (LI) in the CPS packet header to compute the boundary of the next packet. The Offset Field (OSF) allows CPS packets to span cells without any wasted payload, as the example in the next section demonstrates. Since the Start Field (STF) is critical to the reliable operation of AAL2, one-bit Sequence Number (SN) and Parity (P) fields provide for error detection and recovery. In order to maintain real-time delivery, the AAL2 protocol times out if no data has been received and inserts a variable-length PAD field to fill out the 48-octet ATM cell payload.

Example of AAL2 Operation

Figure 12-11 illustrates an example where AAL2 multiplexes four real-time, variable bit rate sources (labeled A, B, C, and D) into a single ATM virtual channel connection. Starting at the top of the figure, each source generates 16-octet samples, which pass across the AAL2 Service Access Point to the Common Part Sublayer (CPS), which forms CPS packets by prefixing each sample with a 3-octet CPS Packet Header (CPS-PH) containing a Channel ID (CID) that identifies the source letter in the illustration. The CPS sublayer collects CPS packets over a specified interval of time and forms CPS-PDUs composed of 48 octets' worth of CPS packets using a 1-octet Start Field (STF) as shown in Figure 12-11.

In this example, each CPS packet consumes 19 octets in the CPS-PDU. If the sources have been inactive, then the STF offset points to the next octet in the CPS-PDU, as shown



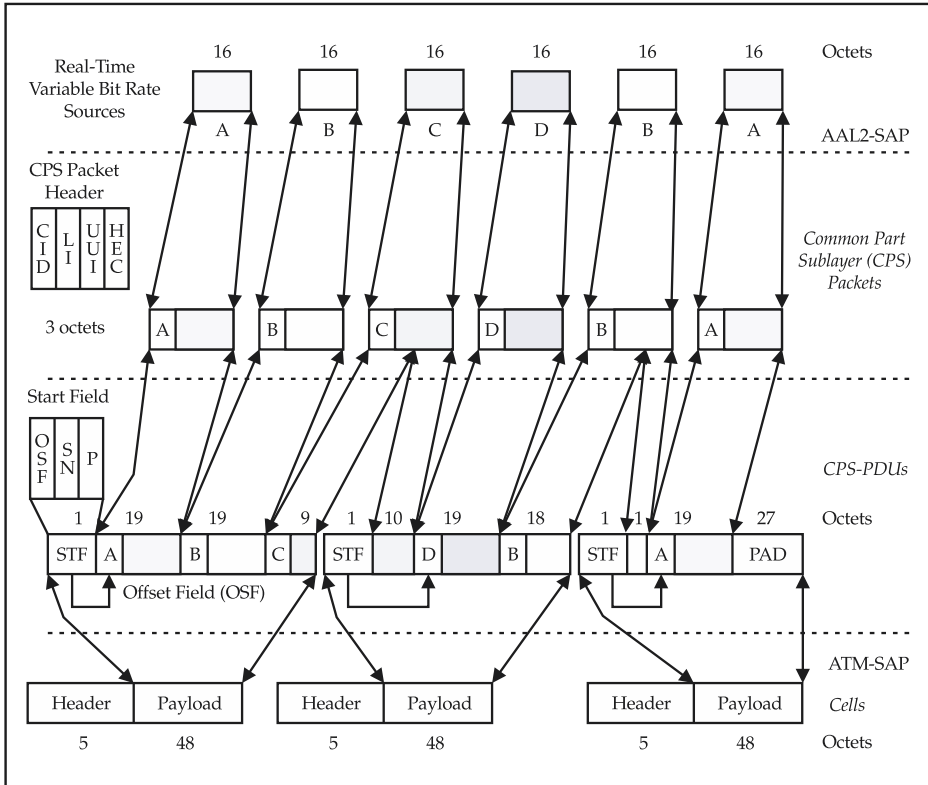


Figure 12-11. Example of AAL2 Operation

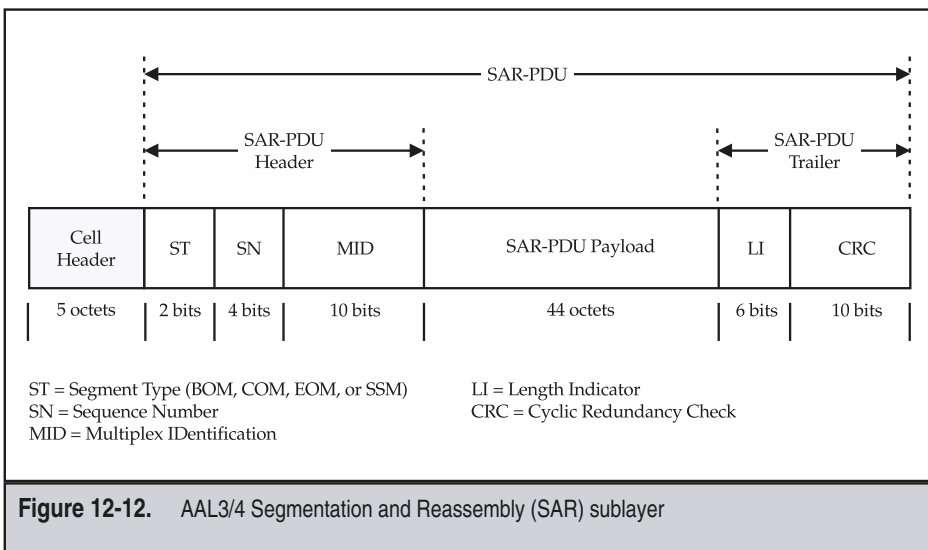
by the arrow in the leftmost CPS-PDU in Figure 12-11. In general, the STF offset points to some other position when CPS packets span CPS-PDU boundaries, as shown in the second and third CPS-PDUs. In order to maintain real-time delivery, the AAL2 protocol times out if no data has been received and inserts a PAD field to fill out the 48-octet ATM cell payload, as shown in the third CPS-PDU in the figure. Finally, the protocol maps AAL2 CPS-PDUs to ATM cell payloads across the ATM-SAP for a Virtual Channel Connection. Thus, AAL2 reduces packetization delay by multiplexing multiple sources together and controls delay variation by inserting the PAD field if the period of source inactivity exceeds a specific timer threshold. As discussed in Chapter 16, minimal packetization delay is critical for voice over ATM due to echo control problems. Furthermore, control of delay variation is critical for both voice and video over ATM.

ATM ADAPTATION LAYER 3/4 (AAL3/4)

ITU-T Recommendation I.363.3 combines AAL3 and AAL4 into a single common part, AAL3/4, in support of VBR traffic, both connection oriented and connectionless. The AAL3/4 protocol conforms to the generic AAL protocol model because it has Segmentation and Reassembly (SAR) and a Convergence Sublayer (CS). Support for connectionless service is provided at the Service-Specific Convergence Sublayer (SSCS) level. The text provides an example of AAL3/4 operation, illustrating the multiplexing function provided by this AAL over a single ATM VCC.

AAL3/4 SAR Sublayer

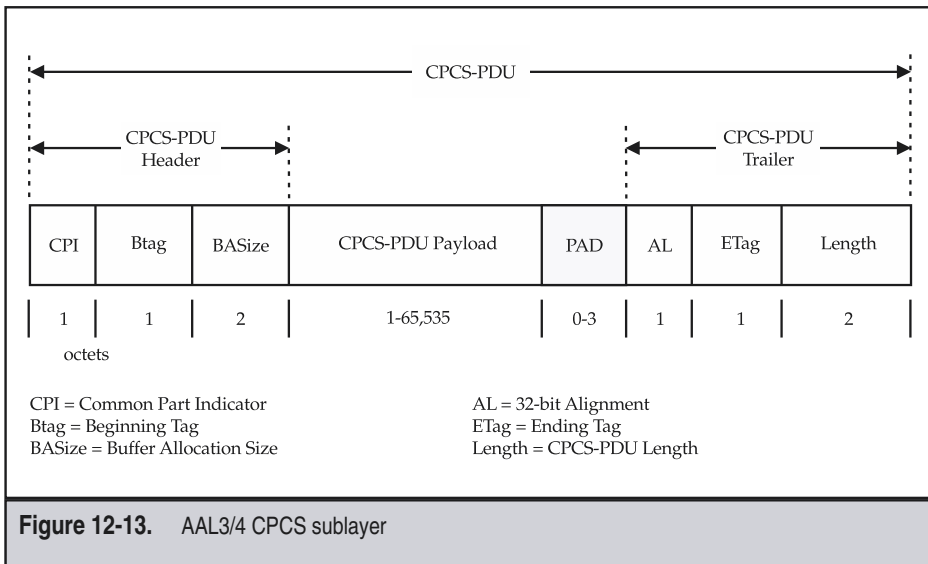
Figure 12-12 depicts the SAR for AAL3/4. The SAR-PDU encoding and protocol function and format are nearly identical to the L2_PDU from IEEE 802.6, as summarized in Chapter 8. The SAR-PDU has a 2-octet header and trailer. The header contains three fields, as shown in the figure. The 2-bit Segment Type (ST) field indicates whether the SAR-PDU is a Beginning of Message (BOM), a Continuation of Message (COM), an End of Message (EOM), or a Single Segment Message (SSM). The sender increments the 2-bit Sequence Number (SN), which the receiver uses to detect lost SAR-PDUs. The numbering and checking begins when the receiver detects a BOM segment. The ten-bit Multiplex Identification (MID) field allows multiplexing of up to 1,024 different CPCS-PDUs over a single



ATM VCC. This is a key function that differentiates AAL3/4 from AAL5, which is important when a carrier charges per VCC, motivating users to do their own multiplexing to minimize cost. However, as described in Chapter 18, IETF RFC 2684 defines an analogous means of multiplexing multiple protocols over a single ATM VCC using AAL5. The AAL3/4 SAR function is essentially the same one used in the 802.6 L2 protocol, where the cell header contains effectively no addressing. The transmitter assigns the MID prior to sending a BOM or SSM segment. The MID value in a BOM segment ties together the subsequent COM and EOM portions of a multisegment message. The SAR-PDU trailer has two fields. The 6-bit Length Indicator (LI) specifies how many of the octets in the SAR-PDU contain CPCS-PDU data. LI has a value of 44 in BOM and COM segments and may take on a value less than this in EOM and SSM segments. The 10-bit CRC checks the integrity of the segment.

AAL3/4 CPCS Sublayer

Figure 12-13 depicts the CPCS-PDU for AAL3/4. The header has three components, as indicated in the figure. The 1-octet Common Part Indicator (CPI) indicates the number of counting units (bits or octets) for the Buffer Allocation Size (BASize) field. The sender inserts the same value for the 2-octet Beginning Tag (BTag) and the Ending Tag (ETag) so that the receiver can match them as an additional error check. The 2-octet BASize indicates how much buffer space the receiver must reserve to reassemble the CPCS-PDU. A variable-length PAD field ranging between 0 and 3 octets makes the CPCS-PDU an integral multiple of 32 bits to make end system processing more efficient.



The trailer also has three fields, as shown in Figure 12-13. The 1-octet Alignment (AL) field simply makes the trailer a full 32 bits to simplify the receiver design. The 1-octet ETag must have the same value as the BTag at the receiver for the CPCS-PDU to be valid. The Length field encodes the length of the CPCS-PDU field so that the PAD portion may be taken out before delivering the payload to the CPCS user.

Example of AAL3/4 Operation

Figure 12-14 depicts an example of the operation of the AAL3/4 SAR and CS sublayers for AAL3/4. Starting from the bottom of the figure, the 48-octet payload of a sequence of cells on the same Virtual Channel Connection (VCC), i.e., cells having the same Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) values, interface with the AAL3/4 SAR sublayer across the ATM-SAP. The 2-bit Segment Type (ST) field indicates that the SAR-PDU is a Beginning of Message (BOM), a Continuation of Message (COM),

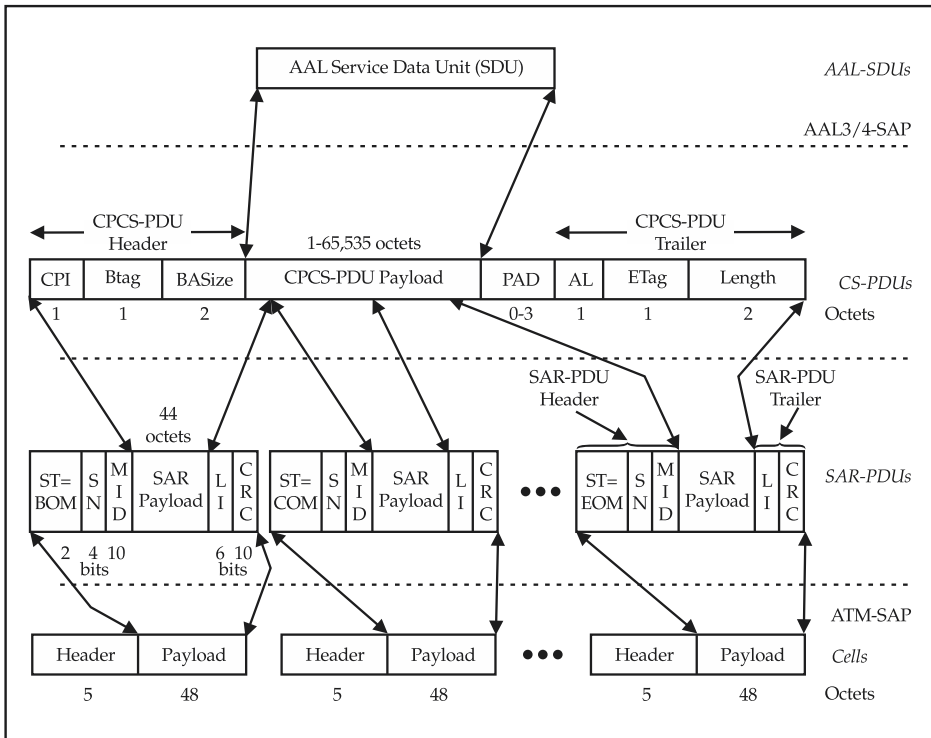


Figure 12-14. Example of AAL3/4 SAR and CS sublayers

and—finally, after a number of cells containing COM indications—a cell containing an End of Message (EOM) segment type, as shown in the example.

If the SAR sublayer receives all SAR-PDUs in a message in sequence with correct CRC values and matching tags, then it passes the reassembled packet up to the CPCS layer. In message mode, AAL3/4 accepts one AAL-IDU at a time and optionally sends multiple AAL-IDUs in a single SSCS-PDU. In streaming mode, the higher-layer protocol may send multiple AAL-IDUs separated in time; the SSCS may deliver these in multiple AAL-IDUs, or reassemble the pieces and deliver only one AAL-IDU [ITU I.363.1]. The principal advantages of AAL3/4 are multiplexing of multiple logical connections over a single ATM VCC, additional error-checking fields, and the indication of message length in the first cell for use in efficient buffer allocation in intermediate or destination switches.

AAL3/4 Multiplexing Example

This section illustrates the operation of AAL3/4 multiplexing through the example shown in Figure 12-15. This depicts a data communications terminal that has two inputs with two 98-byte (or octet) packets arriving simultaneously destined for a single ATM VCC. Two parallel instances of the CPCS sublayer encapsulate the packets with a header and trailer. These then pass to two parallel Segmentation and Reassembly (SAR) processes that segment the CPCS-PDU on two different MIDs, resulting in a BOM, COM, and EOM segment for each input packet.

Because all of this occurred in parallel, the ATM cells resulting from this process are interleaved on output, as shown on the right-hand side of the figure. This interleaving over a single VCC is the major additional function of AAL3/4 over AAL5, as seen by comparison with the AAL5 example in the next section. Also, the multiple levels of error checking in AAL3/4 make the probability of an undetected error very small.

ATM ADAPTATION LAYER 5 (AAL5)

If AAL1, AAL2, and AAL3/4 appear complicated, you can now appreciate the motivation for developing a Simple Efficient Adaptation Layer (SEAL). Standards assigned the next available number to this lightweight protocol, which made it AAL5. As of publication time, it was still the last word in AAL supporting packet data, a number of proposals for AAL6 never having achieved sufficient backing for standardization. The Common Part (CP) AAL5 supports Variable Bit Rate (VBR) traffic, both connection-oriented and connectionless. Support for connectionless or connection-oriented service is provided at the Service-Specific Convergence Sublayer (SSCS) level, as defined for access to SMDS in Chapter 17. Most other data protocols operate over AAL5, and not AAL3/4—SMDS, the ATM Data Exchange Interface (DXI), and ATM Frame-Based UNI (FUNI) being the only exceptions that optionally support AAL3/4. Furthermore, the ATM control plane operates over AAL5, as described in Chapter 13. Part 4 provides examples of video, Frame Relay, LAN protocols, and IP, all operating over AAL5. However, despite its simplicity,

AAL5 Segmentation and Reassembly (SAR) Sublayer

Figure 12-16 depicts the SAR-PDU for AAL5. The SAR-PDU is simply 48 octets from the CPCS-PDU. The only overhead the SAR sublayer makes use of is the Payload Type code points for *AAL_Indicate*. *AAL_indicate* is zero for all but the last cell in a PDU. A nonzero value of *AAL_Indicate* identifies the last cell of the sequence of cells indicating to the receiver that reassembly can begin. This makes the reassembly design simpler and makes more efficient use of ATM bandwidth.

AAL5 Common Part Convergence (CPCS) Sublayer

Figure 12-17 depicts the CPCS-PDU for AAL5. The payload may be any integer number of octets in the range of 1 to $2^{16} - 1$ (65,535). The PAD field has a variable length chosen such that the entire CPCS-PDU is an exact multiple of 48 so that it can be directly segmented into cell payloads. The User-to-User (UU) information is conveyed transparently between AAL users by AAL5. The only current function of the Common Part Indicator (CPI) is to align the trailer to a 64-bit boundary, with other functions for further study. The Length field identifies the length of the CPCS-PDU payload so that the receiver can remove the PAD field. Since 16 bits are allocated to the Length field, the maximum payload length is $2^{16} - 1 = 65,535$ octets. The CRC-32 detects errors in the CPCS-PDU. The CRC-32 is the same one used in IEEE 802.3, IEEE 802.5, FDDI, and Fiber Channel. Chapter 25 compares the undetected error performance of AAL5 and HDLC.

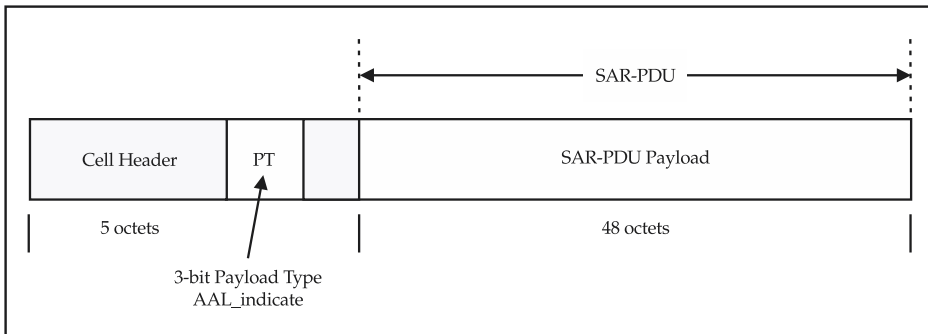
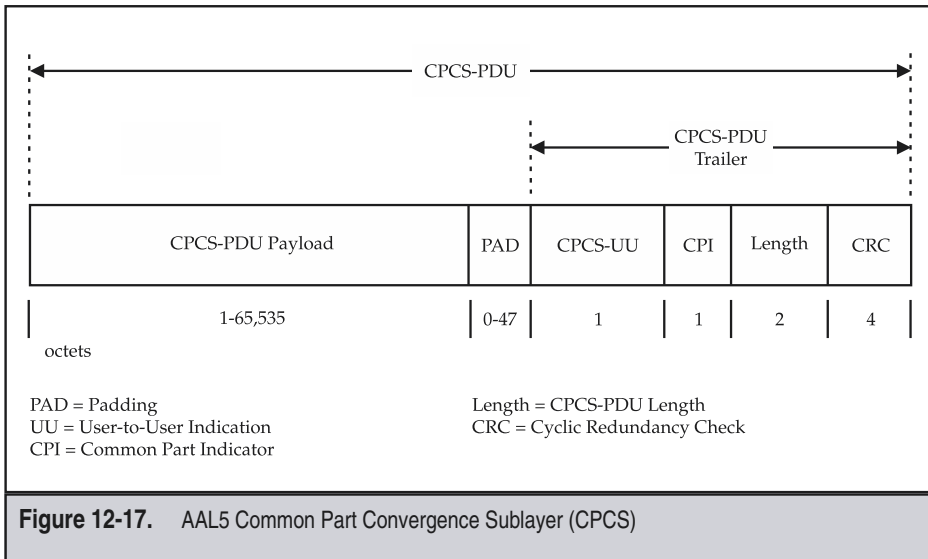


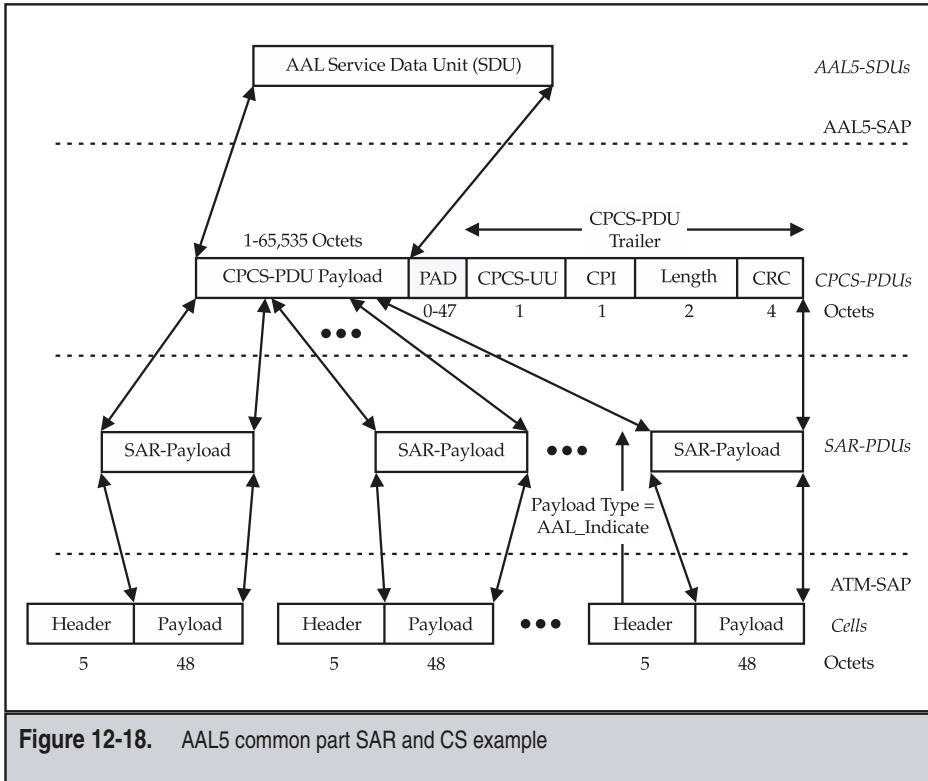
Figure 12-16. AAL5 Segmentation and Reassembly (SAR) sublayer

Example of AAL5 Operation

Figure 12-18 depicts an example of the operation of the AAL5 SAR and CPCS sublayers. The relative simplicity of AAL5 with respect to AAL3/4 is readily apparent by comparing this figure with Figure 12-14. Starting from the ATM cell stream on a single VCC at the bottom of the figure, note that the only overhead the SAR sublayer uses is the Payload Type field in the last cell of a sequence of cells corresponding to a single PDU (i.e., packet). A nonzero value of the AAL_Indicate field identifies the last cell in the sequence of cells, indicating that the receiver can begin reassembly. The SAR sublayer reassembles the CPCS-PDU and passes it to the CPCS sublayer, which first uses the CRC-32 field to check for any errors in the received PDU. Normally, CPCS discards corrupted PDUs, but it may optionally deliver them to an SSCS. The CPCS removes the PAD and other trailer fields before passing the AAL-SDU across the AAL5-SAP. The transmit operation is the reverse of the preceding description.

The payload may be any integer number of octets in the range of 1 to $2^{16} - 1$ (65,535). The PAD field has a variable length chosen such that the entire CPCS-PDU is an exact multiple of 48 so that it can be directly segmented into cell payloads. The User-to-User (UU) information is conveyed transparently between AAL users by AAL5. The only current function of the Common Part Indicator (CPI) is to align the trailer to a 64-bit boundary, with other functions for further study. The Length field identifies the length of





the CPCS-PDU payload so that the receiver can remove the PAD field. Since 16 bits are allocated to the Length field, the maximum payload length is $2^{16} - 1 = 65,535$ octets. The CRC-32 detects errors in the CPCS-PDU. The CRC-32 is the same one used in IEEE 802.3, IEEE 802.5, FDDI, and Fiber Channel.

AAL5 Multiplexing Example

Figure 12-19 depicts the same example previously used for AAL3/4 to illustrate the major difference in multiplexing operation. The figure depicts a data communications terminal that has two 98-byte packets arriving almost simultaneously, destined for a single ATM VCC, this time using the AAL5 protocol.

On the left-hand side of the figure, the two 98-byte packets arrive in close succession. Two parallel instances of the CPCS sublayer add PAD and trailer fields to each packet. Note that in AAL5, the entire packet need not be received before it can begin the SAR function, as would be required in AAL3/4 to insert the correct Buffer Allocation Size

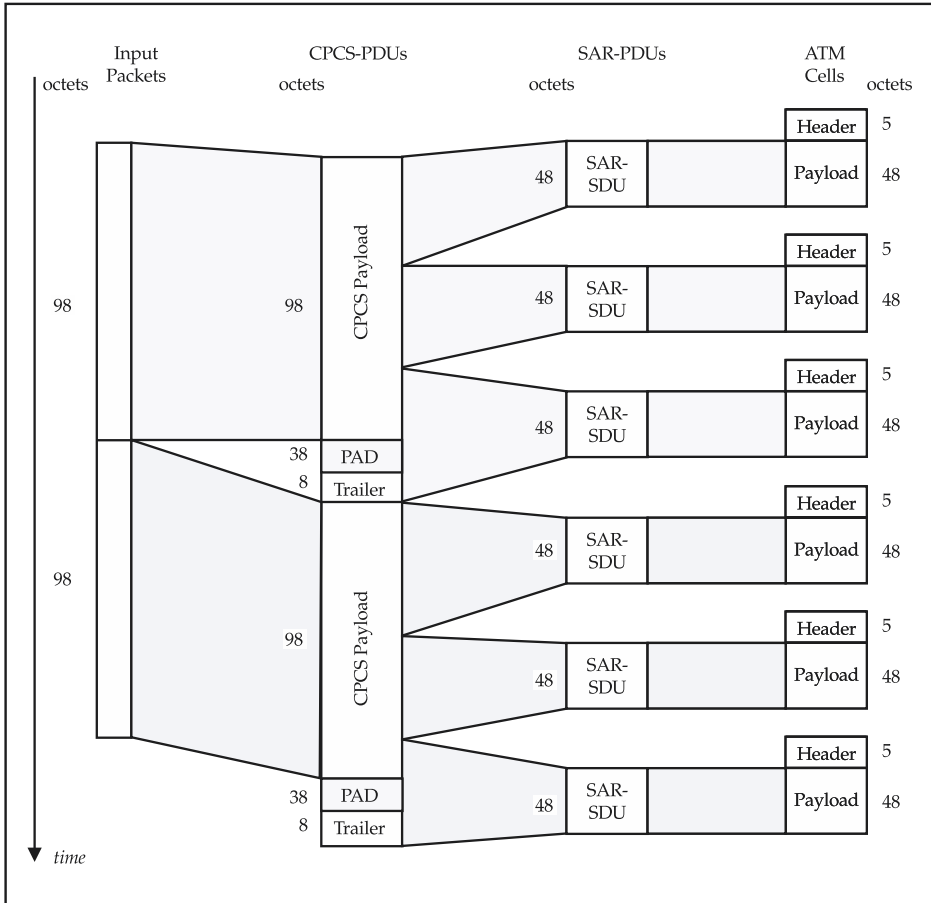


Figure 12-19. Multiplexing example using AAL5

(BASize) field. Two parallel Segmentation and Reassembly (SAR) processes segment the CPCS-PDU into ATM cells, as shown in the middle of the figure. In the example, these cell streams share the same VCC; hence, the device can send only one at a time. The AAL5 implementation is simpler than AAL3/4, but is unable to keep the link as fully occupied as the additional multiplexing of AAL3/4 could if the packets arrive much faster than the rate at which SAR and ATM cell transmission occurs. Furthermore, the serialization process causes greater delay variation in AAL5 than for AAL3/4 when sharing the same VCC. However, putting delay variation-sensitive packet streams on separate VCCs solves this problem when using AAL5.

MULTI-SERVICE TUNNELING OVER MPLS (AND OTHER PROTOCOLS)

Unlike B-ISDN, the IETF had no predetermined architectural vision about how to support multiple services, and this was not the intent of the “multiprotocol” part of the MPLS acronym. From the IP architectural perspective, everything runs over IP, and IP is capable of running over (virtually) anything. But a concept similar to that of ATM adaptation layers has arisen in the IETF’s Pseudo-Wire Emulation Edge-to-Edge (abbreviated PWE3) working group. Although no standard existed at the time of writing, the requirements and architectural framework as well as a popular encapsulation were in draft form, and that is what we summarize in this section.

In a related vein, the ATM Forum has published a means for carrying ATM cells over MPLS. Once this is done, all of the previously described AAL operations then become available to higher-layer applications because the protocol substrate is then ATM everywhere (including ATM over MPLS in at least some places in a network). Since this standard provides the means for supporting AALs and the applications that use them, we summarize this protocol here.

Some of the protocol stacks resulting from the preceding approaches violate the strict layering of protocols from the OSI model, as described in Chapter 5. For example, ATM over MPLS, or MPLS over ATM, involves operating one layer 2 protocol on top of another. Even greater divergence with the OSI paradigm exists in proposals for carriage of a layer 2 protocol over a layer 3 protocol, for example, Ethernet over IP. Although many of these approaches turn the OSI model upside down or create layering chaos, the reader should keep in mind that arbitrary stacking of protocols (or said another way, tunneling one protocol over another) does not come without some trade-offs. The principal advantage is that by stacking protocols on top of each other, arbitrary topologies at layer 2 or 3 can be created from a variety of underlying ATM, MPLS, FR, Ethernet, or IP networks. An important class of such topologies is that of virtual private networks, as discussed in Part. There are several disadvantages. The first is the complexity of configuring and maintaining these stacked protocols. A second is that of performance, which means that if any of the underlying networks do not support the performance needed by the top-level application, then the utility of the approach diminishes. Finally, some protocol stacks are more efficient than others in support of a particular application. Part 8 provides a discussion of these issues as they relate to the future direction of ATM and MPLS.

GENERAL CONCEPT OF PROTOCOL TUNNELING

Figure 12-20 illustrates some of the important generic concepts involved in tunneling one protocol over another. At the top of the figure, each point-to-point tunnel has a pair of endpoints. A tunnel may be unidirectional or bidirectional. The tunnel endpoints receive

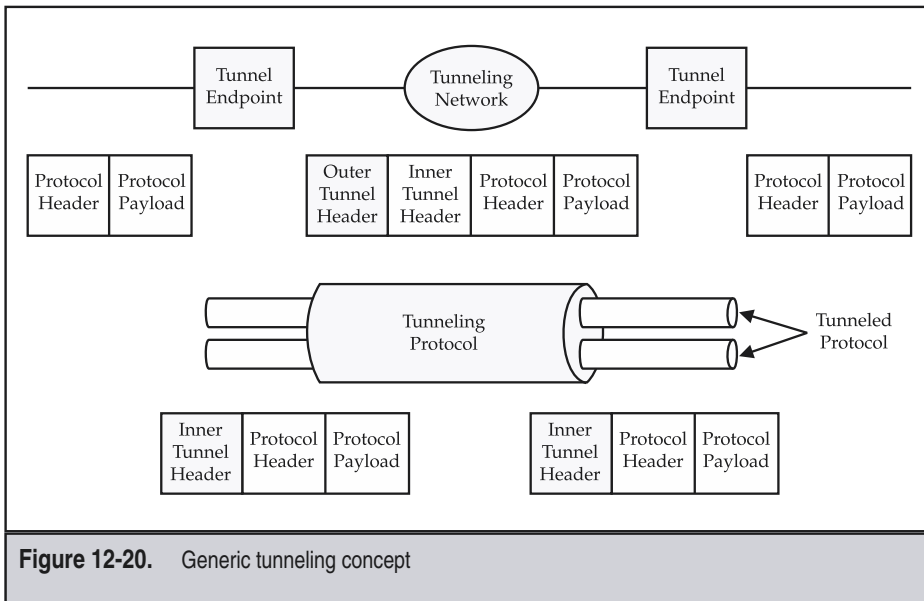


Figure 12-20. Generic tunneling concept

information from a particular protocol (e.g., IP), which has a header and a payload. A tunnel endpoint may in general support multiple protocols (or multiple instances of a single protocol) with another tunnel endpoint, in which case, an inner tunnel header field is added at the source and removed at the destination tunnel endpoints. A sending tunnel endpoint must add an outer tunnel header that is meaningful to the tunneling network for the purposes of forwarding the packet to the destination tunnel endpoint. The receiving tunnel endpoint removes this outer (or topmost) tunnel header, and processes any inner (or lower) tunnel header(s) that are present to forward the original protocol header and payload on toward its destination. As a means to improve efficiency, some protocols compress the information in the protocol header upon entry to the tunnel and then expand it upon exit from the tunnel. We now look at a few specific instances of ATM and MPLS-based tunneling protocols designed to support specific forms of this generic tunneling networking model.

A number of tunneling protocols similar in function to the generic model described previously have been defined by the IETF. These include IP in IP [RFC 2003, RFC 2473], Generic Routing Encapsulation (GRE) [RFC 2784], IPsec [RFC 2401], and version 3 of Layer 2 Tunneling Protocol (L2TP) [RFC 2661], as well as the ATM- and MPLS-based protocols introduced in this section. The L2TPv3 protocol is being considered as a control signaling protocol and/or an encapsulation method.

ATM FORUM'S ATM OVER MPLS NETWORK INTERWORKING

ATM Forum specification AF-AIC-0178.000 specifies an encapsulation and related protocols that allow carriage of ATM cells or AAL5 PDUs over an MPLS LSP. Means to support both ATM virtual path and virtual channel connections are defined. In ATM Forum and ITU terminology, the trunking of one protocol over another is often called *network interworking*, a definition that the reference configuration for support of ATM over MPLS shown in Figure 12-21 illustrates. At the left and right, an ATM- (or AAL-) based service connects via an ATM network. In the center of the figure, at the boundary between the ATM and MPLS networks, stands an interworking network element (INE). The INE encapsulates a stream of ATM cells (or AAL5 PDUs) received from the ATM network into one or more interworking LSPs using a MPLS shim header. The INE then encapsulates the one or more interworking LSPs into a topmost (level-1) “transport” LSP so that the MPLS network can deliver the payload to the destination INE. The destination INE removes the topmost label and uses the stacked interworking label to determine the interworking processing and eventual ATM destination. Let’s look inside the ATM over MPLS encapsulation to better understand this function.

Figure 12-22 summarizes the resulting encapsulations defined in the ATM Forum’s ATM over MPLS encapsulation specification. The MPLS transport and interworking labels are the 32-bit (i.e., 4-octet) MPLS shim header defined in Chapter 11. These labels form the transport and interworking LSPs shown in Figure 12-21. The next bit after the MPLS interworking label defines the encapsulation mode: a 0 indicates that the payload is an ATM cell, and a 1 indicates that the payload is an AAL5 PDU. We now describe

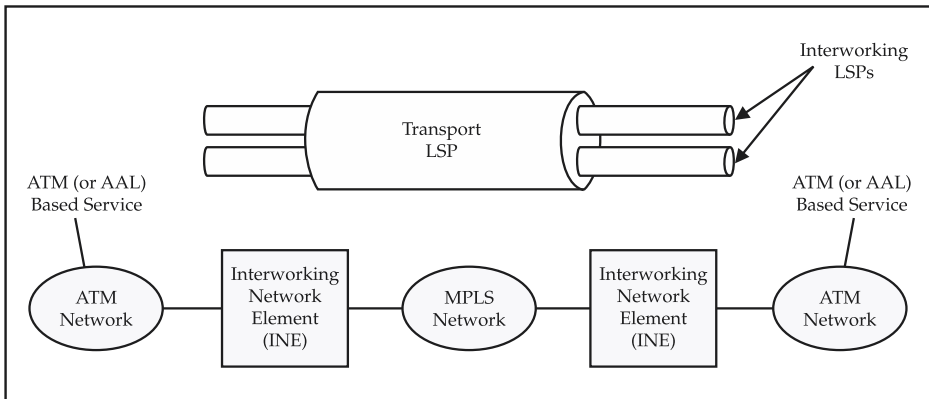
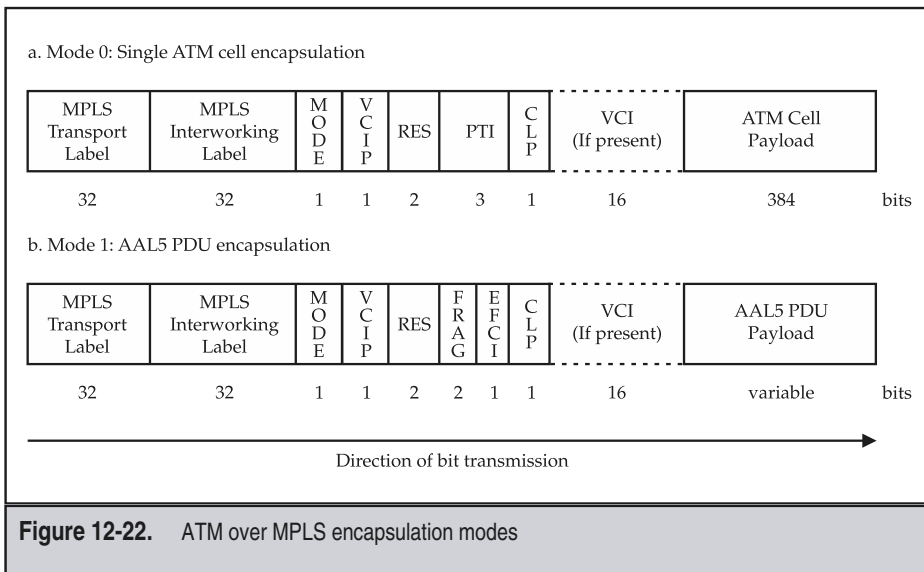


Figure 12-21. ATM over MPLS network interworking reference configuration



mode 0, ATM cell encapsulation, with reference to Figure 12-22a. The next bit is the VCI Present (VCIP) indicator, which, if set to 1, means that the 16-bit VCI field shown in dashed lines is present; otherwise, no VCI is present. VCIs may be absent, for example, if only a single VCC exists on an interworking LSP. Normally, to achieve greater efficiency, multiple cells would be multiplexed within a single LSP. In this case, the 50-octet data structure beginning with the mode bit, including the 2-octet VCI field and ending with the 48-octet (384-bit) payload field, may be repeated multiple times, constrained only by the MPLS LSP MTU length. ATM cell payload may be repeated. The next two bits are reserved (RES), followed by the ATM Payload Type Indicator (PTI) and Cell Loss Priority (CLP) bits from the ATM cell header, as described in Chapter 11. Since the entire PTI field is present, this encapsulation mode allows support of ATM OAM cells. Finally, the 48-octet (i.e., 384-bit) ATM cell payload follows. This PDU is, of course, encapsulated in some other L2 protocol, such as Packet Over SONET (POS), as described in Chapter 11.

Figure 12-22b illustrates the frame mode, AAL5 PDU encapsulation over MPLS. An INE receives cells and reassembles the entire AAL5 PDU before encapsulation. The VCIP and RES fields are identical in function to that described for cell mode previously. The 2-bit fragmentation (FRAG) field indicates a beginning, continuation, end, or single segment message in a manner similar to AAL3/4 in the event that the INE must fragment the AAL5 PDU in order to meet MTU constraints of the MPLS network. Only the explicit forward congestion indication (EFCI) bit of the ATM cell header PTI field is needed, since there is no support for OAM and the AAL indication is implicitly accounted for in frame mode. The CLP bit has the same function as in the ATM cell header. The AAL5 PDU

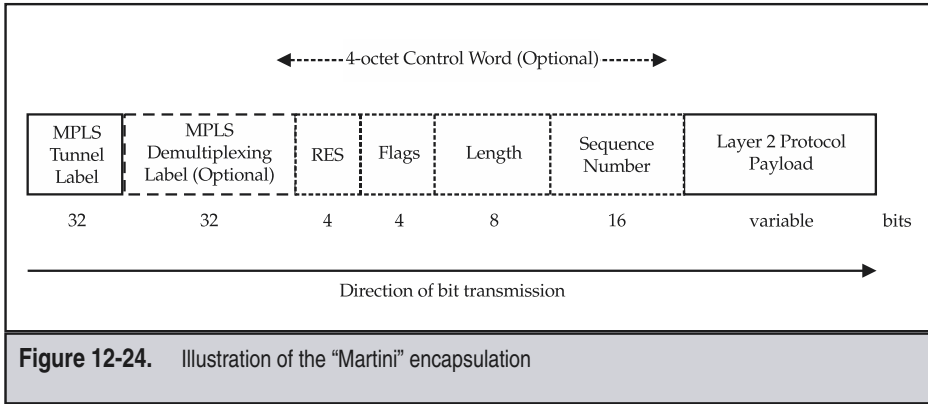
protocol layers. The view is that PSN tunnel control and management procedures already defined for IP or MPLS tunnels would be used without change. The pseudo-wire (PW) payload encapsulation layer makes the choice of PSN tunneling protocol essentially transparent to higher layers, except, of course, for performance impairments such as loss or errors that cannot be masked. The PW payload encapsulation layer may perform sequencing, duplicate detection, and loss detection. It may also perform absorption of delay variation and/or timing transfer in support of an emulated service that requires it—for example, TDM. A PE also provides additional support required to emulate a specific service—for example, relaying of TDM alarms or FR status signaling, or ATM OAM functions. The scope of PWE3 includes the protocols necessary to establish and maintain pseudo-wire and service emulation features.

Finally, as shown for PE2, a provider edge device may also contain native service processing (NSP)—for example, Ethernet bridging, FR/ATM service interworking (see Chapter 17), or a TDM cross-connect function. The interface between an NSP and the PWE3 protocol stack is called a “virtual physical termination” in the architecture to emphasize that the NSP function could be in the PE or in the CE, which, in either case, are connected via a physical interface. The motivation for placement of the NSP within a PE is that it may well cost less and be easier to manage because it simplifies the pseudo-wire to that of homogeneous operation.

“MARTINI” MULTI-SERVICE ENCAPSULATION

As a more concrete example of the PWE3 architectural concept, this section briefly describes an encapsulation approach often referred to by the last name of the principal author of an Internet draft, namely, Luca Martini [Martini 02]. Figure 12-24 illustrates the general structure of the “Martini” encapsulation that was designed to support a number of layer 2 protocol virtual connections (VCs), for example, ATM, FR, HDLC, and PPP over a generic MPLS tunnel, as indicated by the leftmost label in the figure. An optional demultiplexing label could support multiple L2 VCs over the same outermost tunnel. At the time of writing, a related Internet draft also described signaling protocols to establish the mapping of VC FECs. One interesting architectural objective of this encapsulation was to use a generic control word, with L2 protocol-specific flags and some reserved bits for other functions. Another thing that is important in the control word was use of a 16-bit sequence number to support protocols like ATM and FR that require sequence integrity over tunneling protocols that may not always deliver packets in sequence (e.g., IP and MPLS).

Later versions of the “Martini” draft added support for more generic outermost and demultiplexing tunnel types. At the time of this writing, the PWE3 working group was working on defining standards for multiple services over MPLS, as well as a new version of L2TPv3, as the allowable tunnel types. The set of services being considered by PWE3 has also expanded beyond that in the “Martini” encapsulation to include circuit emulation and other forms of ATM and AAL. The ITU-T has even become involved in this standardization effort. We come back to topics related to the potential future direction and application of multi-service tunneling in Part 4, as well as in Part 8.



REVIEW

This chapter covered the enabling middleware for B-ISDN applications: the Common Part (CP) ATM Adaptation Layers (AALs). The text introduced AALs in terms of the service classes that specify functional attributes of constant or variable bit rate, the need for timing transfer, and the requirement for connection-oriented or connectionless service. The text then introduced the generic AAL protocol model in terms of the Convergence Sublayer (CS) and the Segmentation and Reassembly (SAR) sublayer, with a further subdivision of the CS sublayer into a Service-Specific (SS) part and a Common Part (CP). The chapter then detailed the formats and protocols for the currently standardized ATM adaptation layers: AAL1, AAL2, AAL3/4, and AAL5. We also provided an example of the operation of each AAL. The text also compared the trade-off between additional multiplexing capability and complexity involved with AAL3/4 and AAL5 through the use of two examples.

The chapter then provided an introduction to the emerging standards in the area of tunneling multiple services over MPLS and IP. We described the generic concept of protocol tunneling and gave the examples of the ATM Forum’s ATM and AAL5 over MPLS tunneling standard, and we reviewed some similar efforts occurring in the IETF at the time of this writing. Now that we’ve introduced the theoretical foundation of the physical, ATM, MPLS, and AAL layers, as well as the concept of tunneling, the next chapters move up the protocol stack to take in an overview of the higher layers and introduce the ATM and MPLS control planes.

CHAPTER 13



Higher-Level User and Control Plane Protocols

This chapter begins with an overview of the ATM- and MPLS-based user plane protocols that support higher-level applications, introducing the coverage in subsequent chapters as background for the role of control plane routing and signaling in support of these applications. We then introduce generic control plane concepts and specific terminology and protocols for ATM and MPLS covered in the rest of this part. The remainder of this chapter then focuses on the ATM control plane protocol as seen by an end user. ATM signaling operates in a manner similar to a telephone call. However, when reading this chapter keep in mind that most ATM-based applications are computer programs issuing these signaling messages in support of applications, and not human beings placing calls across a telephone network.

OVERVIEW OF HIGHER-LAYER ATM AND MPLS PROTOCOLS

The standards bodies listed in Chapter 3 active in the arena of ATM and MPLS either have defined or are in the process of defining support for a number of higher-layer protocols in the user plane in support of voice, video, emulation of TDM circuits, WAN protocols (e.g., FR), LAN protocols (e.g., Ethernet), and IP. As we shall see, many of these higher-layer user plane applications work hand in hand with control plane protocols to meet user application needs. In particular, a consistent theme in the ATM-based solutions is the *emulation* of connectionless data services through address resolution and fast circuit switching in a network of clients and servers designed to support specific end user protocols. On the other hand, for MPLS there are two different paradigms. The first, in support of non-IP protocols, is similar in some ways to ATM connection establishment and an adaptation layer protocol. The second applies to protocols designed to specifically support IP, where a tight coupling exists between the IP routing function and the MPLS signaling function, and connection establishment is control (or topology) driven, as opposed to data (or flow) driven. As motivation for the details of ATM and MPLS signaling and routing described in the remainder of this part, this section provides an overview of higher-layer protocols operating over ATM and MPLS covered in Part 4.

Circuit Emulation Voice, Video, and WAN Data Protocols

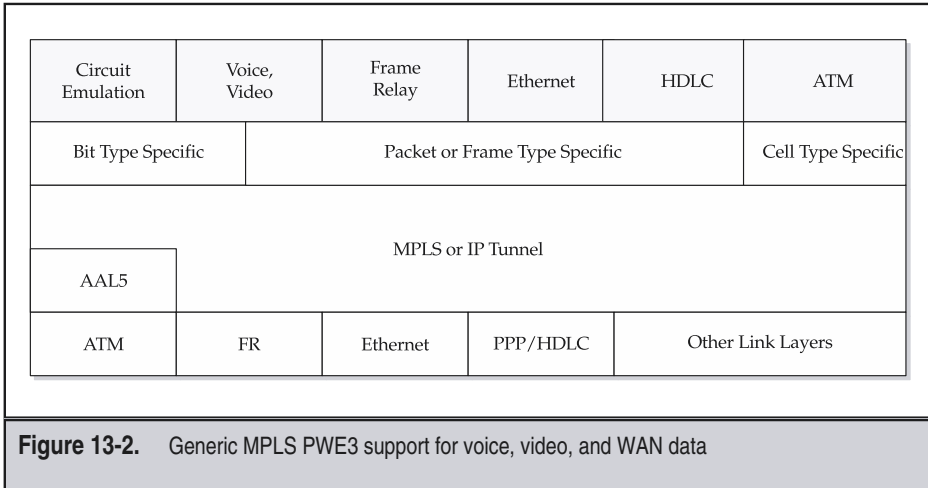
Figure 13-1 illustrates the ATM AAL SSSS and higher-layer user plane protocols covered in Chapters 16 and 17. Chapter 16 covers support of TDM circuit transport, voice, and video. Circuit emulation along with voice over ATM applications makes exclusive use of AAL1. An early ATM Forum video on demand specification also used AAL5 in conjunction with the real-time VBR ATM service category, since AAL2 was not standardized at

SMDS over ATM	ATM DXI, FUNI	ATM-Based Circuit Emulation	Voice, Video over ATM		ATM DXI, FUNI	FR Network Interworking	FR Service Interworking
AAL 3/4		AAL1	SSCS			FR-SSCS	
			AAL2		AAL5		
ATM							

Figure 13-1. ATM user plane protocols for voice, video, and WAN data

the time of development. Voice and video standards also make use of AAL2 SCS, as indicated in the figure. Chapter 17 covers higher-layer support for WAN data protocols over ATM, specifically, Frame Relay (FR), SMDS, the ATM Data Exchange Interface (DXI) and the ATM Frame-based UNI (FUNI). Note that the majority of these protocols utilize AAL5, while only ATM access to SMDS requires AAL3/4. ATM DXI and FUNI make support for AAL3/4 optional. The only protocol employing the FR-SSCS sublayer is FR/ATM network interworking, a protocol designed to support the trunking of Frame Relay over ATM as detailed in Chapter 17.

Figure 13-2 depicts the generic encapsulation architecture being considered in the IETF pseudo-wire edge-to-edge emulation (PWE3) working group for carriage of voice, video, and WAN data protocols over MPLS or IP tunnels. As described in Chapter 11, MPLS operates over a broad range of link layer protocols, as shown at the bottom of the figure. Above the MPLS layer, there are three generic categories of encapsulation for specific types of services being considered in PWE3. A bit type-specific service encapsulation supports emulation of TDM circuits and constant bit rate voice and video streams. A packet or frame type-specific service supports packetized voice or video for trunking applications, FR, Ethernet, and HDLC. An example of this is the “Martini” encapsulation described in the preceding chapter. Finally, there is a cell type specific encapsulation that is capable of supporting ATM over MPLS. An example here is the ATM Forum-defined encapsulation of ATM over MPLS described in the preceding chapter. Since few of these protocols were standardized at the time of writing, and there were several approaches under active discussion, we discuss only some high-level considerations involved with the bit-type encapsulation in Chapter 16, with Chapter 17 covering the packet- or frame-type encapsulation.



Local Area Networking and IP-Based Applications

Figure 13-3 illustrates the ATM-based user plane protocols covered in Chapters 18 and 19 that support LAN protocols (e.g., Ethernet) and IP. Notice that all of these data protocols operate over AAL5. Chapter 18 describes LAN Emulation (LANE) and multiprotocol encapsulation. Multiprotocol encapsulation performs a comparable function to AAL3/4 by multiplexing multiple network layer protocols (e.g., IP, IPX, Appletalk, DECnet, etc.) over a single ATM Virtual Channel Connection (VCC). RFC 2684 defines support for the Internet Protocol (IP) over ATM along with many other protocols. Chapter 19 describes how classical IP subnetworks work over ATM, as well as the protocol that implements a multicast capability over ATM.

Figure 13-4 illustrates the generic approach being taken by the IETF's Provider-Provided Virtual Private Network (PPVPN) working group at the time of this writing in support of LAN, layer 2, and IP-based VPNs over MPLS or IP tunnels. A pseudo-wire involves only point-to-point communication, while a VPN may involve many such point-to-point communications, or other topologies, such as broadcast or multicast. The network-based VPNs use one of two forms of routing, the first effectively aggregating routing for multiple VPNs through use of a single instance of a protocol (e.g., BGP), or the second where an instance of a routing protocol is (virtually) dedicated to each VPN. The far right-hand side of the figure shows IP, the transport protocols, and suite of applications that are in widespread use. These applications include voice, video, and circuit emulation, as well as many others. For background and references for more information on these protocols, see Chapter 8. Chapter 18 covers high-level considerations the layer 2-oriented VPNs, for example, Ethernet, FR, and ATM, since these were in the early stages of standardization at the time of this writing. Chapter 19 summarizes the standards work and de facto standard deployments in the area of network-based IP VPNs.

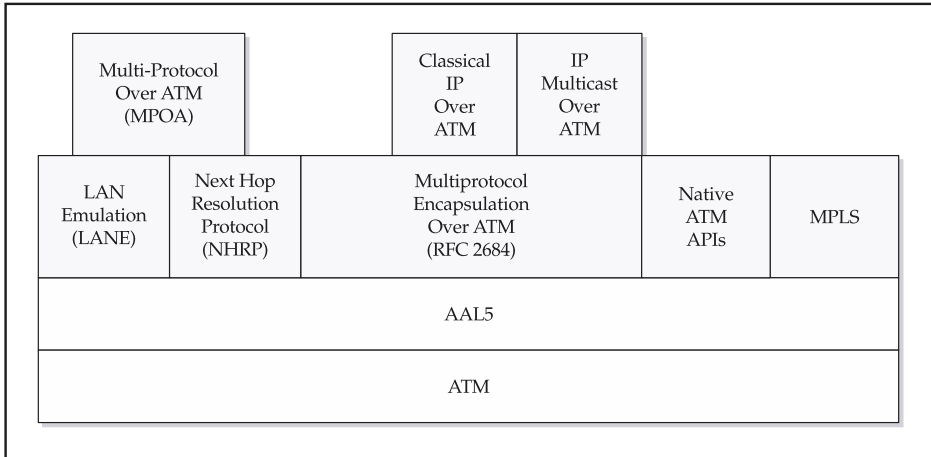


Figure 13-3. ATM support for LAN and IP-based applications

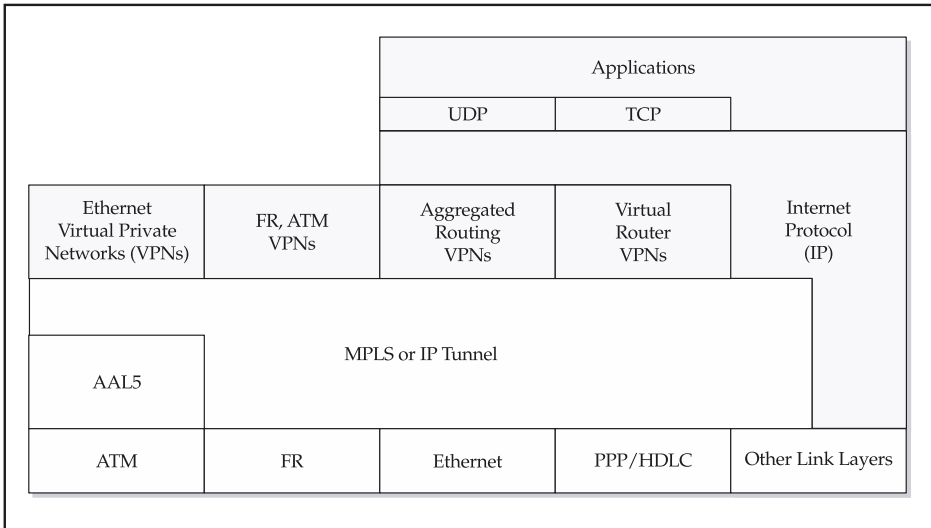


Figure 13-4. IP/MPLS higher-layer protocol support for LANs and VPNs

ATM Service Category and AAL Support for Applications

How does ATM support the different levels of quality and type of bit rate for all of these applications? The answer is a combination of the ATM Adaptation Layer (AAL) protocols and ATM service category introduced in Chapter 11. Table 13-1 illustrates applications that utilize particular combinations of AAL and ATM layer service category detailed in Part 5. As seen from the blank spaces in the table, application standards do not exist for every combination of AAL and ATM layer service category. Part 4 covers the higher-layer application protocols listed in this table, as introduced previously. A native ATM API together with basic signaling procedures can use any combination of AAL and ATM service category. However, most implementations currently use the standard combinations listed in Table 13-1, although PNNI and some proprietary protocols do directly access the ATM layer, as indicated in the column headed Null.

OVERVIEW OF ATM AND MPLS CONTROL PLANE PROTOCOLS

This section first gives a generic description of control plane protocols as background. We then summarize the context as well as the specific ATM and MPLS signaling and routing protocols described in Chapters 13 through 15.

Service Category	AAL1	AAL2	AAL3/4	AAL5	Null
CBR	Circuit emulation, voice, video			LANE 2.0, PNNI, IPoATM	PNNI
rt-VBR		Voice, video		Video, LANE 2.0, PNNI	PNNI
nrt-VBR			SMDS	FR, LANE 2.0, PNNI, IPoATM	PNNI
ABR				LANE 2.0, PNNI	PNNI
UBR				LANE, IPoATM	PNNI
GFR				FR, IPoATM	PNNI

Table 13-1. Combinations of AAL and ATM service category used by applications

Generic Control Plane Functions

The control plane handles connection switching, identifier/label distribution, path selection, admission control, and parameter communication. It is composed of one or more protocols that perform signaling and routing, which use addressing to identify specific entities requesting an ATM or MPLS service. The basic functions in the ATM and MPLS control protocols have a number of similarities yet some important differences.

For ATM and MPLS modes that are connection-like, there are many similarities in connection signaling and routing. An originator requests that the network make a connection to a specific destination identified by an address using the signaling protocol. The originator and/or the network determine the best path to the destination based upon the address using a routing protocol. The intermediate nodes attempt to establish the connection to the destination. The network then indicates the success or failure of the attempt back to the originator using the signaling protocol. Intermediate nodes or the destination may use signaling to either accept or reject the connection attempt. In any event, the signaling protocol informs the originator of whether the connection attempt succeeded or failed.

As described in Chapter 10, the origin of MPLS was to specifically support IP, and therefore some modes of IP over MPLS are not connection-oriented. In these cases, there is a control protocol that distributes labels, but there is no overall coordination as to how all labels must be distributed before data can flow. This is still a quite useful mode, since if MPLS forwarding is not established, then the intermediate LSRs can simply use IP forwarding.

Switched and Permanent ATM Virtual Connections

ATM uses one of three basic methods to establish a connection: a Permanent Virtual Circuit (PVC), a Switched Permanent Virtual Connection (SPVC), or a Switched Virtual Connection (SVC). A PVC is configured by a network management function to establish or remove the connection, often with proprietary methods. An SVC involves a network that allows a user application to signal for the dynamic establishment and release of connections. An SPVC is in a sense the combination of these, by initiating the establishment and release a connection by management actions, while utilizing the dynamic signaling protocol also used for SVCs. These connection establishment techniques all involve three basic processes: connection request signaling, admission control, and routing.

In switched ATM networks, users signal switches, which in turn signal other switches, which in some cases signal other networks. ATM switches and users employ different signaling protocols for each of these contexts. Users interface to switches and communicate the connection request information via a *User-Network Interface (UNI)* signaling protocol. Networks interconnect via a more complex *Network-Network (NNI)* signaling protocol. Switches employ an interswitch signaling protocol, usually based upon an NNI protocol, which is sometimes called a *Network-Node Interface (NNI)* as well. These interswitch signaling protocols frequently employ vendor proprietary

extensions. Private and public networks may use different NNI signaling protocols because of different business needs. Private switched networks usually connect to public switched networks via UNI signaling.

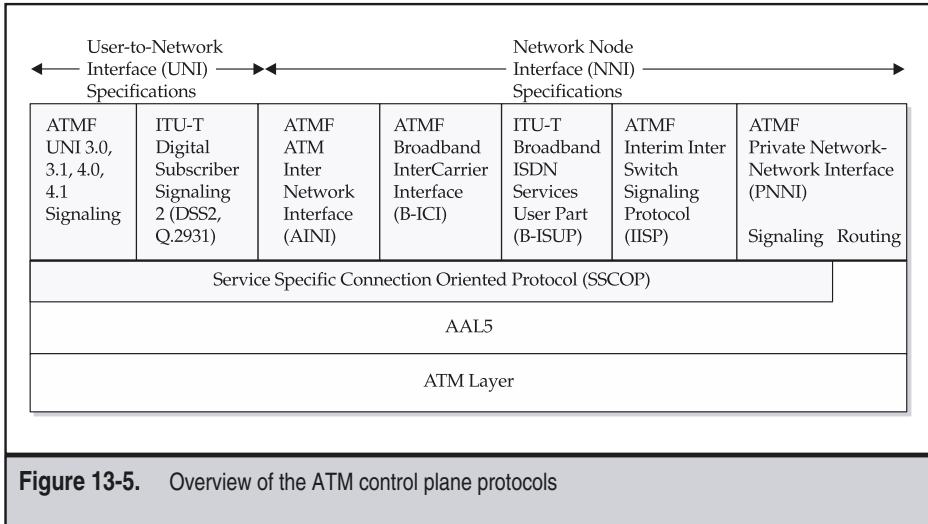
Many implementations utilize standards-based ATM UNI and NNI signaling protocols to establish SVC or SPVC connections across a network. Also, some implementations employ network management protocols to emulate signaling functions by making individual ATM cross-connects (i.e., VP or VC links) at each switch along a particular route to build up an end-to-end VPC or VCC PVC. These centralized network management approaches generally operate at a much slower connection setup rate than distributed signaling protocol implementations. This can result in unacceptable performance if a large number of connections must be restored in response to a failure. On the other hand, centralized control may provide other features not defined in the standard distributed signaling and routing protocols.

As covered in Part 5, many higher-layer data protocols operating over ATM can require large connection establishment rates because SVC establishment and release are driven by the need to transfer data between specific endpoints. For example, in the LAN Emulation protocol, a LAN station potentially sets up an SVC for each LAN address that it communicates with. Although each user typically sets up a few connections to various servers per unit time, the aggregate call rate scales roughly with the number of attached LAN users. The ATM signaling architecture responds to this challenge by distributing intelligence to each device, eliminating the bottleneck of centralized control.

ATM Control Plane Protocols

The ATM control plane provides the means to support establishment and release of either an SVC or an SPVC on behalf of the user plane for a point-to-point, or point-to-multipoint, VPC or VCC, as defined in Chapter 11. These protocols determine the path taken by a switched VPC or VCC. They also perform admission control to ensure that the requested QoS is met for cells sent at a rate no greater than that specified by ATM traffic parameters, as detailed in Chapter 20.

The shaded area of Figure 13-5 illustrates the ATM UNI control plane protocols detailed in this chapter as well as the NNI protocols described in Chapter 15. The specifications for the Service-Specific Connection-Oriented Protocol (SSCOP) provide a guaranteed, reliable packet delivery service to all ATM signaling protocols. This chapter describes SSCOP as background and then covers the signaling protocols at the User-to-Network Interface (UNI). The ATM Forum has produced four versions of UNI signaling protocols, numbered 3.0, 3.1, 4.0, and 4.1. ITU-T Recommendation Q.2931 specifies B-ISDN signaling on the ATM UNI, and to some extent the ATM Forum and the ITU-T have developed added functionality in parallel and by using each other's work. Therefore, the ATM Forum UNI and Q.2931 specifications are closely aligned as described in Chapter 14. Note that the ITU-T's formal name for the ATM UNI signaling protocol is the Digital Subscriber Signaling System 2 (DSS2), indicating it as the next evolutionary step after the DSS1 signaling used for N-ISDN.

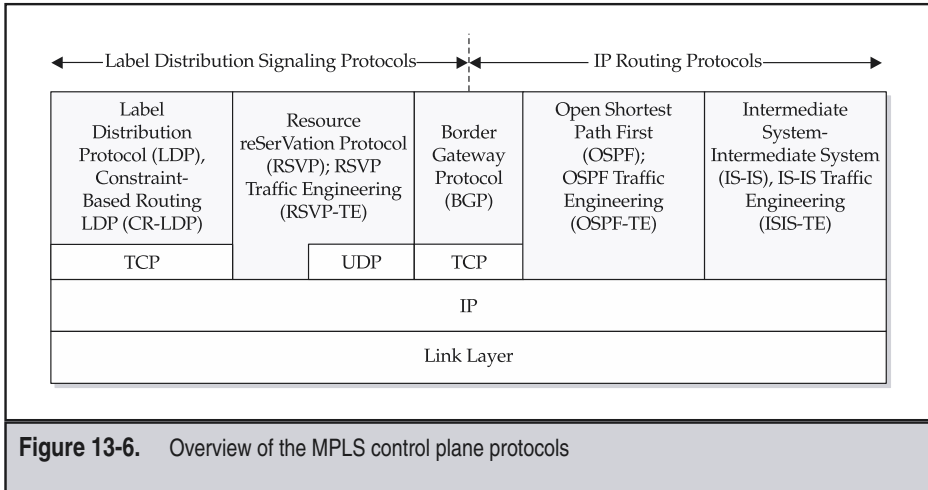


MPLS Control Plane Protocols

The shaded area of Figure 13-6 illustrates the MPLS-related signaling and routing protocols described in Chapter 14 that work together to establish MPLS label switched paths (LSPs). Shown on the left-hand side are the label distribution signaling protocols. The label distribution protocol (LDP) and its constraint-based routing counterpart (CR-LDP) both require the reliable TCP transport protocol. On the other hand, the Resource Reservation Protocol (RSVP), along with its traffic engineering extensions (RSVP-TE), does not require a reliable transport layer and can run directly over IP or UDP. The border gateway protocol (BGP) shown in the middle of the figure can play a role as either a signaling or routing protocol running over TCP. The right-hand side of the figure shows the principal IP interior gateway protocols (IGPs), namely the Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) protocols with or without traffic engineering extensions. There is also a case similar to ATM PVCs where the label switching at each hop can be established by a management system, as discussed in Chapter 27. However, for the most part, MPLS runs in an IP network.

ATM CONTROL PLANE STRUCTURE AND AAL

This section begins by introducing the ITU-T B-ISDN signaling protocol stacks, and then it starts from the bottom and moves up the stack. At the most basic level, there are several methods to associate signaling with physical interfaces and ranges of ATM VP and VC identifiers. We also summarize the low-level service-specific protocols that make AAL5 work with B-ISDN and ATM signaling protocols.



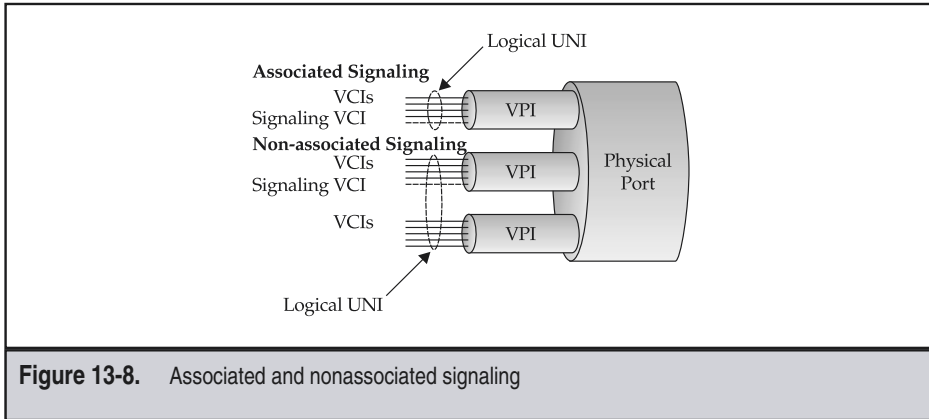
ITU-T B-ISDN Signaling Protocols

The B-ISDN control plane handles all virtual connection-related functions, most importantly, SVCs. The higher-layer and service-specific AAL portions of the signaling protocol are now well standardized. This section summarizes the B-ISDN UNI and NNI signaling protocols and also covers the protocol model for the signaling AAL's Service Specific Convergence Sublayer (SSCS).

Figure 13-7 illustrates the relationships between the major ITU-T signaling standards. The left-hand side of the figure shows the B-ISDN User-Network Interface (UNI) signaling protocol stack. As shown in the center of the figure, the B-ISDN Network Node Interface (NNI) interconnects public networks but is sometimes used between switches within a single network. The right-hand side of the figure also shows the N-ISDN interworking function (IWF) along with the N-ISDN UNI signaling protocol stack. We now cover the B-ISDN and UNI signaling protocol stacks.

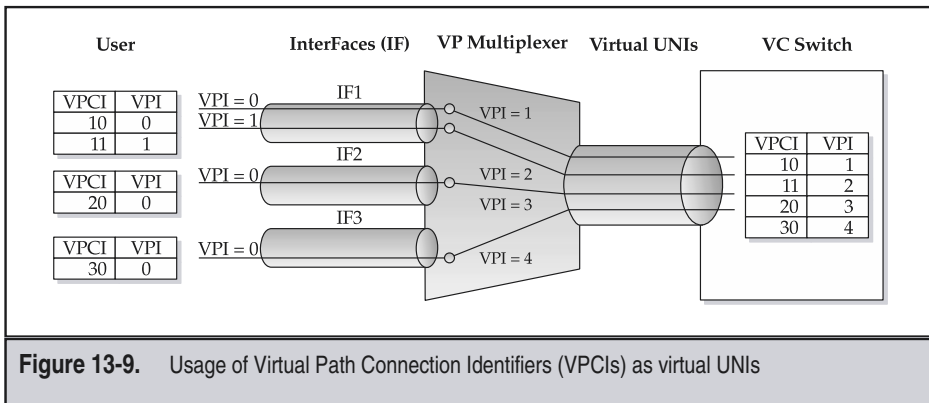
ITU-T Recommendation Q.2931 specifies signaling over the B-ISDN UNI. It borrows heavily from both the Q.931 UNI signaling protocol for N-ISDN and the Q.933 UNI signaling protocol for Frame Relay. ITU-T Recommendation Q.2130 specifies the Service Specific Coordination Function (SSCF) for the UNI. ITU-T Recommendation Q.2110 specifies the Service Specific Connection-Oriented Protocol (SSCOP), as described later.

The ITU-T standards adapt the N-ISDN User Part (ISUP) concept for supporting B-ISDN UNI signaling between networks at an NNI using a protocol called B-ISUP. The B-ISUP protocol operates over the Message Transfer Protocol 3 (MTP3), identical to that used in Signaling System 7 (SS7) for out-of-band voice and N-ISDN signaling. This choice of standards will allow B-ISDN network signaling the flexibility to operate over existing



described in the text that follows. The VPI/VPCI(s) controlled by a signaling channel are sometimes called a *logical UNI*.

Figure 13-9 illustrates an example of nonassociated signaling. In the example, a VP multiplexer maps the VCI=5 signaling channel to different VPI values, each with a different VPCI into a VC switch. The end user, the VP multiplexer, and the VC switch must be configured with compatible VPCI mappings (in this case physical interface identifier [IFn] plus the user VPI) as indicated in the figure. Looking inside the VC switch, VPCI=11 corresponds to physical interface IF1 and user VPI=1 on the user side of the VP multiplexer, which the VC switch sees on VPI=2 on the port from the VP multiplexer. Annex 8 of the ATM Forum UNI 4.1 signaling specification also defines a *virtual UNI* capability where only the VP multiplexer and the VC switch utilize the VPCI. In this case, the end



user employs associated signaling as defined in UNI 3.1 where the user does not use the VPCI concept. Annex 2 of the Forum's UNI signaling 4.1 specification also defines a *proxy signaling* capability where a single signaling interface allows a processor to control multiple VPCIs. This capability allows a proxy agent to set up connections on behalf of user devices that do not have signaling capability.

When the proxy capability was added to UNI 4.0, it specified the use of a virtual connection into a switch to carry signaling information between the proxy controller and the switch, but left the establishment and maintenance of this connection as implementation dependent. At the time of writing, a draft ATM Forum UNI Proxy SVC addendum further detailed how to automatically interconnect a switch with a proxy agent controller using an SVC in a standard manner, providing enhanced resilience and the potential to provide advanced user screening and authorization procedures on the remote controller. Furthermore, this configuration allows invocation of an additional feature in UNI 4.1; the ability to connect an SVC or SPVC to any VPI/VCI combination (including values reserved for control signaling) on an ATM or FR/ATM interworking UNI port.

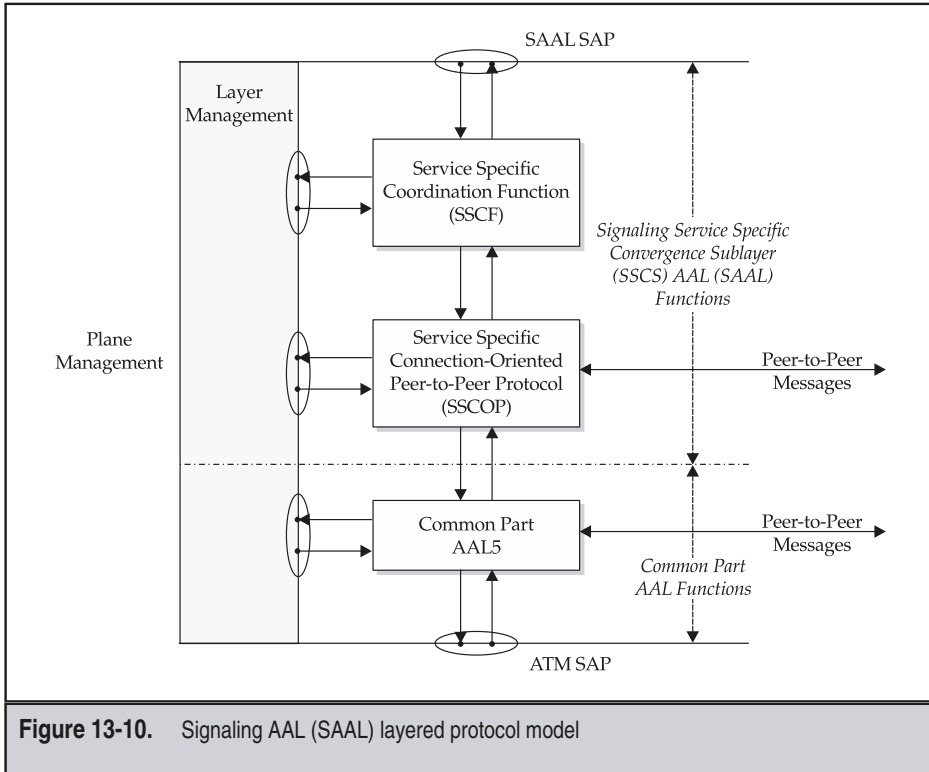
Layered Signaling AAL Model

Figure 13-10 illustrates the protocol model for the Signaling AAL (SAAL) specified in ITU-T Recommendation Q.2100. The Common Part AAL (CP-AAL) is AAL5 as described in Chapter 13. The following two protocols comprise the SSCS portion of the SAAL: Service Specific Coordination Function (SSCF), and Service Specific Connection-Oriented Protocol (SSCOP), as described later.

The SAAL primitives define services at the SAAL Service Access Point (SAP). The CP AAL5 interfaces with the ATM layer at the ATM SAP. A one-to-one correspondence exists between an SAAL SAP and an ATM SAP. Corresponding layer management functions manage the signaling SSCF and SSCOP protocols and the CP-AAL as separate layers as indicated on the left-hand side of Figure 13-10. Layer management sets parameters in the individual layer protocols; for example, timers and threshold, as well as monitoring their state and performance [ITU Q.2144]. For example, layer management may use the state of SSCOP to determine the state of the underlying physical link or virtual connection between two ATM devices. Plane management coordinates across the layer management functions to monitor and maintain the overall end-to-end signaling capability.

Service Specific Coordination Function (SSCF)

The Service Specific Coordination Function (SSCF) provides services to the Signaling AAL (SAAL) independent of underlying layers for the transparent relay of information using the choice of an unacknowledged data transfer mode or an assured data transfer mode. The SSCF provides these capabilities primarily by mapping between a simple state machine for the user and the more complex state machine employed by the SSCOP protocol. ITU-T Recommendation Q.2130 defines the SSCF at the UNI, while Recommendation Q.2140 defines the SSCF at the NNI.



Service Specific Connection-Oriented Protocol (SSCOP)

ITU-T Recommendation Q.2110 defines the Service Specific Connection-Oriented Protocol (SSCOP) serving both the UNI and NNI SSCF functions. SSCOP is a sophisticated link layer, peer-to-peer protocol that performs the following functions:

- ▼ Guaranteed sequence integrity, that is, in sequence message delivery
- Error correction via error detection and selective retransmission
- Receiver-based flow control of the transmitter
- Protocol level error detection and error reporting to layer management
- Keep alive messaging during intervals of no data transfer
- Local retrieval of unacknowledged or queued messages
- Capability to establish, disconnect, synchronize, and report status for SSCOP connections
- ▲ Transfer of user data in either an unacknowledged or assured mode

SSCOP is a complex protocol, but it is specified at the same level of detail as a successful protocol like HDLC. As the name implies, a connection must be established *before* any data transfer occurs. The unacknowledged mode is a simple unacknowledged datagram protocol, similar to the User Datagram Protocol (UDP) in the IP protocol suite. Much of the complexity of SSCOP occurs in the assured data transfer mode. SSCOP uses a number of message types to perform the described functions. See Q.2110 for details.

Figure 13-11 illustrates an example of the SSCOP selective retransmission strategy. First, the error detection capability of AAL5 reliably determines whether the destination signaling node receives a frame successfully. SSCOP requires that the transmitter periodically poll the receiver as a keep-alive action, as well as a means to detect gaps in the sequence of successfully received frames. The receiver must respond to the poll, and if more than a few poll responses are missed, the transmitter takes down the connection. A key feature is where the receiver identifies that one or more frames are missing in its sequence, as illustrated in the figure. The transmitter then resends only the missing frames. Chapter 25 shows how this selective retransmission protocol significantly improves throughput when compared with “Go-Back N” retransmission strategy, such as those employed in X.25 and

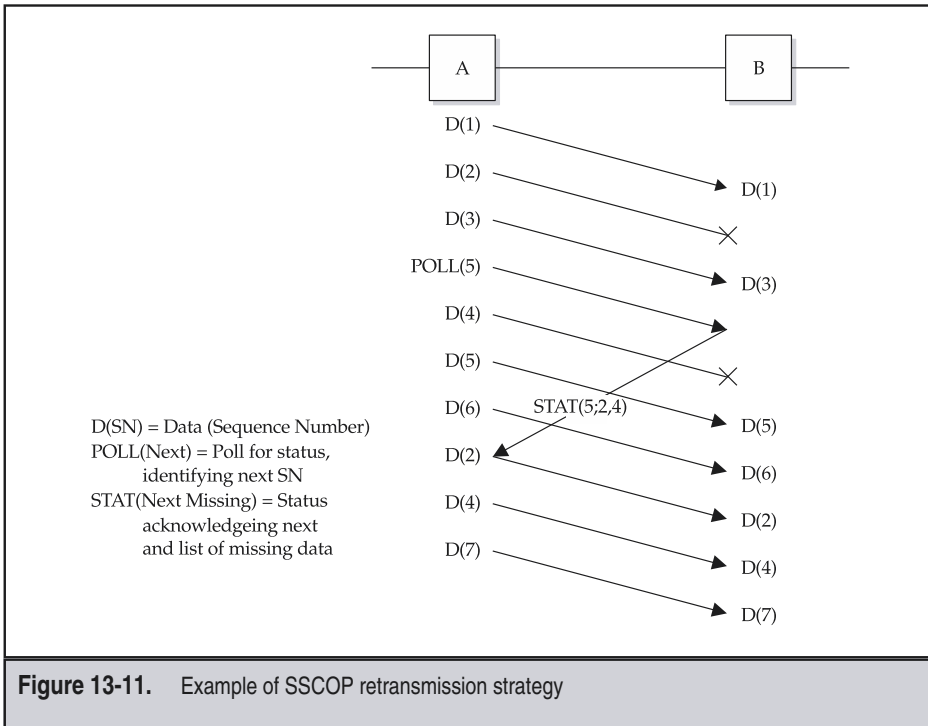


Figure 13-11. Example of SSCOP retransmission strategy

TCP. SSCOP PDUs employ a 24-bit sequence number that achieves high throughputs on very high speed links, such as those typically used in ATM networks.

ITU-T Recommendation Q.2111 specifies extensions to SSCOP for operation over a multilink or connectionless network (e.g., Ethernet or IP). The extensions involve a straightforward mapping to IP or UDP datagrams, as well as the means to more efficiently handle out-of-sequence packets, links with variable QoS, or other anomalies of a connectionless networking environment. At the time of this writing, the ATM Forum was also specifying SSCOP operation over something other than AAL5 for the use of MPLS/ATM Network interworking and PNNI for optical network control.

ATM USER-NETWORK INTERFACE (UNI) SIGNALING

As described in Chapter 5, ATM signaling shares many characteristics with basic telephony, with extensions that add the capabilities to specify bandwidth, Quality of Service, various end system attributes, different connection topologies, and address formats. This section describes the basic functions of the ATM UNI signaling protocol standards. The text presents a comparison of ATM Forum and ITU-T signaling standards. Next, we introduce the basic signaling message types and review the role of some key information elements in these messages.

Base Signaling Functions: Q.2931 and UNI 3.1

Released in 1994, the ATM Forum UNI 3.1 specification is the oldest version of ATM signaling specification that has some degree of interoperability with ITU-T Recommendation Q.2931 and backward interoperability with ATM Forum UNI 4.0 and 4.1. The older ATM Forum UNI 3.0 specification is not interoperable with any of the modern signaling specifications. The principal ATM signaling functions defined in the ATM Forum UNI specifications and ITU-T Recommendation Q.2931 are:

- ▼ Point-to-point and point-to-multipoint SVC establishment and release
- Identification of called (and optionally calling) party using E.164 or NSAP formatted addresses
- VPI/VCI selection and assignment
- ▲ Communication of traffic and higher-layer protocol parameters

ATM Forum UNI Signaling 4.0 and ITU-T Standards

Successive ATM Forum UNI versions increasingly aligned with the ITU-T Q series of standards in the specification of control plane functions. Table 13-2 compares the capabilities defined in the ATM Forum UNI 3.1, 4.0, and 4.1 signaling specifications and those defined in the applicable ITU-T Q Series Recommendations. Note that some ATM Forum UNI 4.1 capabilities are not yet standardized by the ITU-T, while a few ITU-T capabilities are also not addressed by the ATM Forum 4.1 specification.

Capability Description	ATM Forum UNI 3.1	ATM Forum UNI 4.0	ATM Forum UNI 4.1	ITU-T Recommendation
Point-to-point calls (SPVC)	No	No	Yes	No
N-ISDN signaling interworking	No	No	No	Q.2931
Signaling of individual QoS parameters	No	Yes	Yes	Q.2965.2
ATM anycast	No	Yes	Yes	No
ABR signaling for point-to-point calls	No	Yes	Yes	Q.2961.3
Generic identifier transport	No	Yes	Yes	Q.2941
Switched Virtual Path (VP) service	Yes	Yes	Yes	Q.2934
Proxy signaling and virtual UNIs	No	Yes	Yes	No
Frame discard	No	Yes	Yes	No
Traffic parameter modification during active calls	No	No	Yes	Q.2963
Traffic parameter negotiation during call setup	No	Yes	Yes	Q.2962
Supplementary services	No	Yes	Yes	Q.2951.1-8
User to User Signaling (UUS)	No	Yes	Yes	Q.2957.1
Domain-based rerouting	No	Yes	Yes	No
Guaranteed Frame Rate (GFR)	No	Yes	Yes	No
OAM traffic descriptor	No	No	Yes	Q.2931
Security	No	Yes	Yes	No
Network call correlation identifier	No	Yes	Yes	No
UBR with BCS or MDCR	No	Yes	Yes	No
PHY/MAC identifier	No	No	Yes	No

Table 13-2. Comparison of ATM Forum and ITU-T UNI Signaling Capabilities

Supplementary services added in UNI 4.0 include Direct Dialing In (DDI), Multiple Subscriber Number (MSN), Calling Line Identification Presentation/Restriction (CLIP/CLIR), Connected Line Identification Presentation/Restriction (COLP/COLR), subaddressing as specified in ITU-T Recommendation Q.2951, and User-to-User Signaling (UUS) as specified in ITU-T Recommendation Q.2957.1. The UNI 4.0 specification also added support for the end-to-end Transit Delay Information Element, which allows the calling party to request bounded delay for the connection. Furthermore, UNI 4.0 adopted a number of ITU-T conventions in support of N-ISDN interworking.

Anycast was also added in UNI 4.0. An anycast address identifies a particular service, and not a specific node as described later in this chapter. The UNI 4.0 signaling specification supports a limited form of connection parameter negotiation at call setup time. The user may include, in addition to the desired traffic descriptor, either the minimum acceptable or an alternative traffic descriptor in the SETUP message. The network responds indicating whether it granted either the original traffic descriptor or the user-specified minimum/alternative. This response is important to applications, such as video conferencing, that operate best with a preferred bandwidth but can “step down” to a lower bandwidth in a manner similar to automatic modem speed negotiation dependent upon line quality. UNI 4.0 also added support for switched VPs, and additional signaling information elements allow the end user to explicitly specify QoS parameters.

ATM Forum UNI Signaling 4.1 and ITU-T Standards

The current version of the ATM Forum UNI Signaling Specification Version 4.1 is primarily a consolidation of known corrections to errors, updates to ITU references, and consolidated specifications of new features, many already documented as addenda to UNI 4.0. Notably, the addressing information in UNI 4.0 is replaced with references to the ATM forum addressing documents [AF ADDR REF, AF ADDR GUIDE]. The status enquiry is enhanced with procedures to support several calls with a single request. Note also that the ABR setup parameter and ABR additional parameters information elements have been renamed to make the name less confusing, as those IEs are also used with GFR. The negotiation procedures for the minimum ATM traffic descriptor have been enhanced to apply to all traffic parameters, including the MBS and SCR parameters, and the general VPCI/VCI assignment procedures have been clarified in relation to the proxy signaling feature and the enhancements mentioned previously. UNI 4.1 also supports Soft PVCs, which was previously only an NNI capability.

With so many different ATM Forum UNI specifications, additional UNI addenda, and other specifications from the ATM Forum and other standards bodies, the ATM Forum was finalizing an architecture specification that defines the set of specifications that apply to a UNI at the 4.1 level, which includes:

- ▼ UNI 4.1 signaling specification [AF UNI 4.1]
- Integrated Local Management Interface (ILMI), Version 4.0 [AF ILMI 4.0]
- ITU-T Recommendation I.610 including amendments
- Traffic Management Specification Version 4.1 [AF TM 4.1]
- Addendum to Traffic Management 4.1: Differentiated UBR [AF DIFF 1.0]
- ▲ Addendum to Traffic Management 4.1, Optional Minimum Desired Cell Rate Indication for UBR [AF MDCR 1.0]

These are the main specifications, but not the full list. The full list determines a compliant UNI 4.1 implementation. Other specifications that apply, the numerous physical layer specifications, for example, are referenced in the architecture specification. The

main reason to establish a list like this is because the specifications that apply to UNI 4.1 do not all have the same revision number (e.g., ILMI 4.0). The UNI 4.1 signaling specification also defines a virtual UNI, as described earlier.

UNI 4.1 Signaling Message Types

ATM Forum UNI 4.1 uses the message types shown in Table 13-3 for point-to-point and point-to-multipoint connections. The table groups the point-to-point messages according to function: call establishment, call release (or clearing), status, and layer 2 signaling link management. The point-to-multipoint messages support the procedures for adding and dropping root-initiated calls. The next section illustrates the use of many of these messages through examples of call establishment and release.

Point-to-Point Connection Control	Point-to-Multipoint Connection Control
Call Establishment Messages	ADD PARTY
ALERTING	ADD PARTY ACKNOWLEDGE
CALL PROCEEDING	ADD PARTY REJECT
CONNECT	PARTY ALERTING
CONNECT ACKNOWLEDGE	DROP PARTY
SETUP	DROP PARTY ACKNOWLEDGE
CONNECTION AVAILABLE CONFIRM	
Call Clearing Messages	
RELEASE	
RELEASE COMPLETE	
Status Messages	
STATUS ENQUIRY	
STATUS (Response)	
NOTIFY	
Signaling Link Management	
RESTART	
RESTART ACKNOWLEDGE	

Table 13-3. UNI 4.1 Signaling Message Types

Signaling Message Information Elements

Each UNI 4.1 signaling message has a number of Information Elements (IEs), some of which are mandatory and others of which are optional as indicated in standards. Q.2931 and UNI 4.1 have many information elements in common, but each also has some unique elements. In fact, the ATM Forum and the ITU coordinate the assignment of IE code points, thereby facilitating interoperability. The following discussion covers the principal information elements used in ATM signaling messages.

Note that all messages related to a particular call attempt each contain a common mandatory information element, the *call reference*, which must be unique to a signaling channel. Also, every message must also contain an information element for their type, length, and protocol discriminator (i.e., the set from which these messages are taken). This structure supports variable-length messages and the addition of new message types in the future as needed. The following narrative highlights some of the key information elements and their usage.

The SETUP message contains the majority of the information elements because it conveys the user's request to the network. The key mandatory information elements used in the SETUP message are: called party number, Broadband Bearer Capability, and ATM traffic descriptor.

The called party number may be either an NSAP-based or E.164 ATM address as defined later in this chapter. The Broadband Bearer Capability specifies the ATM service category introduced in Chapter 11 and detailed in Chapter 20, namely CBR, rt-VBR, nrt-VBR, UBR, GFR, or ABR. The ATM traffic descriptor defines the parameters of the traffic contract, such as PCR, SCR, and MBS, in both the forward and backward directions. The traffic descriptor also determines whether the network tags cells with the CLP bit for inputs exceeding the contract per ITU-T Recommendation Q.2961.1.

The connection identifier gives the value of the VPI (and VCI) for a switched VPC (or VCC) connection. The user may request a specific value in the SETUP message, or accept a value assigned by the network in an ALERTING, CALL PROCEEDING, or CONNECT message.

The cause (code) specified in ITU-T Recommendation I.2610 provides important diagnostic information by indicating the reason for releasing a call. The cause IE must be present in the first message involved in call clearing. For example, the cause indicates whether the destination is busy, if the network is congested, or if the requested traffic contract or service category isn't available.

The key attributes of the some other optional parameters not already described are as follows. The root point-to-multipoint call procedures utilize the endpoint reference identifier and endpoint state number information elements. The Available Bit Rate (ABR) parameters detail the requested (and granted) service defined by the ATM Forum as described in Chapter 22. The AAL parameters along with the broadband low- and high-layer information elements convey information between end users about the end systems. Only the end user may employ the called and calling party subaddress information elements to convey additional addressing information across the network. The request of a specific service category is somewhat more complex in UNI 4.0 and 4.1, since no explicit service category is signaled. In UNI 3.1, the Broadband Bearer Capability IE

indicated the “traffic type,” e.g., CBR and VBR, and the “timing” requirements. The IE was changed in UNI 4.0, and now the service category is derived from information in the Broadband Bearer Capability IE (containing the broadband bearer class and the ATM transfer capability), and the ATM Traffic Descriptor IE (containing the best effort indicator). Table A9-1 in UNI 4.1 summarizes the different combinations of this information that now define ATM service categories. This table also indicates how this correlates to the old UNI 3.0/3.1 information.

In UNI 3.0/3.1, the QoS parameter IE would indicate the QoS class (values 0 through 4), and although this is still possible to do, the QoS parameter IE is ignored in UNI 4.0 and UNI 4.1 when the Extended QoS parameter IE is present. The Extended QoS parameter IE and the end-to-end Transit Delay parameter IE are used to signal the desired conformance definition for the call, as discussed in Chapter 20. Note that the end-to-end transit delay parameters IE can be used in conjunction with the “old” QoS parameter IE, and it then supersedes all other delay information that may be present in the QoS parameter IE. The Transit Network Selection (TNS) IE allows an end user to specify the desired network provider. The STATUS message uses the call state IE to indicate the current condition of the referenced call or endpoint in response to a STATUS ENQUIRY message, and with the enhanced procedures, it allows several calls to be included in one status request.

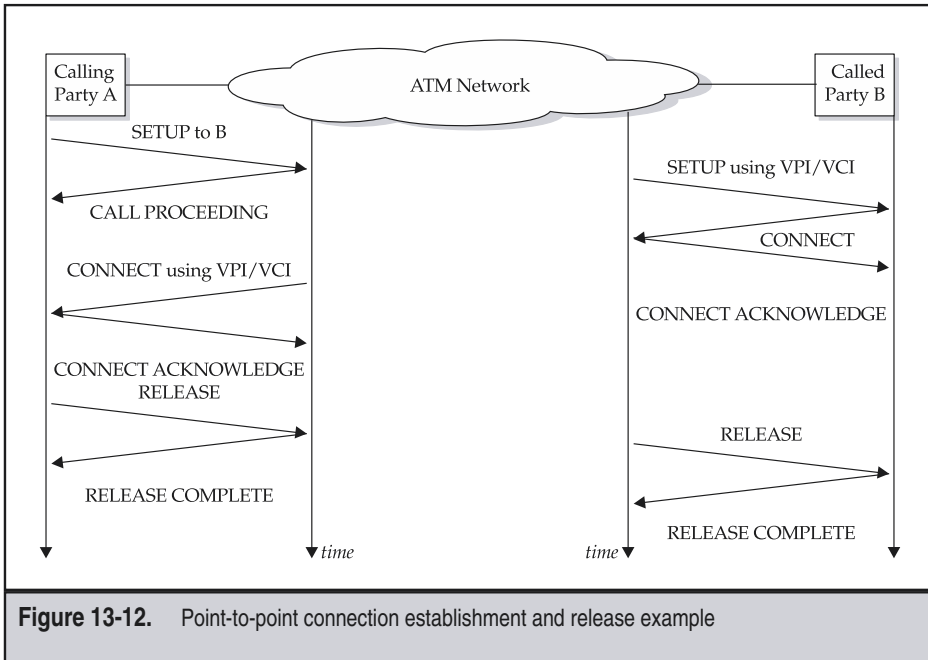
Examples of ATM Signaling Procedures

Signaling procedures specify the valid sequence of messages exchanged between a user and the network, the rules for verifying consistency of the parameters, and the actions taken to establish and release ATM layer connections. A significant portion of signaling standards and specifications handle error cases, invalid messages, inconsistent parameters, and a number of other unlikely situations. These are all important functions, since the signaling protocol must be highly reliable to support user applications. This section gives an example of signaling procedures for calls using the message types and information elements described previously for point-to-point connection establishment and release, and root-initiated point-to-multipoint connection establishment.

Standards specify signaling protocols in several ways: via narrative text, via state machine tables, or via a semi-graphical Specification Definition Language (SDL). The ATM Forum UNI specifications and the ITU-T Q.2931 use the narrative method as well as the SDL technique. For complicated protocols, such as Q.2931, a very large sheet of paper would be needed to draw the resulting state machine in a manner such that a magnifying glass would not be required to read it. The SDL allows a complicated state machine to be formally documented on multiple sheets of paper in a tractable manner. In fact, ITU-T Recommendation Q.2931 dedicates almost one hundred pages of SDL diagrams out of a total of 250 pages.

Point-to-Point Connection

Figure 13-12 illustrates the point-to-point connection establishment example. This example employs: a calling party with ATM address A on the left, a network shown as a cloud in the middle, and the called party with ATM address B on the right. Time runs from top to



bottom in all of the examples. Starting from the upper left-hand side of the figure, the calling party initiates the call attempt using a SETUP message indicating B in the called party number IE. Recommendation Q.2931 requires that the network respond to the SETUP message with a CALL PROCEEDING message as indicated in the figure. The network routes the call to the physical interface connected to B and outputs a SETUP message indicating the specified VPI/VCI values in the connection identifier IE. Optionally, the SETUP message may also communicate the identity of the calling party A in the calling party number IE, similar to the calling line ID service in telephony. If the called party chooses to accept the call attempt, it returns the CONNECT message, which the network propagates back to the originator as rapidly as possible in order to keep the call setup time low. Optionally, the called party user may respond with either a CALL PROCEEDING or an ALERTING message prior to sending the CONNECT message; however, unlike the network side, Recommendation Q.2931 does not require the user side to respond to a SETUP with the CALL PROCEEDING or ALERTING message. Both the user and network sides confirm receipt of the CONNECT message by sending the CONNECT ACKNOWLEDGE message as shown in the figure.

The bottom part of Figure 13-12 illustrates the point-to-point connection release example, or in other words the process used to hang up the call. Using the same reference configura-

tion and conventions are the same as in the point-to-point connection establishment example. Either party may initiate the release process, just as either party may hang up first in a telephone call. This example illustrates the calling party as the one that initiates the disconnect process by sending the RELEASE message. The network then propagates the RELEASE message across the network to the other party B. The network also responds to A with a RELEASE COMPLETE message as indicated in the figure. The other party acknowledges the RELEASE request by returning a RELEASE COMPLETE message. This two-way handshake completes the call release process.

Recall from Chapter 11 that a VCC or VPC is defined in only one direction; that is, it is simplex. A point-to-point duplex SVC (or a SPVC) is actually a pair of simplex VCCs or VPCs: a forward connection from the calling party to the called party, and a backward connection from the called party as illustrated in Figure 13-13. Applications may request different forward and backward traffic parameters and ATM service categories. For example, a file transfer application might set up an SVC with the forward direction having ten times the bandwidth as the backward direction; since the backward channel is used only for acknowledgments. A video broadcast might specify large forward traffic parameters with zero backward bandwidth.

Thus, the forward and backward VPIs (and VCI for a VCC), as well as the ATM address associated with the physical ATM UNI ports at each end of the connection, completely define a point-to-point SVC shown in Figure 13-13. Furthermore, the VPI and VCI assignments may differ for the forward and backward directions of a VPC or VCC at the same end of the connection, as well as being different from the other end of the connection, as illustrated in the figure. In the case of a VCC, the VPI value is often zero. A convention where the VPI (and VCI for a VCC) is identical at the both ends of a connection may be used; this is a common implementation method for PVCs because it simplifies operation of ATM networks. Since the SVC procedures dynamically assign the VPI (and VCI for VCCs), the values generally differ for each end of the connection.

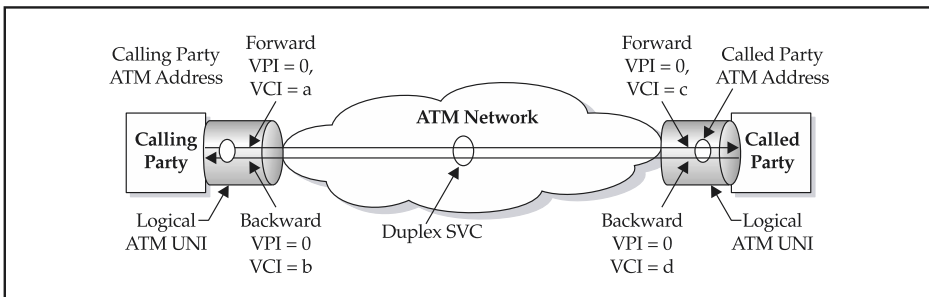


Figure 13-13. Point-to-point switched virtual connection

Point-to-Multipoint Connection (Add Party)

The ATM Forum UNI 3.1 specification first defined the “root” initiated point-to-multipoint connection capability in 1994, as standardized in ITU-T Recommendation Q.2971 one year later. UNI 4.0 signaling retained this capability and added a “Leaf” Initiated Join (LIJ) procedure that was removed in the UNI 4.1 specification because it was too complex to support at the NNI. The root-initiated point-to-multipoint connection process is similar to that of three-way telephone calling, where a conference leader (the root) adds other parties to an existing call. The root-initiated point-to-multipoint connection procedure meets the needs of broadcast audio, video, and data applications, for example, IP multicast over ATM, as described in Chapter 19.

Figure 13-14 illustrates an example of a root node setting up a point-to-multipoint call from an originator (root) node A to two leaf nodes B and C connected to a local ATM switch on a single ATM UNI, and a third leaf node D connected to a separate ATM UNI. In the example, root node A begins the point-to-multipoint call by sending a SETUP message to the network requesting setup of a point-to-multipoint call identifying leaf node B’s ATM address. The network responds with a CALL PROCEEDING message in much the same way as a point-to-point call. The network switches the call attempt to the intended

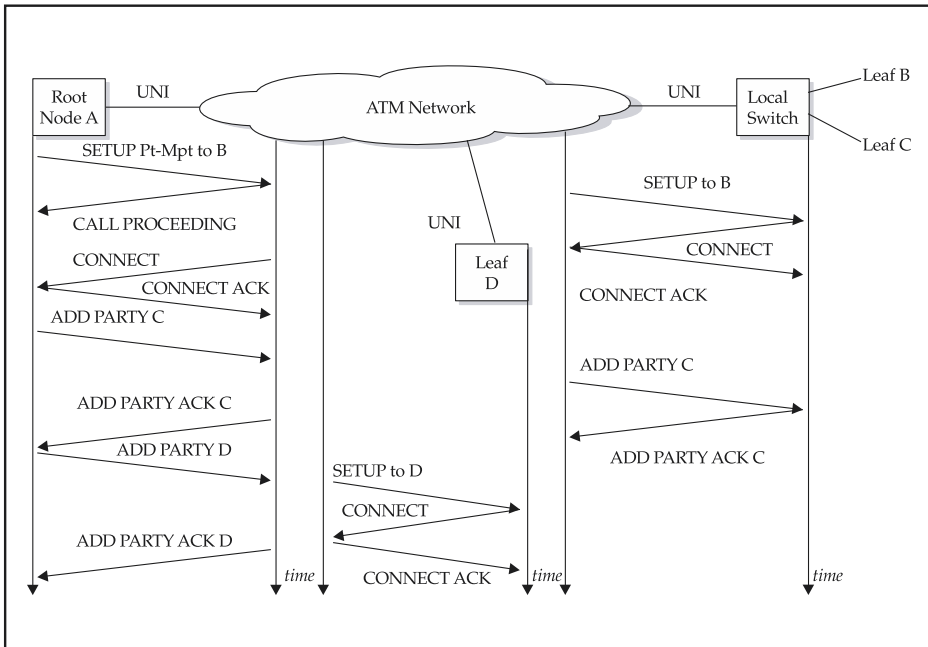


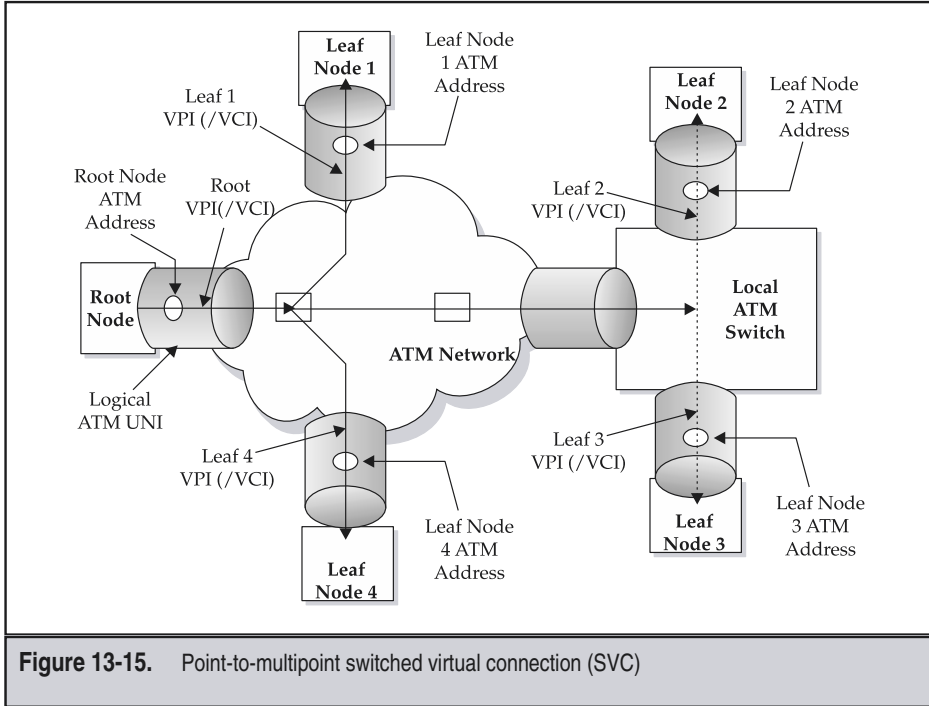
Figure 13-14. Root-initiated point-to-multipoint connection establishment example

destination and issues a SETUP message to node B identifying the assigned VPI/VCI. Leaf node B then indicates its intention to join the call by returning a CONNECT message that the network in turn acknowledges with a CONNECT ACKNOWLEDGE message. The network informs the calling root node A of a successful addition of party B through a CONNECT and CONNECT ACKNOWLEDGE handshake as shown in the figure.

Continuing with the same example, the root node requests addition of party C through the ADD PARTY message. The network relays the request to the same ATM UNI used by party B through the ADD PARTY message to inform the local switch of the requested addition. In other words, the network uses the SETUP message only to add the first party on any particular ATM UNI and uses the ADD PARTY message for any subsequent leaf added to an ATM UNI that already has a connected party in the point-to-multipoint call. Party C responds with an ADD PARTY ACKNOWLEDGE C message that the network propagates back to the root node A. Finally, the root node A requests addition of leaf party D through an ADD PARTY message. The network routes this to the UNI connected to party D and issues a SETUP message, since this is the first party connected on this particular ATM UNI. Node D responds with a CONNECT message, to which the network responds with a CONNECT ACKNOWLEDGE message. The network communicates the successful addition of leaf party D to the call to the root node A through the ADD PARTY ACKNOWLEDGE D message.

The leaves of the point-to-multipoint call may disconnect from the call using the DROP PARTY message if one or more parties would remain on the call on the same UNI, or using the RELEASE message if the party is the last leaf present on the same UNI. For example, the local switch could disconnect party C by sending a DROP PARTY message, but it must send a RELEASE message if party B later disconnected. If the root node initiates disconnection, then it drops each leaf in turn and finally releases the entire connection. Note that the root node is aware of all the parties in the call, since it added each one to the point-to-multipoint connection.

A point-to-multipoint SVC (or SPVC) has one root node and one or more leaf nodes. The VPI (and VCI for VCCs) along with the ATM address associated with the signaling channel of the root node, and the ATM address and VPI and VCI for the signaling channel for each leaf node of the connection define a point-to-multipoint connection, as shown in Figure 13-15. There is essentially only a forward direction in a point-to-multipoint connection, because the network allocates zero bandwidth in the backward direction. However, the network must provide a backward flow for OAM cells and for use by other protocols. Note that more than one VPI/VCI value and ATM address on a single physical interface may be part of a point-to-multipoint connection. This means that the number of physical ATM UNI ports is always less than or equal to the number of logical leaf endpoints of the point-to-multipoint connection. The implementation of a point-to-multipoint connection should efficiently replicate cells at intermediate switching points within the network as illustrated in the figure. Replication may occur within a public network, or within a local switch. A minimum spanning tree (see Chapter 9) is an efficient method of constructing a point-to-multipoint connection. Both the LAN Emulation and IP Multicast over ATM protocols make extensive use of switched point-to-multipoint ATM connections when emulating broadcast LAN protocols.



ATM CONTROL PLANE ADDRESSING

The control plane routing and signaling functions rely on addressing to identify interfaces, switches, and other functions. We first summarize the standardized ATM address formats and encodings. This section concludes with an overview of the ILMI address registration procedure and the ATM Name Service.

Control Plane Addressing Levels

Two capabilities are critical to a switched network: addressing and routing. *Addressing* occurs at the link level between ATM devices at the VP and VC identifier levels as previously described, but more importantly at a logical network, or end-to-end level. Addressing also occurs in the association of signaling channels with bearer channels. Since the VPI/VCI is unique only to a physical interface, the higher-level address must be unique across all interconnected networks. Ideally, the address should be unique across all networks in order to provide universal connectivity if the networks later interconnect. Once each entity involved in switching virtual connections has a unique address, there is

an even more onerous problem of finding a route from the calling party to the called party. *Routing* solves this problem by one of two means: either static, manual configuration, or dynamic, automatic discovery.

ATM Level Addressing

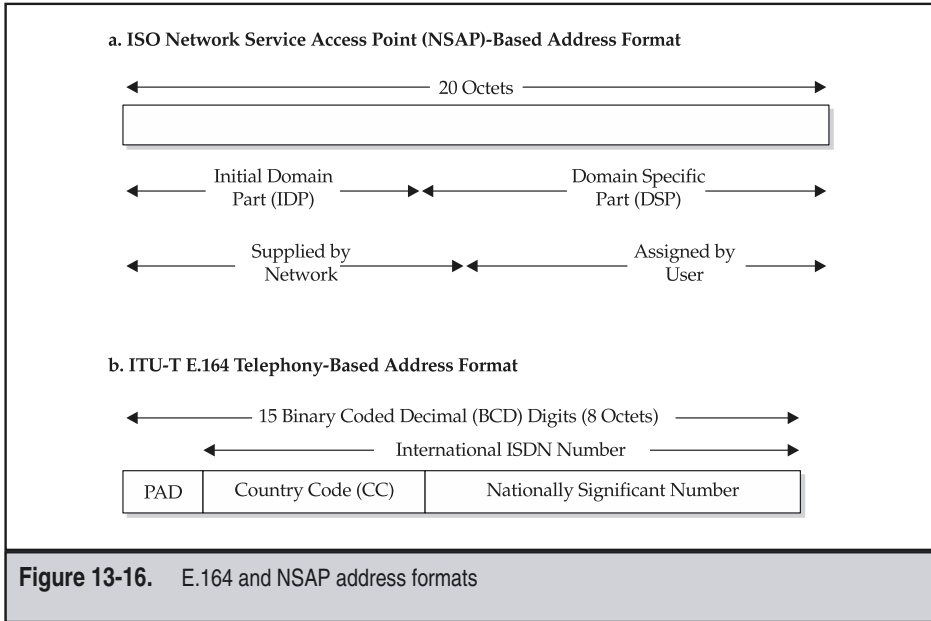
The signaling protocol automatically assigns the VPI/VCI values to SVC calls between ATM addresses corresponding to ATM UNI signaling channels according to a set of rules. The VPI/VCI values in the case of associated signaling, or the VPCI/VCI values in the case of nonassociated signaling, are unique to the signaling channel. Each ATM UNI signaling channel must have at least one unique ATM address in order to support SVCs, but it may also have more than one ATM address. One or more ATM addresses can uniquely identify an interface, and the attached user can receive an ATM SVC call on any of these addresses. An address prefix associated with an interface can either identify a set of users or identify a route toward a destination. The same address or prefix associated with multiple interfaces usually identifies a set of route choices for reaching that address or prefix.

ATM Addressing Formats

Two types of ATM Control Plane (SVC) addressing plans identify an ATM UNI: a data-oriented Network Service Access Point (NSAP)-based format defined by the International Organization for Standardization and the telephony-oriented ITU-T E.164 standard. An important contribution of the ATM Forum UNI specifications toward the goal of global ATM internetworking was adoption of an address structure based upon the ISO NSAP syntax. On the other hand, the ITU-T initially adopted the use of telephone number-like E.164 addresses as the addressing structure for public ATM (B-ISDN) networks to interwork with legacy telephone and Narrowband ISDN networks. This fundamental lack of agreement on the address family was one factor that thwarted the deployment of ATM SVCs.

Figure 13-16a illustrates the ISO NSAP-based ATM End System Address (AESA) format. International (e.g., British Standards Institute) and national (e.g., ANSI) standards bodies assign the Initial Domain Part (IDP) to various organizations, such as carriers, companies, and governments, for a nominal fee. The remainder of the 20-octet address is called the Domain Specific Part (DSP). The next section details the AESA formats. The network provider supplies the IDP part obtained from an administrative body as well as part of the DSP. The domain's network administrator defines the remaining octets. The end user part contains at least 7 octets. The NSAP standards define a more rigid structure than adopted by the ATM Forum, which is why we say that the Forum's address structure is *NSAP based* and not NSAP formatted. The reason the ATM Forum chose a more flexible format was to achieve better scalability through hierarchical assignment of the IDP part of the address in PNNI.

Figure 13-16b illustrates the ITU-T-specified E.164 address format. This is the same format used for international telephone numbers, which begins with a country code (e.g., 01 for North America, 44 for the UK, etc.), followed by a number defined within that



country. This plan served voice telecommunications well for over 50 years, but it assumed only one carrier per country. Telecommunications deregulation created multiple carriers within a country, violating the underlying paradigm of the E.164 numbering plan. Furthermore, with the proliferation of fax machines, cellular phones, and multiple phones per residence, the E.164 numbering plan had too few digits for metropolitan areas, necessitating renumbering of area codes and even individual numbers. Unfortunately, this need to change addresses to support continued growth in the telephony sector occurs on an increasingly frequent basis in response to growing demand. ITU-T standards work that evolves the E.164 plan to assign a country code to specific carriers is an attempt to address the emerging global competitive nature of networking.

The international E.164 number contains up to 15 binary coded decimal (BCD) digits padded with zeros on the left-hand side to result in a constant length of 15 digits. The ITU-T assigns a Country Code (CC) of from one to three digits as standardized in Recommendation E.163. The remainder of the address is a Nationally Significant Number (NSN). The NSN may be further broken down as a National Destination Code (NDC) and a Subscriber Number (SN). The North American Numbering Plan (NANP) is a subset of E.164. Currently, Neustar administers the NANP. The NDC currently corresponds to an

Numbering Plan Area (NPA) code and switch NXX identifier for voice applications in North America.

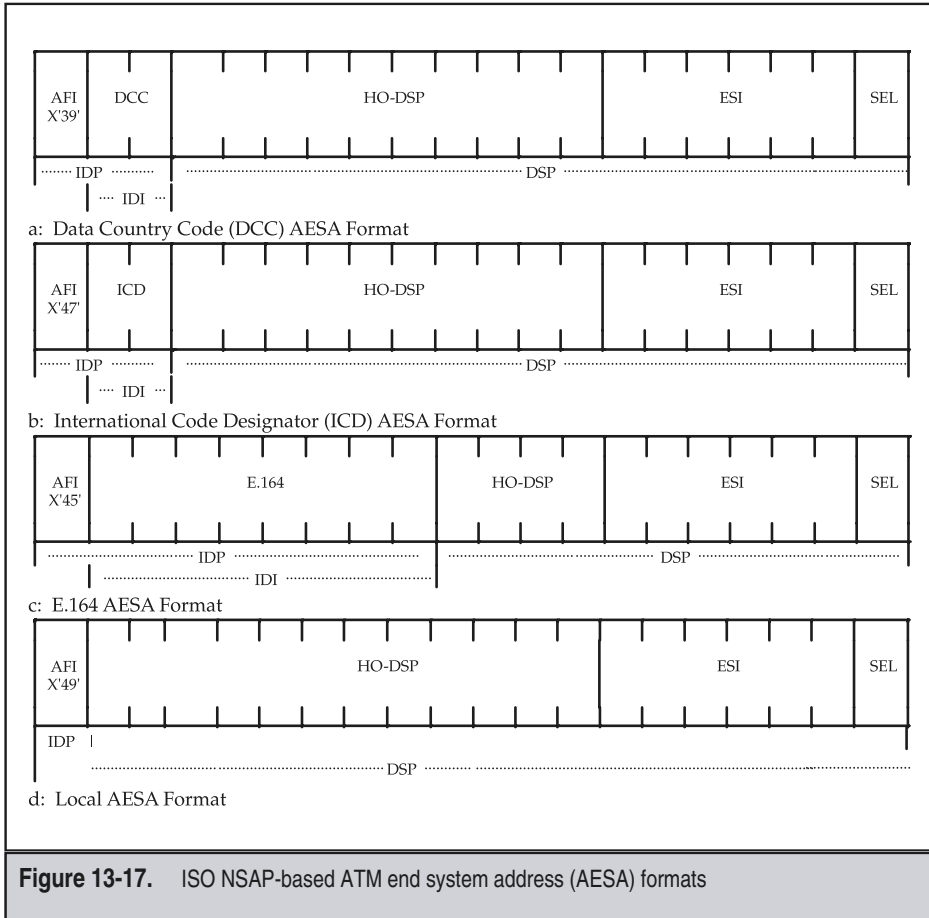
NSAP-based numbers can be hard to remember (unless your parents spoke binary code while you were growing up), so the ATM Forum's ATM Name Service [AF ANS 2.0] provides a means to look up an ATM address according to a "name," which may be a human-readable name, an IP address, or another ATM address. The ANS specification represents an ATM End System Address (AESA) as a string of hexadecimal digits with the "." character separating any pairs of digits for readability.

ATM Forum ATM End System Address (AESA) Formats

An ATM End System Address (AESA) is derived from the International Organization for Standardization (ISO) standard defined in ISO/IEC 8348, and it also is available as ITU-T Recommendation X.213; these specify the format, semantics, syntax, and coding of AESAs. The ATM Forum UNI specifications define four AESA formats as shown in Figure 13-17: DCC, ICD, E.164, and local. Many ATM Forum specifications discuss the use of AESAs. Fortunately, the ATM Forum has now consolidated all addressing-related material into two documents that supersede all other ATM addressing material: the ATM Forum addressing reference guide [AF ADDR GUIDE] and the addressing user guide [AF ADDR USER]. For ATM addressing issues, these are the two documents to consult. We now summarize each format.

As seen from Figure 13-17, each AESA address format has an Initial Domain Part (IDP) followed by a Domain Specific Part (DSP). The IDP has two parts: the Authority and Format Identifier (AFI) and the Initial Domain Identifier (IDI). The combination of the AFI and the IDI uniquely specify an Administrative Authority (AA) that has responsibility for allocating and assigning values of the DSP. The length of IDP field varies depending upon the particular AESA format. The one-byte AFI field identifies the format for the remainder of the address. The DSP has a High-Order DSP (HO-DSP) and low-order part composed of an End System Identifier (ESI) and a Selector (SEL) byte. The length of the DSP varies but is always 20 bytes minus the size of IDP.

The true NSAP format subdivides the DSP into a fixed hierarchy that consists of a Routing Domain (RD), an Area identifier (AREA), and an End System Identifier (ESI). ATM Forum UNI 3.1 combined the RD and AREA fields into a single High-Order DSP (HO-DSP) field in order to achieve a flexible, multilevel hierarchy prefix-based routing protocol in PNNI. The specific use of the Higher Order DSP (HO-DSP) is determined by the standards body identified in the IDP. The End System Identifier (ESI) and SElector (SEL) portions of the DSP are identical for all IDI formats as specified in ISO 10589. The ESI must be unique within a given IDP+HO-DSP address prefix. The ESI can also be globally unique, for example, a unique 48-bit IEEE MAC address. Beware that not all MAC addresses are unique, since some devices allow the user to assign the MAC address. The SElector (SEL) field is not used for routing, but End Systems (ES) may employ it for local multiplexing.



Group Addresses and Anycast

An ATM *group address* acts in the role of a service identifier rather than an address. An ATM group represents a collection of ATM end systems. An ATM group has one or more members, and an ATM end system can be a member of zero or more ATM groups at any time. An ATM end system may join or leave a group at any time by registering a group address using the ILMI client address registration and deregistration procedures described later.

The membership of an ATM end system in a group may have an address scope associated with it that determines how widely advertised or known that address is. Group addresses are distinguished by their AFI values, and there is a fixed relationship between the AFIs of individual addresses and AFIs of group addresses. A well-known group address is used to identify the group of devices that provide a well-known ATM service, for example, the LAN Emulation Configuration Server (LECS) service or the ATM Name System (ANS).

If a user places a point-to-point call to a group address, then the network routes the call to the port “closest” to the source associated with the specified group address. Since the network sets up the connections to any of several possible destinations, the ATM Forum specification calls this capability anycast signaling, which is the only ATM Forum–defined service using a group address. Several higher-layer protocols, such as LANE and MPOA, make use of this *anycast* and group addressing function to provide multiple servers within a network. The ILMI address registration protocol supports dynamic registration of group addresses so that networks dynamically discover new or relocated servers. This procedure also supports the means for the scope of a registering anycast server, which restricts the level of address advertisement and hence prevents use by end users of particular anycast servers that may be located too far away.

ILMI Address Registration

The ILMI protocol [AF ILMI 4.0] includes mechanisms for address registration at the UNI. Address registration allows the network to communicate to the user the valid address prefixes for a particular logical ATM UNI; the user then registers complete addresses by suffixing the address prefix with the ESI and SEL fields. Optionally, the user may register a connection scope along with each address. Thus, ILMI overcomes the need to manually configure large numbers of user addresses. It also enables source authentication, since the originating switch may screen the calling party address information element in the SETUP message against the set of registered addressed prefixes. ILMI address registration is a key component of automatic configuration of PNNI reachability information when using ATM SVC networking.

A switch uses the ILMI address registration protocol to provide one or more 13-octet NSAP-formatted prefixes to an end system, which in turn constructs an AESA by appending its ESI and SEL byte to result in a 20-octet address that is then returned to the switch. The end system may register multiple addresses using different network prefixes or different ESIs. The switch uses the addresses to identify the UNI for purposes of call delivery. For the purposes of address registration, the value of the Selector field is irrelevant. A switch may also use the ILMI address registration protocol to inform end systems of native E.164 numbers assigned to the public UNI by an ATM service provider. In this case, the switch supplies the whole ATM address to the end system. In effect, the network prefix is the native E.164 number and can only be validly combined with a null user part.

The Service Registry MIB information portion of the ILMI provides a general-purpose service registry for locating ATM network services, such as the LAN Emulation Configuration Server (LECS) and the ATM Name Server (ANS). Either of these servers could be assigned a group address and accessed via the anycast signaling procedure described earlier.

Bi-Level Addressing

Since customers and ATM service providers can independently acquire ATM address prefixes, there is a need to route SVCs to any address prefix, independent of the service provider to which the users with that address prefix are connected. There is a need, when supporting SVCs across two or more public ATM networks that route by provider address, to carry some other ATM address to complete the connection across another ATM network. For example, a customer may have a private ATM network that has sites connected to multiple ATM public networks that all support SVCs.

The ATM Forum defined bi-level addressing [AF BI ADDR] to meet these needs by performing processing of certain address prefixes in an external database, leaving the switch to support a smaller set of address prefixes for its own network and the ones that it attaches to. The basic idea is to use two ATM called party addresses in the SETUP message called an ATM Terminating Interface Address (ATI) and an ATM Destination Point Address (ADP). The ATI address is the address identifying the terminating public network interface, while the ADP is the customer-provided ATM address. The ingress public switch initiates bi-level addressing by placing the called party number received in the SETUP message in the ADP, using the result of the database translation that returns the terminating network public interface, and inserting this value in the ATI. The original called party address carried in the ADP across one or more provider networks is restored at the ATI terminating interface and again used in the called party number information element as call setup progresses.

The ATM Forum defines two specifications to support bi-level addressing. The ATM Name System (ANS) [AF ANS 2.0] is used for storing the binding between an ATI and one or more ADPs and supporting translation queries, as described in the next section. The transported address stack (TAS) [AF TAS 1.0] defines the encoding and procedures used to transport the ATI and ADI addresses in ATM signaling messages.

ATM Name Service (ANS)

The ATM Forum adopted the Domain Name System (DNS) concept from the Internet to resolve names into ATM addresses in the ATM Name Service (ANS). ANS supports both NSAP-based and E.164 ATM addresses. In the Internet, a DNS resolves a host name and organization in an e-mail address (i.e., user@host_name.org) or a Web site (e.g., http://www.usersite.org) to an IP address. Most human beings find it easier to remember a name than a number. There are exceptions among us, such as those capable of rattling off

IP addresses and other numeric data more readily than their own children's names; however, you won't likely encounter them at too many cocktail parties.

The ATM Name System is based on the Internet's Domain Name System protocol [RFC 1034] and derives its name structure similarly. The specification provides the details of the information kept for each ATM end system, how the end system contacts the ATM Name System server, and the form of the queries and replies. ATM names use the same names as in the Domain Name System, but new resource records are defined to contain the ATM addresses associated with those names. ANS supports both types of ATM addresses: AESAs and native E.164 numbers. The protocol also specifies the means for servers to communicate in the processing of providing service to ANS clients. The basic directory services defined in ANS are:

- ▼ Name to address translation to discover the location of services in a SVC environment
- ATM address to name translation to discover the domain name that corresponds to an ATM address
- ATM address to ATM address mapping to discover an ATM address given another number (native E.164 address given an AESA)
- ATM address to Internet Protocol (IP) address mapping to discover the IP address for dual hosts
- IP to ATM address mapping to discover the ATM address for dual hosts
- Discovery of the ATM Terminating Interface (ATI) addresses that serve a particular ATM Destination Point (ADP), supporting ATM bi-level addressing
- Secure dynamic registration and update of ATM addresses for a given host or service
- Secure dynamic registration and update of ATI addresses for a given ADP address
- Query responses that can be authenticated
- ▲ Distribution of public keys

REVIEW

This chapter defined B-ISDN and MPLS from the top-down perspective in terms of the user application and control planes. We provided an overview of how ATM and MPLS support a wide range of user plane protocols, such as voice, video, and LAN and WAN data protocols as an introduction to Part 4 and as motivation for the control plane protocols. The text then summarized how ATM and MPLS control plane protocols support the services provided by the user plane.

The chapter then introduced the control plane, which is central in performing the functions needed in an ATM Switched Virtual Connection (SVC) service. We defined the context for the various signaling protocols and articulated the structure of the control plane Service Specific Convergence Sublayer (SSCS) and its constituent components: the Service Specific Coordination Function (SSCF), and the Service Specific Connection-Oriented Protocol (SSCOP). Next, we delved into the important concept of the ATM UNI and ATM addressing, with an overview of group addresses and anycast and bi-level addressing; the Integrated Local Management Interface (ILMI) address registration procedure; and the ATM Name Service (ANS).

CHAPTER 14



MPLS Signaling and Routing Protocols

This chapter begins by describing the MPLS control plane architecture. We present the concepts of MPLS control plane function as a combination of routing and signaling protocols as a means necessary to achieve constraint-based routing as an introduction to the more complex ATM PNNI protocol described in the next chapter. We then describe basic concepts as well as the specific MPLS signaling protocols. The text then summarizes traffic-engineering extensions to IP routing protocols, which are essential to the implementation of constraint-based routing by MPLS signaling protocols, which are widely used in real-world networks. The chapter then concludes with some illustrative examples of the operation of MPLS control plane protocols.

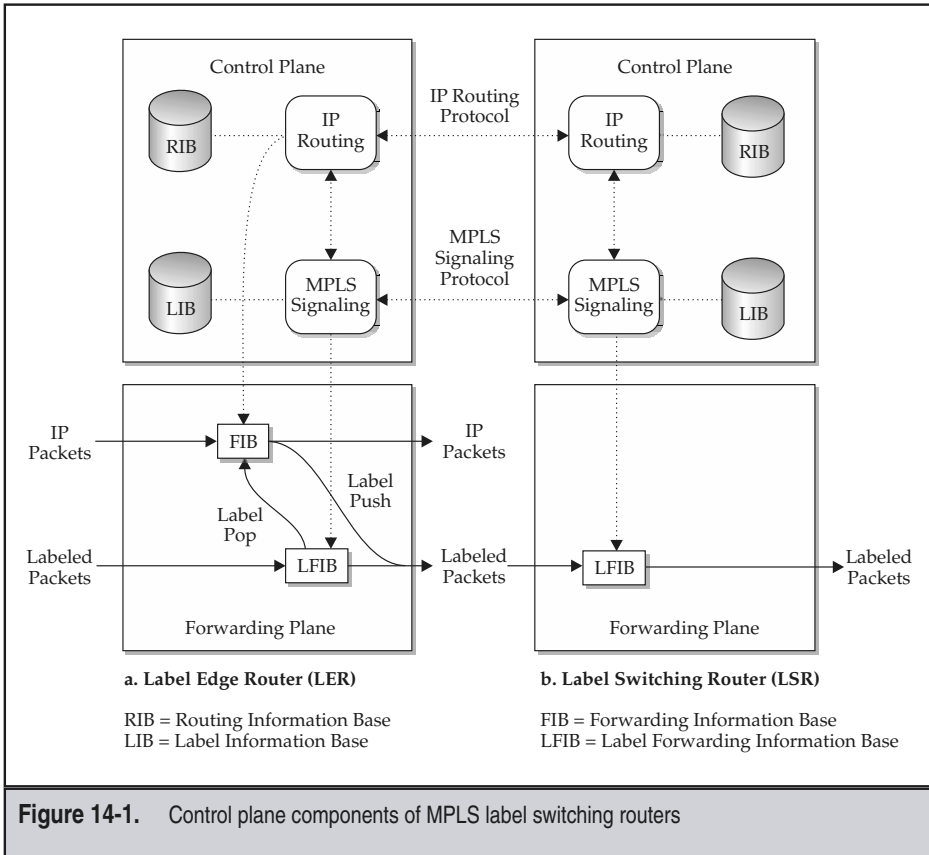
MPLS CONTROL PLANE ARCHITECTURE

This section begins with a high-level model of the MPLS forwarding and control planes. It then summarizes the motivation for adding a signaling protocol and augmenting IP routing protocols to support traffic engineering via constraint-based routing. The discussion then summarizes the various modes in which MPLS label distribution signaling protocols can operate and be used in MPLS-augmented IP networks.

MPLS Control and Forwarding Plane Model

MPLS can be viewed as having a forwarding plane and a control plane, as illustrated in Figure 14-1. In Chapter 11, we covered the forwarding plane, which is composed of two forwarding lookup tables in an MPLS-capable router; a hybrid IP and MPLS forwarding information base (FIB) and an MPLS-only label FIB (LFIB), as shown at the bottom of the figure. RFC 3031 defines a router supporting MPLS as a label switching router (LSR) and a router that first applies (or is the last to remove) a label for an IP packet as a label edge router (LER). As seen from the model in the figure, this fact is represented by virtue of an LSR having only an LFIB, while an LER has both an LFIB and a FIB. (Note that the LSR and LER functions are logical, and that any real router could act as an LER and as an LSR.) As described in Chapter 11, the FIB contains mappings from a forwarding equivalence class (FEC), for example an IP address prefix, to a MPLS label and a next-hop physical interface. On the other hand, the LFIB defines operations only on labeled packet inputs, but it may generate a labeled or unlabeled packet output.

It is the job of the control plane protocols to distribute information necessary for each LER and LSR to configure the FIB and LFIB. As shown in the upper part of Figure 14-1, this can be viewed as being composed of a routing protocol using a routing information base (RIB) operating in conjunction with an MPLS signaling protocol for distributing labels that uses a label information base (LIB). Normally, the establishment of an MPLS label switched path (LSP) occurs in response to a control- or topology-driven process. As described in Chapter 11, this separation of forwarding and control planes allows one to install an MPLS control protocol on an ATM switch, and thus increase the useful life of an ATM switch in a network. Although at the time of writing, the IETF work focused on IP-based signaling and routing protocols in the control plane, other protocols could be defined in the future.



Why does MPLS require a signaling protocol, when the classical IP router studied in Chapter 9 only needs IP routing? One important driver for the use of an MPLS signaling protocol in conjunction with a routing protocol arises from the need to perform constraint-based routing of an MPLS label switched path (LSP) as contrasted with that of shortest-path routing in a traditional IP network. We now look at some motivation for this fundamental architectural difference before describing more specific aspects of the MPLS control plane architecture.

Motivation for Constraint-Based Routing

As described in Chapter 9, in a connectionless protocol like IP, each router independently makes a decision about the next forwarding hop based solely upon the contents of a received

packet header (i.e., as mapped to an FEC) along with topology information learned about the rest of the network via a routing protocol and configured policy information. Typically, such a network computes the least-cost or shortest path based upon the metrics configured for its interior link state routing protocol. However, often a networking problem requires routing subject to multiple optimization criteria or, alternatively, routing subject to one or more constraints. As a result, the preferred route by traffic engineering may be different from the least-cost route. A constraint may take on a number of forms, including limits on QoS parameters, sufficient capacity, avoidance (or exclusive use of) particular link types such as satellites (or fiber optic) facilities, or other general policies [RFC 2702].

Figure 14-2a depicts a simple five-router network used to illustrate the differences between shortest-path and constraint-based routing when the constraint is capacity. The five routers labeled A through E shown in the figure are interconnected by DS3 links, each with a capacity of 45 Mbps. This example concerns the routing of 10 Mbps flows from a port on router A to a port on router B. For purposes of simplicity, we assume that no other flows are active and that the shortest path is the one with the least number of hops. Shortest path routing directs all of the flows over the shortest (i.e., least number of hops) path A-C-B, creating an overload on this route, potentially impacting performance. This situation can occur with shortest-path routing because the optimization considers only one parameter—the path length or cost—and does not consider other constraints, such as required capacity or quality.

Connection-oriented protocols handle capacity-constrained routing naturally. Typically, they assign connections to the best path until it is full and then assign connections

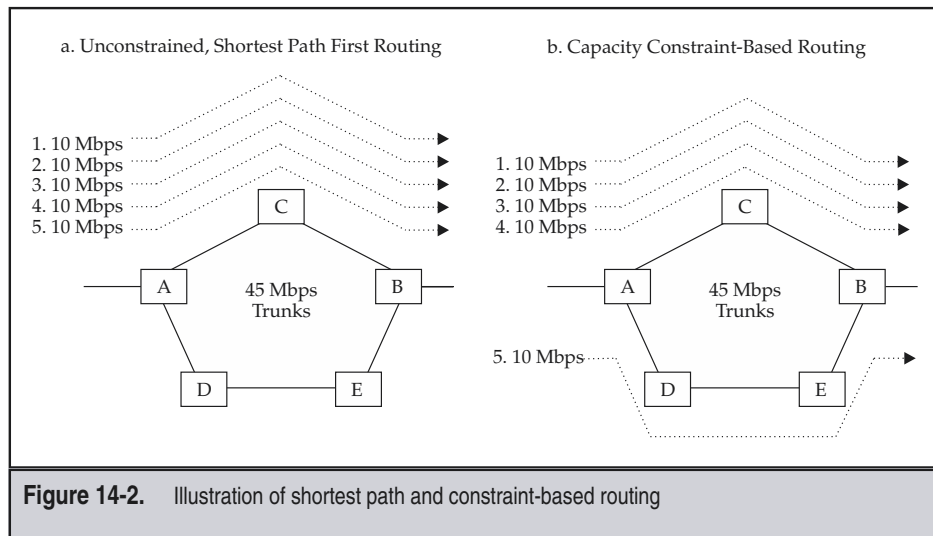


Figure 14-2. Illustration of shortest path and constraint-based routing

to the next-best path. If no path can support the connection request; then the network blocks the attempt. Another approach that achieves the same result is for the source to determine an explicit route that meets all constraints and then signal the desired configuration to nodes along the path. Figure 14-2b illustrates the same network with capacity-constrained routing. The network chooses the shortest path, A-C-B, for the first four 10 Mbps connection requests from A and B; however, the network cannot route the fifth connection request on the shortest path, because it is full. Therefore, constraint-based routing selects the next-best path with available capacity, namely A-D-E-B, as shown at the bottom of the figure.

At best, constraint-based routing makes the optimal routing decision according to the state of the network at the time of the request. However, the routing in such a network can become suboptimal when supporting connections with long holding times. Consider the example of Figure 14-2b again, where the five connections are active. After a period of time, connections 1 through 4 release, leaving connection 5 on a suboptimal path. If other connection requests arrive, for example, four connections between D and E each requiring 10 Mbps, the network may block one of these requests, or else choose the suboptimal route D-A-C-B-E. One way around this issue is to periodically optimize the network by rearranging long-duration connections in a relatively nondisruptive manner. However, rearrangement adds complexity to connection-oriented routing and processing and also creates the potential for interruptions in service unless some form of make-before-break grooming is implemented.

MPLS Label Distribution Control Protocol Attributes

This section summarizes the following attributes for controlling the distribution of MPLS labels, as described in RFC 3031 and the standardized MPLS control protocols described later:

- ▼ Hop-by-hop versus explicit routing of LSP tunnels
- Unsolicited downstream versus downstream on demand
- Conservative versus liberal label retention mode
- Ordered versus independent LSP control
- FEC aggregation and granularity
- ▲ Support for merging and nonmerging LSRs

Hop-by-Hop Versus Explicit Routing LSP Tunnels

Since the path taken by a packet is defined by the ingress router (i.e., LER) mapping of a FEC to a label and then determined by MPLS label switching thereafter, RFC 3031 calls this use of MPLS an *LSP tunnel*. This means that only the head end of the tunnel need know the FEC that is assigned to the LSP. The MPLS protocol architecture defines two options for distributing labels for an LSP tunnel. The first is the mode commonly used in IP networks, called hop-by-hop routing, where each LSR independently chooses the next hop for each FEC. A number of label distribution modes can be used with this option. The

second mode is called explicit (or source) routing, where each LSR does not independently choose the next hop, but instead a single LSR, typically the ingress router, specifies several (or all) of the nodal hops in an LSP. In general, an *abstract node* in an explicitly routed LSP can be a specific IP address, an address prefix, or an autonomous system (AS) number. If the LSP must traverse only a specific set of abstract nodes, then we say that the path is *strictly explicitly routed*. In this case, the source node may precisely set the TTL. Otherwise, if the path must contain an abstract node of the explicit route, but may contain others, then we say that the path is *loosely explicitly routed*. Note that the hops in loosely explicitly routed path may change during the duration of an LSP, or they can be pinned so that the path does not change.

Explicit routing can direct an LSP along a path that differs from the one determined by the normal hop-by-hop IP routing protocol. It can be used to distribute traffic over multiple paths to route around network failures or congested links or preallocate a backup LSP on a link or node diverse path to enable fast protection switching. The sequence of LSRs in an explicitly routed LSP may be determined automatically from information exchanged via a dynamic routing protocol or by manual configuration.

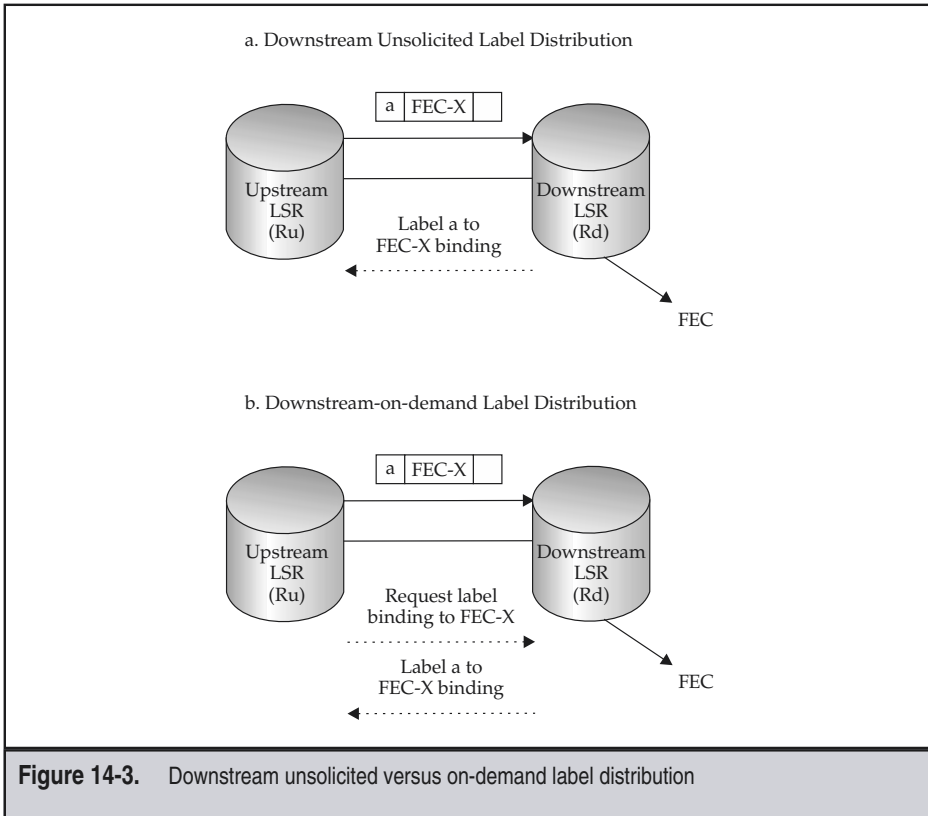
Unsolicited Downstream Versus Downstream on Demand

Figure 14-3 illustrates the terminology defined in RFC 3031 for label distribution protocols. The position of an LSR with reference to the flow of labeled packets for an MPLS LSP is referred to as either upstream or downstream, as shown in the figure. Before the upstream router can send labeled packets corresponding to a particular FEC reachable by the downstream LSR, the downstream router must first issue a message to bind a label to that FEC. This can occur in one of two ways. The first, shown in Figure 14-3a, allows a downstream LSR to distribute bindings to an upstream LSR, even if it did not explicitly request such a binding, in a technique called *downstream unsolicited* label distribution. If the downstream LSR is the normal next hop for IP routing for a particular FEC, then the upstream LSR can use this type of label binding for forwarding packets for packets in that FEC.

Figure 14-3b illustrates the second method, where the upstream LSR explicitly requests from its downstream LSR a label binding for a particular FEC, in a mode known as *downstream on-demand* label distribution. In this mode, the downstream router may not be the normal IP routing next hop for a particular FEC, a fact that is important for explicitly routed LSPs. Both label distribution methods can be supported in a network, but for each adjacency the upstream and downstream LSRs must negotiate and agree to use only one technique.

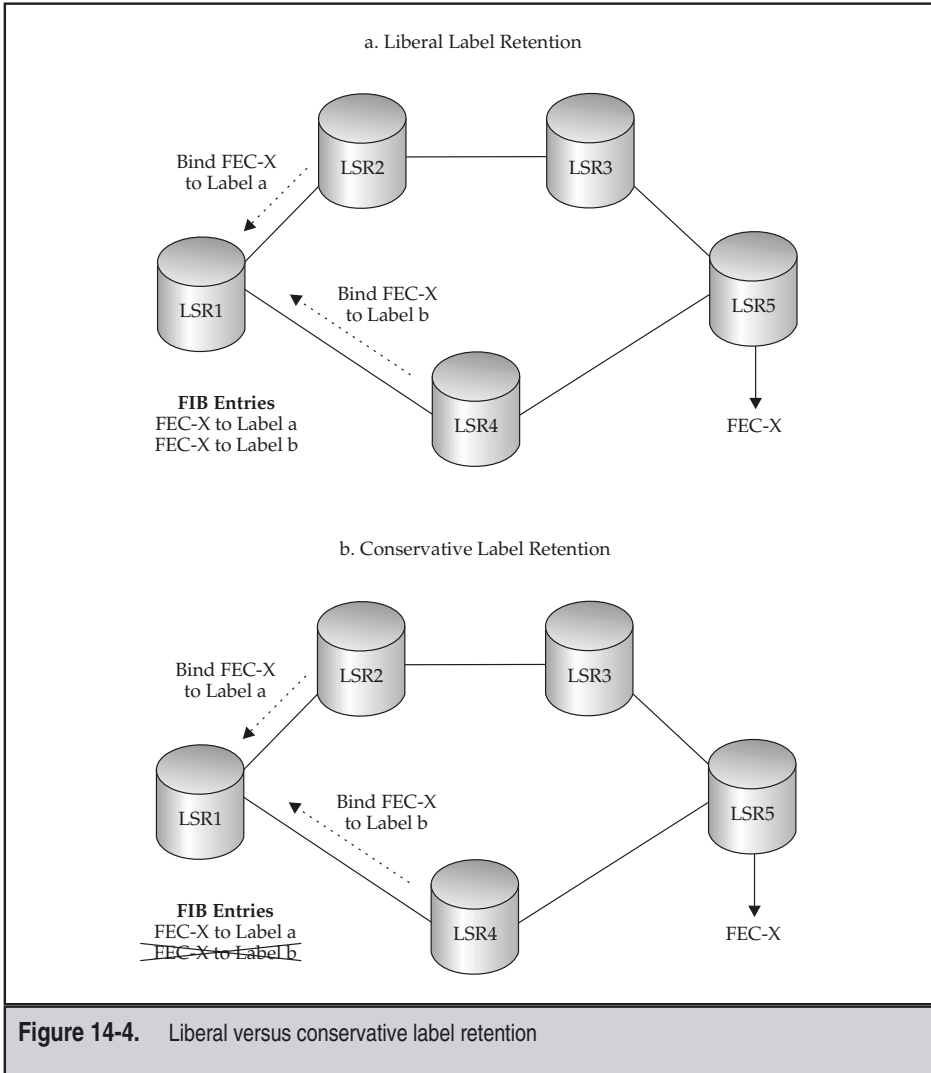
Conservative Versus Liberal Label Retention Mode

An upstream LSR may receive label bindings for a particular FEC from several downstream LSRs, for example, as shown in Figure 14-4. If an upstream LSR is using hop-by-hop forwarding and has a binding for an FEC to a label on the link to a downstream LSR that is not the current IP routing next hop, there are two ways to deal with the label bindings other than the next hop. Figure 14-4a shows the case in which an upstream LSR retains all label bindings, regardless of whether the downstream LSR is the current



IP routing next hop, in a mode called *liberal label retention*. The advantage of this mode is that the upstream LSR may immediately use the binding if the downstream LSR becomes the next hop for the particular FEC. For example, if hop-by-hop path routing is used, the normal path from LSR1 for FEC-X would be via LSR4. In the event that the link between LSR1 and LSR4 were to fail, LSR1 could immediately begin sending labeled packets to LSR2. However, note that this mode requires an LSR to maintain more label bindings and may create transient routing loops as demonstrated by a subsequent example.

Figure 14-4b illustrates the case in which the upstream LSR discards all bindings, except for the one that is the current next hop, in a mode called *conservative label retention*. This mode has the advantage that the LSR needs to retain fewer FEC-to-label bindings, but it increases the response time to changes in routing, since new label bindings must be requested or distributed when routing changes. This is an important mode for LSRs that support a limited number of labels, for example, an ATM switch.



Ordered Versus Independent LSP Control

A multipoint-to-point MPLS LSP has one or more associations of label bindings at LSRs along the path, as shown in Figure 14-5. When an FEC corresponds to an address prefix distributed by IP routing, the establishment of label binding associations at an LSR can be

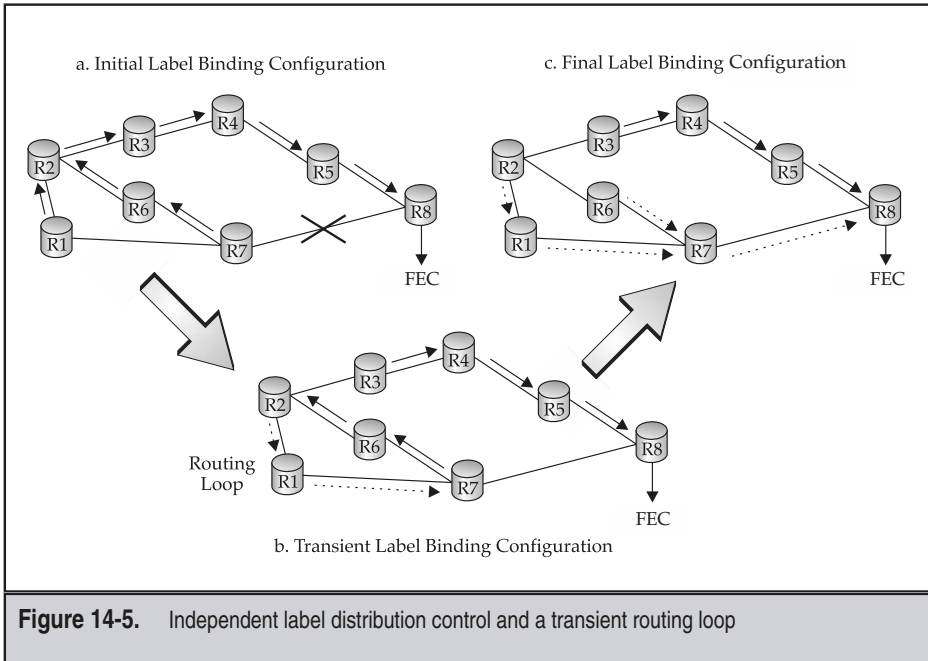


Figure 14-5. Independent label distribution control and a transient routing loop

done in one of two ways. The first case is *independent control*, where at the point when each LSR recognizes a particular FEC, it makes a decision to bind a label to that FEC and distribute that binding to its label distribution peers. This is similar to normal IP routing, where each router makes an independent decision about where to forward a packet. Independent control has the advantage of establishing an LSP rapidly, since label bindings can occur in parallel between many pairs of LSRs and traffic flow can begin before all label bindings are established.

However, independent control creates the potential for formation of routing loops, and it therefore requires use of a loop detection or mitigation mechanism. Figure 14-5a illustrates an initial label binding configuration for a network of eight routers with minimum hop routing for an FEC associated with egress router R8 under the condition when the link between R7 and R8 is in a failed condition. When the link between R7 and R8 becomes available, the routers redistribute labels in a transient phase, for example, that shown in Figure 14-5b. In this example, R1 is the first to issue the new label bindings (shown by dashed lines), and a routing loop forms between R1-R7-R6-R2-R1. Once all routers recognize the newly available R7-R8 link, the label binding configuration eventually reaches a loop-free, minimum-hop configuration, as shown in Figure 14-5c.

The second case is that of *ordered control*, where a downstream LSR performs label binding only if it is the egress LSR for that FEC or if it has already received a label binding

for the router downstream from it for that FEC. Ordered LSP establishment begins at the egress LSR and proceeds sequentially in a backward direction along the path to the ingress LSR. Explicitly routed LSPs must use ordered control, and the sequential distribution of label bindings causes a delay before traffic flow over the LSP begins. On the other hand, ordered control provides a means for loop avoidance and achieving a consistent level of aggregation.

Figure 14-6 shows a similar example for ordered control and how it avoids formation of a routing loop at the expense of increased response time for a restoration action. Figure 14-6a shows the same initial label binding configuration in the preceding example. Figure 14-6b shows a phase of orderly release of label bindings upon detection that the R7-R8 link is now active. During this period, these routers may not forward packets until a label binding to the FEC is reestablished. Figure 14-6c shows the phase of sequential label bindings proceeding backward from the egress router R8. Observe that a routing loop does not form, but that the ordered processing of release and reestablishment of label bindings will take longer than the preceding independent control example.

Note that in both of the preceding examples the same final label binding topology results. In an MPLS network, ordered and independent control are fully interoperable; however, unless all LSRs along the path of an LSP use ordered control, an LSR along the path could try to use the LSP before it is established, potentially causing packet loss or routing loops.

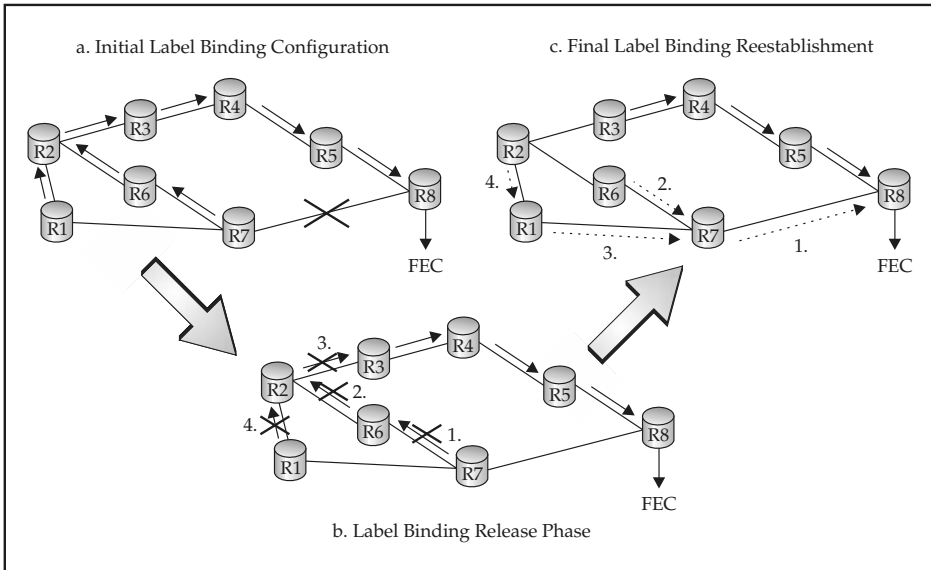


Figure 14-6. Ordered label distribution example

FEC Aggregation and Granularity

There is a choice in the way that FECs can be partitioned and assigned to LSPs. At one extreme, a separate LSP could be established for each FEC address prefix in the routing table. In this case, observe that all traffic for the set of FECs leaving a particular egress router would follow the same hop-by-hop path through the network. This suggests that the set of FECs corresponding to address prefixes served by a particular egress router could be combined into another larger aggregate FEC. In this case, an MPLS control protocol can set up a single LSP for all of these FECs, instead of one LSP for each FEC associated with the egress router.

The fact that MPLS supports different levels of FEC aggregation granularity impacts the label distribution control mode. In ordered control, an LSR can adopt the granularity used by its next hop, which recursively results in the ingress router using the granularity employed by the egress router for those FECs. On the other hand, in independent control potentially different levels of granularity require proper coordination. The simplest solution for independent control would be to configure the granularity for each FEC consistently across all LSRs.

Merging Versus Nonmerging LSRs

As defined in RFC 3031, an LSR that is capable of label merging can receive packets from different incoming interfaces with different labels, and send these packets out the same outgoing interface with the same label. The result is a multipoint-to-point LSP. Note that once the labels are merged, information about their source interface or label is lost. As discussed in Chapters 10 and 11, the forwarding plane of an LSR may or may not be capable of merging labeled packets when implementing a multipoint-to-point tree rooted at the egress LSR. In particular, this is often an important issue with ATM switches that cannot perform a VC label merge. Whether an LSR supports label merging has implications on the control protocol. In particular, the control protocol must be capable of performing different label distribution procedures for a merging LSR than those supported for a nonmerging LSR.

MPLS LABEL DISTRIBUTION SIGNALING PROTOCOLS

This section summarizes important aspects of protocols specified for signaling the distribution of MPLS labels using the generic terminology previously defined, as summarized in Table 14-1. The MPLS architecture of RFC 3031 does not mandate a single label distribution protocol, and therefore there are a number of choices, each with advantages and disadvantages discussed in the following text. As defined in many IETF-defined protocols, each protocol message is composed of a number of objects, each with a Type-Length-Value (TLV) encoding structure.

Label Distribution Protocol (LDP)

Cisco's tag switching protocol summarized in Chapter 10 is the ancestor of the label distribution protocol (LDP), as specified in RFC 3036. Unlike other approaches for label distribution

Label Distribution Signaling Protocol				
Mode	LDP	RSVP-TE	CR-LDP	BGP
Hop-by-hop routing	Yes	Yes	No	Yes (1)
Explicit routing	No	Yes	Yes	No
Unsolicited distribution	Yes	No	No	Yes
On demand distribution	Yes	Yes	Yes	No
Independent control	Yes	No	No	Yes
Ordered control	Yes	Yes	Yes	No
Point-to-point LSPs	Yes	Yes	Yes	Yes
Multipoint-to-point LSPs	Yes	Yes	No	No

Note: (1) Each hop is an Autonomous System.

Table 14-1. Support for MPLS Label Distribution Modes by Various Protocols

that piggyback on an existing protocol, LDP is entirely new and designed for that purpose alone. RFC 3037 describes the applicability of LDP as being useful where efficient hop-by-hop routing is important, for example in VPN label distribution or as a means to tunnel between BGP border routers. As described later, LDP has been extended to support constraint-based routing (CR-LDP).

The LDP Protocol

LDP peers are LSRs that use the following LDP message types to exchange information related to label distribution:

- ▼ **Discovery** To exchange periodic Hello messages for announcement and verification of a directly or nondirectly connected LSR
- **Session** To establish, negotiate parameters for, initialize, maintain, and terminate LDP peering sessions
- **Advertisement** To create, change, or delete FEC-to-label mappings
- ▲ **Notification** To communicate advisory and error information

LDP discovery messages are exchanged over UDP, while all other message types require reliable delivery and therefore use TCP, which has an option to use Message Digest 5 (MD5) authentication for heightened security. Label space allocation is necessary because different platforms or interfaces may support different quantities or types of labels. Label space is allocated on a per-platform basis or per-logical/physical interface basis, as

identified by a six-octet *LDP identifier* composed of a four-octet *LSR identifier* (e.g., a router loopback address) along with an identifier for the label space allocated. LDP sessions may be established with a directly or nondirectly connected peer, with a session for each label space allocation between the peers handled by a separate session. The per-interface mode is applicable only when LSRs are directly connected. LDP sessions between nondirectly connected peers are useful for VPN and pseudo-wire applications, as described in Part 4.

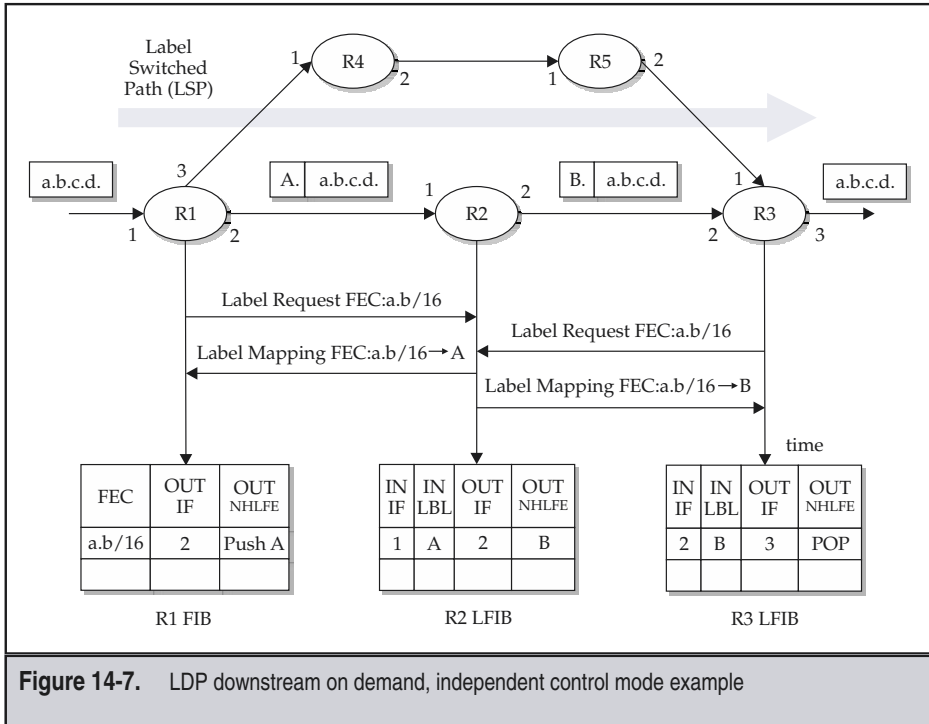
LDP FEC-to-label advertisement primarily uses two messages: label mapping and label request. Once LSRs have discovered each other and established a session, peers exchange *label mapping* messages that contain an FEC TLV and a label TLV, along with some optional parameters. The *label request* message contains an FEC TLV and optional parameters and is defined in support of downstream on demand label distribution. An *FEC TLV* is a set of FECs, each of which is either an address prefix or a full host address. A *label TLV* is an MPLS shim header, FR DLCI, or ATM VPI/VCI used to encode the label at the link layer, as described in Chapter 11. In the case of ATM, RFC 3038 defines additional LDP messages and procedures to distribute a per-LSP VCID in order to correlate the VPI/VCI mappings of each ATM VC link to the LSP. In order to support loop detection, label request and label mapping messages may have either a path vector or hop count TLV optional parameter. Path vector is a list of LSRs that the message has traversed, while the hop count is the number of LSRs traversed. If an LSR sees itself on the path vector or if the hop count exceeds a threshold, then the LSR must stop using the FEC-to-label mapping and signal an error notification that a loop was detected.

LDP also defines messages for specific situations. A *label withdraw* message allows an LSR to request that a peer stop using a specific label binding. The *label release* message indicates that a previously received or requested label is no longer needed. Also, a *label abort request* terminates a pending label request message.

An LSR stores the current state of bindings received from its LDP peers in a label information base (LIB), as shown in Figure 14-1. The LSR must then determine which bindings to use to configure the label forwarding information base (LFIB). The basic rule for matching a packet's destination IP address to an FEC to determine the label binding is that of the longest match. When several equally good matches occur, a typical implementation is to load-balance the mappings across those choices if they have equal routing metrics.

Example of LDP Downstream on Demand Independent Control

Let's look at an example of LDP in operation with reference to Figure 14-7 to illustrate the use of the label request and label mapping messages in downstream on demand mode with independent control. The figure uses the conventions introduced in Chapter 11 for identifying LSRs, their interfaces, and FIB and LFIB table entries. The middle of the figure shows the time sequence of LDP message exchanges between peers that establish an LSP from ingress router R1 through R2 to egress router R3 for an FEC specified as the IP address prefix a.b/16. Note that these messages are exchanged over the links between the routers. In this example, R1 initiates the process by requesting a label from its next hop R2 for FEC a.b/16. Since independent control is used, R2 can return a label mapping to R1 before it receives the label mapping from downstream R3. R2 and R3 then respond with



label mapping messages as shown in the figure, which result in the R1 FIB along with R2 and R3 LFIB entries shown at the bottom of the figure that form the desired LSP indicated by the bold arrow.

LDP supports other label distribution modes. When configured for downstream unsolicited mode, routers do not use the label request message. If ordered control were configured on each interface in the preceding example, the label request messages would cause the label mapping messages to be returned in sequence from R3 to R2 and then R2 to R1. In general, in downstream on demand with ordered control, the egress label mapping occurs first. Then each label mapping back occurs back toward the ingress only after completion of the downstream mapping, which is naturally done by RSVP as described in the next section.

RSVP Traffic Engineering (RSVP-TE)

As described in Chapter 8, the IP header had a field for type of service, but most implementations supported only a best-effort service. The second attempt by the IETF to define QoS was the Intserv approach, which used the Resource Reservation Protocol (RSVP) as

described in RFC 2205 (see Chapter 8). Although not widely adopted for this purpose, RSVP already had many of the mechanisms necessary to perform label distribution signaling subject to routing constraints, and Cisco originally defined it as a method to implement tag distribution. After several years of work, the IETF standardized RSVP traffic engineering (RSVP-TE) extensions in RFC 3209 to complete the job. RFC 3210 defines the applicability of RSVP-TE as supporting downstream on demand label distribution with support for resource allocation for explicitly routed LSPs. It also summarizes other uses of RSVP-TE to support make-before-break rerouting, tracking the actual path taken via a route record function, as well as support for priority and preemption.

RSVP messages are either sent as IP protocol 46 or encapsulated in a UDP datagram. Since part of the Intserv vision was that every router along the path could potentially process RSVP messages, the use of RSVP for MPLS label distribution by a service provider requires some additional security measures. One is use of MD5 authentication, and another is configuration of filters on non-MPLS interfaces that block RSVP messages.

Although RSVP runs over multicast, MPLS signaling primarily uses the much simpler unicast mode. The principal function of RSVP is to establish reservations for a unidirectional flow of packets. RSVP messages normally follow the hop-by-hop path of normal IP routing, unless the explicit route option is present. RSVP-aware routers along the path may intercept any message and process it. RFC 2205 defines three types of RSVP messages: reservation setup, tear down, and error, which have additional objects and uses as described in RFC 3209. RSVP-TE also defines an additional Hello message.

RSVP Reservation Setup Messages

RSVP uses the concept of receiver-based reservation, which involves the sender first issuing a *Path* message that identifies the flow and its traffic characteristics. The Path message contains a session ID, a sender template, a label request, a sender Tspec, and optionally an explicit route object. The *session ID* contains the destination IP address paired with a 16-bit tunnel ID that uniquely identifies an LSP tunnel. As described earlier, only the ingress LSR need have knowledge of the FEC assigned to an LSP tunnel, and therefore, unlike LDP, the FEC mapped to the LSP tunnel is not included in any RSVP message. The *label request* object supports downstream on demand mode, and it also includes a demultiplexing field so that an LSP for more than one protocol can be established. The *sender template* contains the sender's IP address paired with an LSP ID, which supports a method of make-before-break grooming when changing the path of an LSP tunnel. MPLS uses either the controlled load service defined in RFC 2211 or a null reservation. In controlled load service, the Traffic spec (called a *Tspec*) uses a peak rate, a token bucket to define the rate and burst size, a minimum policed unit, and a maximum packet size. See Chapter 20 for more information on the use of these traffic parameters. We describe explicit routing later.

Once the path message reaches its destination, the receiver responds with a *Resv* message if it wishes to initiate the label binding requested in the path message. The Resv message traverses the same hop-by-hop sequence of nodes as the path message in the reverse direction using previous hop information communicated in the Path message. It also contains

the session ID from the corresponding path message, an optional route record object, and reservation style-dependent information. The *fixed filter (FF)* style has a label and Tspec assigned to each sender-receiver pair. The *shared explicit (SE)* style assigns a different label to each sender, but they all explicitly share the same flow spec reservation. The *route record* object captures the actual route taken by the LSP starting from the egress back toward the ingress. This can be used by a router to pin a loose explicit route to a particular path by copying the recorded route received in a Resv in a path message into the explicit route object in a Path message sent in the opposite direction.

RVSP-TE Tear Down, Error, and Hello Messages

RSVP-TE defines two messages for tearing down an LSP: Path tear and Resv tear. Both tear messages are sent in the opposite direction from the corresponding Path or Resv message. The tear message removes any installed state associated with its paired message. The tear messages can be used to remove state in response to a failure as the first step in a rerouting action.

There are error notification messages for Path and Resv messages, as well as an optional Resv confirmation message. The error messages communicate a policy violation, a message coding error, or some other problem. For example, if an LSR finds that it cannot support the Tspec specified in an Resv message, it does not forward the Resv message upstream but instead generates a ResvErr message downstream to clear the LSP establishment attempt. The explicit route and record route options of RSVP-TE have a number of unique error codes to assist in debugging problems.

RFC 3209 defines an optional RSVP-TE Hello message, which allows an LSR to more rapidly detect that a neighbor has failed than would occur in a normal RSVP failure to refresh condition or detection of a link failure by an IP routing protocol. This can be quite useful in a fast rerouting application.

Downstream on Demand Ordered Control Explicit Routing

Figure 14-8 shows an example of the RSVP-TE message exchange used to install an LSP across a path other than the shortest hop path using the explicit route object (ERO). Router R1 determines that it will assign FEC a.b/16 to an LSP tunnel, and it computes an explicit route R4-R5-R3 to reach the next hop of that FEC. R1 initiates establishment of this LSP tunnel by issuing a Path message to R4 with an explicit route, Tspec, sender template that contains the sender's address) and a label request object. Each RESV message associated with this LSP tunnel contains the session ID and filter spec of the original sender R1 so that the messages can be correlated together. Next, R4 accepts this request and sends the path message to the next router in the ERO, R5, which in turn sends the message on to egress router R3.

Now at the destination of the Path message, R3 determines that the R3-R5 link can support the request and that it is the final hop on the path to FEC a.b/16. R3 responds with a Resv message containing the ERO, the Tspec of the reserved capacity, a filterspec matching the sender, and assignment of the implicit null label to this link. As defined in RFC 3031, the implicit null label is a convention used in label distribution that allows the

Of course, the processing for initial Path and Resv messages is much greater than that for refreshing the state of a previously received message; however, for a large number of LSPs, refresh processing has a significant performance impact. Furthermore, since refresh handles RSVP message loss, there can be a significant lag in recovering state in the event of a lost message.

One could address the scaling problem by increasing the refresh interval, but then signaling latency in the event of lost messages would increase. RFC 2961 specifies a solution to these processor scaling and signaling lag problems. The mechanisms specified include a message bundling to reduce processor load, as well as a means for a router to more readily identify an unchanged message. Also, message acknowledgment is added, which makes RSVP message transport reliable and handles the case of lost Path or Resv tear messages that are not refreshed in normal RSVP operation. Finally, the solution specifies a summary message that refreshes state without requiring transmission of the entire refresh message. These changes have addressed the issues of RSVP refresh overhead in deployed real-world MPLS networks.

RSVP-TE Priority, Preemption, and Resource Affinity

RFC 3209 also defines several additional objects that are quite useful in managing resources and implementing routing constraints in an MPLS network. The optional *session attribute* object defines a holding priority that determines whether another session with a higher setup priority can preempt its reservation. It also allows a sender to request fast service restoration from transit routers. At the time of writing, the IETF MPLS working group was finalizing standards to implement fast restoration in a standard way.

The session attribute may also address a general resource affinity constraint, which is called link coloring in RFC 2702. A routing protocol may distribute, or a link may be configured, with a 32-bit link attribute vector, which can be operated on by an LSR using three masks in the session attribute object: exclude any, include any, or include all. If a link vector matches any bit in the *exclude any* mask, then that link is excluded. If a link vector matches any bit in the *include any* mask, it is included. If a link vector matches all bits in the *include all* mask, then it is included. All three tests must pass for a link to be included. Resource affinities can be used to implement routing constraints such as avoiding satellite links, keeping LSP routing within certain geographic boundaries, or forcing transmission over high-speed links.

Constraint-Based Routing LDP (CR-LDP) Extensions

RFC 3212 defines constraint-based routing extensions to the label distribution protocol (LDP) described earlier by adding support for explicit routing, communication of LSP traffic parameters, resource classes (i.e., affinities), and priority/preemption. RFC 3213 describes the applicability of CR-LDP, which states that only the following LDP label distribution modes should be implemented in support of traffic-engineered LSPs: downstream on demand, conservative label retention, and ordered control. Note that RSVP-TE supports essentially the same list of features and label distribution modes. The reason

there are two IETF standards that support essentially the same function is primarily historical, in that some vendors and providers drove to extend RSVP, while others strove to extend LDP. After the RSVP refresh problem was addressed, there was no clear consensus regarding technical superiority of one protocol over the other, and in the usual IETF fashion, it was decided to standardize both protocols and let the marketplace decide. ITU-T Recommendation Y.1310 states that CR-LDP is the preferred label distribution protocol, since it is most like the ATM-based signaling protocols standardized by the ITU-T and therefore presents the greatest prospects for interoperability with native ATM protocols. At publication time, there were more implementations of RSVP-TE than there were of CR-LDP. Since many of the CR-LDP functions are similar to the generic MPLS architecture and RSVP-TE, this section provides only a brief summary of unique aspects of CR-LDP.

One of the most significant claimed benefits of CR-LDP is that it uses the reliable TCP transport protocol and is hard state, as compared with RSVP-TE, which is soft state and must use the more complex refresh overhead reduction protocol described earlier to be scalable. This is most important in LSRs that must support large number of LSPs. CR-LDP uses the same discovery and session messages as LDP. It adds a few objects to other existing messages. In particular, it adds TLVs for an explicit route, traffic parameters, route pinning, resource class, and preemption to the LDP label request message. CR-LDP also augments the label mapping message with a traffic parameter TLV. Although the encoding may differ, the function of most CR-LDP objects is similar to its RSVP-TE counterpart, with few exceptions. For example, the traffic parameter TLV is defined in terms of two token buckets, the first with a peak rate and burst size, along with a second at a committed rate and burst size, along with an excess burst size that may result in a higher drop precedence. Specific support for MPLS control of ATM switches also differentiates CR-LDP from RSVP-TE, with RFC 3215 detailing a state machine for this specific purpose.

Use of BGP for Label Distribution

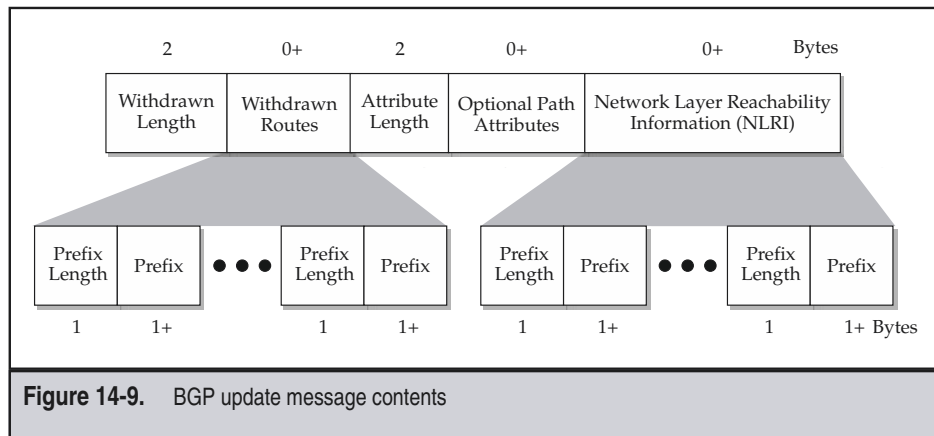
The Border Gateway Protocol version 4 (BGPv4) defined in RFC 1771 is the routing protocol that binds the interconnected set of service provider networks into the global Internet. It also empowers enterprise sites to home to multiple service providers for greater resilience and performance. Since it is the only routing protocol used between providers, RFC 3107 extends BGP to support MPLS label distribution so that interprovider LSPs could be established.

BGP has its own set of unique terminology and acronyms. An important concept is that of a unique 16-bit Autonomous System (AS) number, defined as a set of routers that makes a single exterior routing policy visible to other AS's. BGP does not communicate interior topology information between AS's, but provides information only regarding address prefixes that either are reachable within, or via transit through, the advertising AS. The use of BGP between border routers within an AS is referred to as interior BGP (iBGP), while use of BGP between routers in different AS's is referred to as exterior BGP (eBGP). BGP runs over a TCP session because it requires reliable, in-sequence delivery. It has three phases of operation: session establishment, update message exchange, and session termination. During session establishment, BGP peers in adjacent autonomous systems

exchange Open messages that contain an AS number, a keep-alive time-out value, and optional parameters, such as authentication. BGP peers periodically exchange keep-alive messages, with detection of time-out driven loss detection resulting in session termination. Following session establishment, BGP peers exchange Update messages that contain the current reachability of address prefixes, called Network Layer Reachability Information (NLRI). After an initial synchronization exchange, incremental routing changes are communicated using the Update message.

Figure 14-9 shows the contents of the BGP Update message, which consists of three parts: withdrawn routes, a list of NLRI address prefixes, and an associated optional list of attributes. Withdrawn routes and NLRI are encoded using the CIDR address/prefix length convention described in Chapter 8. BGP peers make local policy decisions regarding whether to advertise an NLRI with selected path attributes or withdraw a previous advertisement. A typical policy is selection of the NLRI with the most specific address prefix match, with a tie broken by selection of a path with the least number of AS hops.

When the Update message contains NLRI information, some path attributes are well-known mandatory, while others are optional. There are three well-known mandatory path attributes: ORIGIN, AS-PATH, and NEXT-HOP. ORIGIN identifies the source of the NLRI, for example, whether it was learned via the interior or exterior routing protocol. as-path lists a *path-vector* of the set of as's traversed so far, or an ordered sequence of AS's. Since the length of AS-PATH is often the deciding factor in choice of a route, BGP is often called a path vector routing protocol. Routers use AS-PATH to avoid loops by not forwarding advertised routes with their own AS numbers. NEXT-HOP identifies the IP address of the border router that should be used to reach the NLRI. BGP has some optional parameters that can perform a form of load balancing: LOCALPREF and MED. LOCALPREF allows the sending AS to indicate a preference for routing traffic out multiple links to another AS, while the multiple exit discriminator (MED) allows a receiving AS to indicate a preference for traffic coming from another AS.



RFC 3107 defines a specific encoding of the BGP multiprotocol extensions defined in RFC 2283 to distribute an MPLS label bound as part of the NLRI. BGP peers negotiate support for this optional capability at session establishment time. The basic procedure is to “piggyback” downstream unsolicited label distribution in parallel with normal BGP route distribution. We give an example at the end of this chapter of how this extension to BGP can be used to signal establishment of an LSP across service provider networks.

IGP TRAFFIC ENGINEERING EXTENSIONS: OSPF AND IS-IS

As described in Chapter 9, traditional routing protocols can be characterized as interior or exterior gateway protocols, called IGP and EGP, respectively. We’ve discussed signaling of LSPs using constraint-based routing, and now to complete the picture this section summarizes the extensions to IGP routing protocols used in IP networks, namely OSPF and IS-IS, necessary to distribute these constraints.

General Modifications for Traffic Engineering

Recall from Chapter 9 that the state of the art in IGP routing protocols is use of the link state paradigm, where every router in a particular domain keeps a database of the state of all links in that domain. Changes to link state are detected by neighbors and flooded throughout the interconnected set of routers in a domain to drive the distributed topology database toward a converged state. The conventional link state consists of whether a link is either up or down, along with a routing metric (e.g., distance or cost) assigned to that link. The up/down state can be viewed as a simple constraint; that is, a path that traverses a link that is down is useless. On the other hand, minimization of a routing metric is not a constraint, but an optimization criterion.

The label distribution protocols described earlier use one of the two general types of constraints for each candidate link: available capacity and resource affinity or class (also called link coloring). In order for the ingress router to compute a constrained route, the information about the additional constraints for each link is necessary. Furthermore, some metrics, such as available capacity, can change more frequently than up/down link state or link coloring, and this potential performance issue must be addressed as well. An important consideration in the design of a routing protocol with capacity constraints involves balancing the desire for an accurate and timely topology database, requiring more frequent flooding of updates, against the potential for overloading nodal route processors with too many flooded messages. Typically, flooding topology updates are made only if a significant change occurs in a constraint, if, for example, available capacity increases or decreases by some percentage, or some reasonable time since the last update occurred, to keep flooding traffic at an acceptable level. Since the topology database may be out of synch with actual network state, a distributed signaling protocol is essential to ensure that resources are reserved at each hop.

There is a basic mathematical reason for why the shortest path routing algorithm is well defined and widely implemented, while the constraint-based MPLS and ATM routing

and signaling algorithms are still undergoing standardization and refinement. Minimizing a single parameter, such as minimum distance, has a relatively simple optimal solution that the Dijkstra algorithm solves in an optimal amount of time. Adding a constraint like available capacity makes the problem considerably more complex. Hence, design of suboptimal, yet still relatively efficient, constraint-based routing and signaling algorithms is an important aspect in a backbone network.

Specific Modifications for IS-IS TE

RFC 1142 documents the OSI Intermediate System to Intermediate System (IS-IS) routing protocol, and RFC 1195 defines extensions and provides guidelines for using IS-IS to support IP routing. Since IS-IS was the most mature IGP routing implementation when Internet growth exploded in the mid 1990s, many ISPs use IS-IS. The IS-IS protocol defines a link state packet (LSP) composed of a number of TLV elements to communicate changes in topology. At the time of writing, the IETF IS-IS working group was defining traffic engineering (TE) extensions to IS-IS [ISIS TE]. The specific extensions involve redefining a TE router TLV and an IS Reachability TLV to add additional information. The TE router TLV provides a four-octet ID that identifies the ingress router of an LSP, and since it is not tied to any interface, the TE router ID becomes a stable reference.

The work also proposes a new sub-TLV that can be defined within an IS-IS TLV, which is the method used to encode TE information in the IS Reachability TLV. This includes the administrative group (also known as resource affinity, resource class, or link color), a TE default metric (which can be different from the shortest path metric), and several bandwidth-related link-level parameters as follows. Maximum bandwidth is the actual capacity of the unidirectional link in bytes per second, while the maximum reservable bandwidth may be greater than this value if oversubscription (see Chapter 20) is employed. The currently unreserved bandwidth is that portion of the maximum reservable bandwidth that is not allocated to any LSP.

Specific Modifications for OSPF-TE

RFC 2328 specifies version 2 of the Open Shortest Path First (OSPF) protocol. OSPF uses a link state advertisement (LSA) composed of a number of TLV elements to communicate topology information. RFC 2370 defines an opaque LSA for carrying arbitrary information, as well as a means to limit the scope of flooding such LSAs to manage the potential impact on routers. The opaque LSA provides a general way to extend OSPF, and at the time of writing, the IETF's OSPF working group was working on traffic engineering extensions [OSPF TE] using the opaque LSA type to distribute attributes and metrics that are essentially identical to those being defined in the ISIS-TE extensions. In particular, an OSPF-TE router address TLV is a four-octet IP address that is comparable in meaning to the IS-IS TE router TLV and can be used to compute a mapping between the OSPF and IS-IS topologies. The OSPF-TE link TLV has essentially the same information as contained in the

ISIS-TE IS Reachability TLV described earlier. That is, it contains a TE metric that can be different from the shortest path metric, a resource class or color, as well as the traffic engineering metrics of maximum bandwidth, maximum (potentially oversubscribed) reservable bandwidth, and currently unreserved bandwidth.

Open Issues and Challenges Ahead

By definition, an IGP is used only within a single network. Since the traffic engineering extensions to an IGP described previously exist only within a single network, it is not possible with current standards to establish a traffic-engineered or constraint-routed LSP across networks. We described the extensions to BGP earlier as the only method available to distribute MPLS labels across networks, which can provide reachability but only has the means to apply a constraint regarding preference of one path over another locally.

Furthermore, the standard support for traffic-engineered LSPs may not even reach across a single network. Both IS-IS and OSPF define means to structure an IGP using a hierarchy where, in OSPF (IS-IS) terminology, a backbone (level 2) area interconnects a number of stub (level 1) areas. There have been a number of proposals and discussions within the IETF to use approaches for supporting traffic-engineered LSPs across a hierarchical IGP. Some are similar to those defined in the ATM Forum PNNI specification for handling a similar problem, as described in the next chapter; however, at publication time, there was not yet a formally adopted IETF work item to support traffic-engineered LSPs across a hierarchical IGP. Solution to these problems will be important in large networks that use MPLS to support applications like pseudo-wire emulation and VPNs.

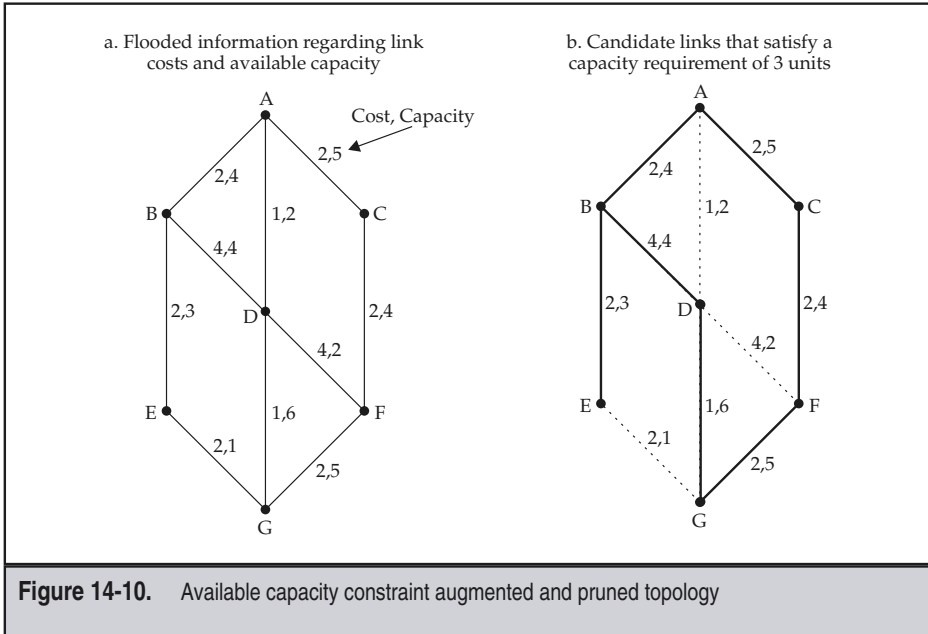
EXAMPLE APPLICATIONS OF MPLS IN IP NETWORKS

In order to place the previously described architectural, signaling, and routing protocol details in a practical context, this section presents several examples that use the various label distribution protocols in real-world network applications.

Traffic Engineering in an IP Backbone

This section presents an example of the functions involved in *constraint-based* signaling and routing protocols in a network routing application requiring a constraint of available capacity. This is a common mode of not only MPLS control protocol operation, but also the ATM PNNI protocol described in the next chapter. The reader may wish to compare this example with the example of Chapter 9 that illustrated the operation of routing without constraints to appreciate the differences between these approaches.

Figure 14-10a illustrates a network where the interior network routing protocol (e.g., ISIS-TE, or OSPF-TE) floods not only the link cost, but the available capacity on the link as well, as shown by the pair of numbers next to each link. Since routers allocate and release



capacity in response to network events, the available link capacity portion of the topology database is ever changing. In Figure 14-10b, router A receives an LSP establishment request with destination router G requiring three units of capacity. In order to select a route subject to the capacity constraint, router A removes all links that fail to meet the capacity constraint of three units. We say that the routing algorithm prunes the links with insufficient capacity from the topology database, as illustrated by dashed lines in the figure.

Now, router A can run a shortest-path algorithm on the pruned topology database to find the path with the least cost. As shown by the arrow in Figure 14-11a, the shortest path is the explicit route that traverses routers A-C-F-G. In the example, as the routers pass an explicit routed signaling message along (e.g., RSVP-TE or CR-LDP), the underlined router letter in the figure indicates the next router along the explicit route. The signaling message retains the entire path so that a response can return through the same routers. If the request encounters insufficient capacity at a particular router, that router can *crank back* the request to the source router, which can then try an alternative route. Figure 14-11b illustrates the network topology database some time after allocating capacity to the connection between routers A and G.

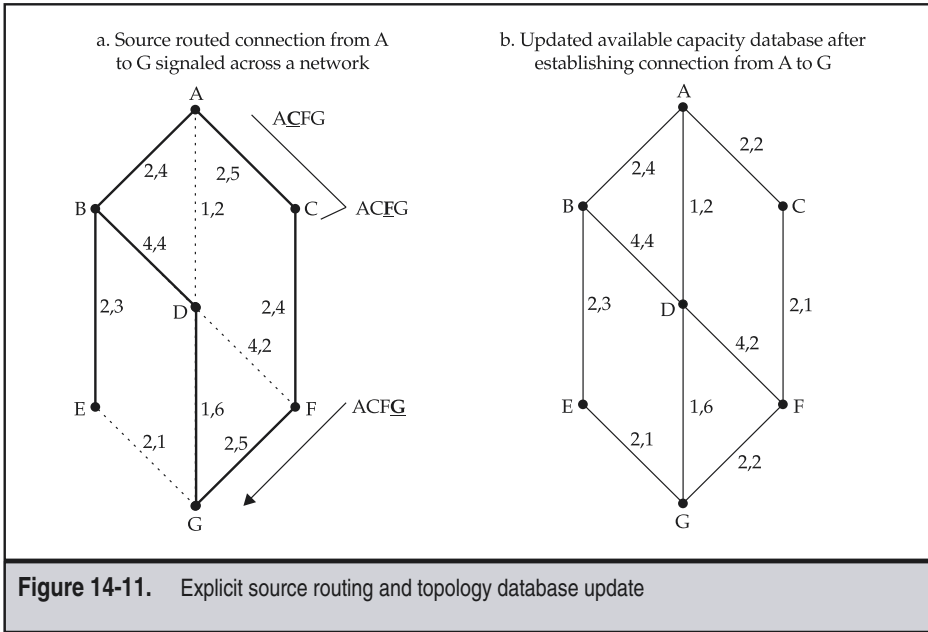
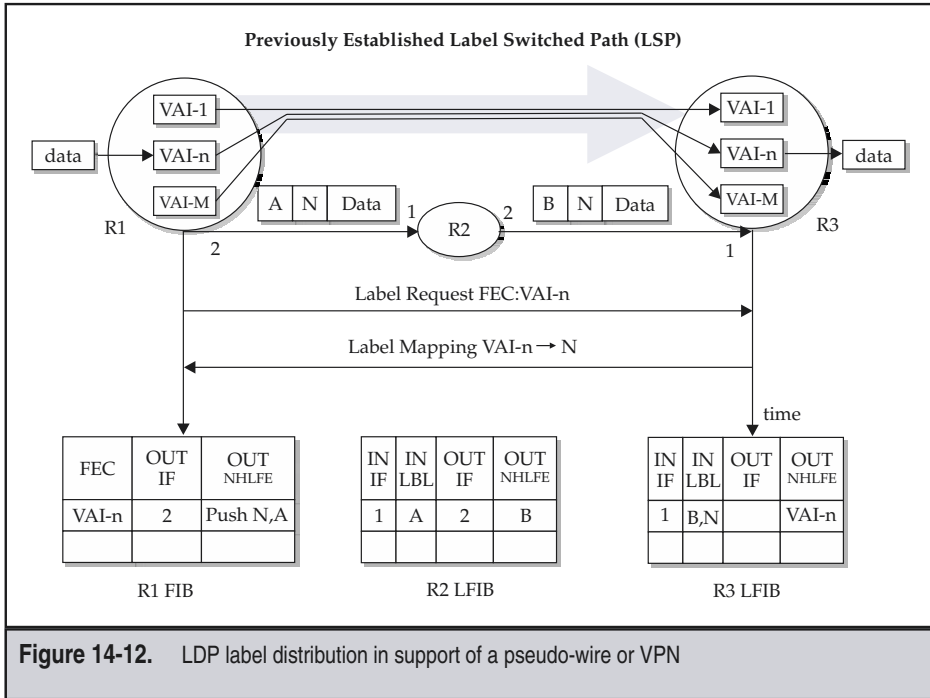


Figure 14-11. Explicit source routing and topology database update

Label Distribution in Support of Other Services

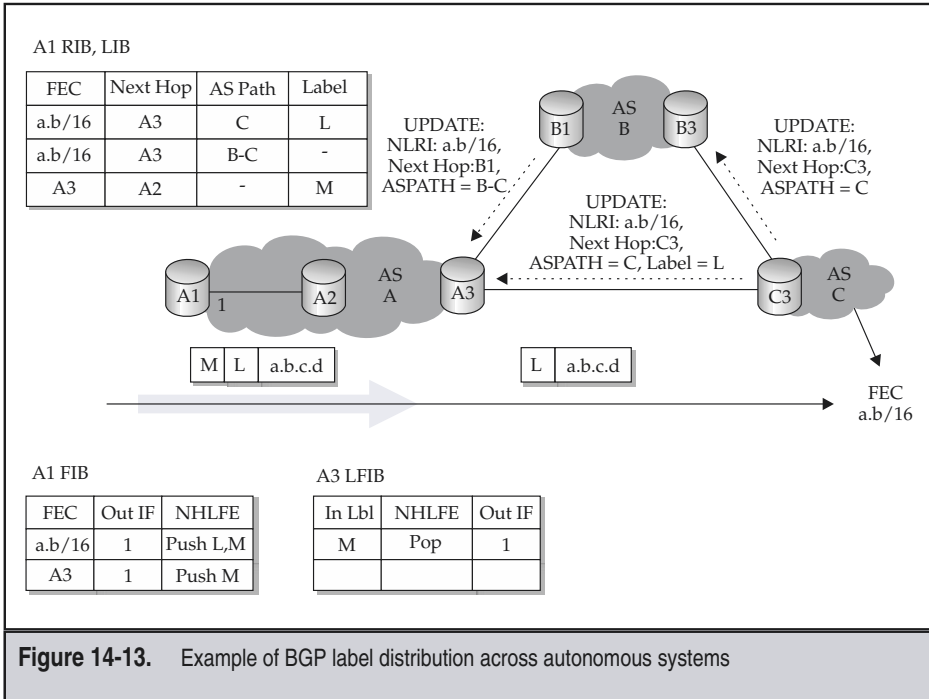
Figure 14-12 illustrates use of the label distribution protocol (LDP) in a label stacking-based application, for example, for a pseudo-wire or a layer 2 or layer 3 VPN, as further described in Part 5. The figure shows a pair of routers R1 and R3 on the left- and right-hand sides containing a number of virtual application instances (VAI), for example a pseudo-wire or a L2/L3 VPN forwarding table. The requirement is that data for each virtual application be delivered over a shared, previously established LSP shown by the large arrow between the two routers hosting these virtual instances of applications. The middle of the figure shows the pair of LDP messages necessary to exchange a label binding between virtual application instance n (VAI- n) in R1 and its counterpart in R3. Similarly, such exchanges could occur (or could be statically configured) between other VAIs, as shown by the multiple arrows sharing the previously established LSP between R1 and R3.

As introduced in Chapter 12, such a stacked label could be implemented over an LSP tunnel or an IP tunnel. Essentially the same LDP procedure summarized previously for distribution of the innermost label would be used in either case.



MPLS Connectivity Across Multiple Providers

BGP can be used to distribute labels for LSPs that traverse more than one service provider. Figure 14-13 illustrates an example involving three autonomous systems: A, B, and C. AS C has allocated address prefix (i.e., FEC) a.b/16 to a customer. Router C3 advertises it as an NLRI to AS A and AS B using a BGP UPDATE message, which also contains the next hop router and the ASPATH as shown in the right-hand side of the figure. The UPDATE message sent by C3 to A3 includes a mapping from FEC a.b/16 to a label L. Router A3 in AS A collects all of these advertisements in its RIB, for example through a mesh of iBGP sessions or a route reflector. In determining how best to forward packets to a.b/16, A1 is able to determine that the shortest AS path is via next hop A3 using label L. Router A1 also knows from its interior routing and label distribution protocol that the best route to A3 is via A2 using the label M. Router A1 is able to determine from the BGP and interior routing information that the best path to a.b/16 results from first pushing on the label L and then the label M. As shown at the bottom of the figure, there is one LSP tunneled inside the other. The first outermost LSP extends from A1 to A3, while the second extends from A3 into AS C.



This example also illustrates a fundamental shortcoming to the path-vector routing paradigm in that there is no metric or routing constraint available for selection of a route. The only mechanisms available are configuration of optional parameters that would statically prefer one route to another.

REVIEW

This chapter began by introducing the basic MPLS control plane signaling and routing protocol functions in support of automatic configuration of the forwarding plane. We illustrated how constraint-based routing is the motivation for adding signaling to the base IP routing infrastructure. The text then summarizes how MPLS signaling protocols distribute labels according to a number of generic characteristics, such as whether the route is hop-by-hop or explicit, whether labels are distributed unsolicited or on demand, and whether control of label distribution is independent or ordered. The text then summarized how the label distribution protocols defined to date map to this taxonomy and then

described the label distribution aspects of each protocol: LDP, RSVP-TE, CR-LDP, and BGP. We then summarized the extensions necessary to an interior routing protocol (i.e., IS-IS or OSPF) in support of constraint-based routing. Finally, the chapter concluded with several examples of how MPLS signaling and routing can be used in traffic engineering, support of other services over MPLS, or MPLS configured across multiple networks.

CHAPTER 15

ATM NNI Signaling and Routing Protocols

This chapter covers the topic of ATM Network-Node Interfaces, also known as Network-Network Interfaces, both abbreviated as NNI. These two meanings of the same acronym identify its dual purpose for use between nodes in a single network, as well as interconnection between different networks. This chapter begins in the private network domain by covering the ATM Forum's early Interim Inter-switch Signaling Protocol (IISP) and its successor, the Private Network-Network Interface (PNNI). Some of the functions pioneered in PNNI, like explicit routing, have been applied to the MPLS protocol, while others, like constraint-based routing across a hierarchically summarized topology, have not. The chapter then summarizes the public network domain NNIs by covering the ATM Forum's (AINI), as well as the ITU-T Broadband ISDN Services User Part (B-ISUP). These protocols involve only signaling, with routing determined by manual configuration, similar to telephone networks.

INTERIM INTERSWITCH SIGNALING PROTOCOL (IISP)

Early on, the ATM Forum recognized the need to produce a standard for a minimum level of interoperability for multivendor private ATM networks. Therefore, the Forum rapidly developed and published the Interim Inter-switch Signaling Protocol (IISP) in late 1994 [AF IISP]. The Forum announced that the IISP standard would fill the void until it could complete the PNNI specification. IISP extended the UNI 3.0/3.1 protocol to a simple network context where each side of a UNI was configured as the network-side master and a user-side slave. The IISP physical layer, ATM layer, and traffic management specifications are identical to the UNI 3.0/3.1 specification. IISP employs the UNI cell format, includes no ILMI, and makes policing optional. IISP specifies a limited set of VCIs ranging from 32 to 255 on VPI 0 to ensure interoperability.

IISP utilizes the NSAP-based ATM End System Address (AESA) format described in Chapter 13 and defines simple hop-by-hop routing based upon matching the longest address prefix in a statically configured routing table. Such manual configuration limited the scalability of IISP networks and is difficult to manage and administer. Furthermore, manual configuration of hop-by-hop routing tables may introduce routing loops, a potential problem the IISP specification identified but provided no guidance on how to avoid. Also, a switch must clear the SVC if it detects a link failure on any IISP interface, a function normally performed only by the network side of a UNI.

PRIVATE NETWORK-NETWORK INTERFACE (PNNI)

The abbreviation PNNI initially stood for either Private Network-Node Interface or Private Network-to-Network Interface, reflecting its two possible uses for connecting nodes within a network or interconnecting networks. The most recent version of PNNI describes explicit support for public networks. Early ATM networks, for example, IISP, required extensive amounts of manual configuration, which led to errors and connectivity failures. The ATM Forum responded to these challenges in 1996 by defining the mother of all routing and

signaling protocols—PNNI [AF PNNI 1.0]—to achieve the goal of automatic configuration along with multivendor interoperability of ATM hardware and software.

Architecture and Requirements

The PNNI protocol specifies interrelated routing and signaling protocols and functions to achieve the goal of controlling ATM connections established between nodes and networks as illustrated in Figure 15-1. Comparing this with the MPLS architecture in Chapter 14, the reader will see a similar split of control and forwarding (or user) plane

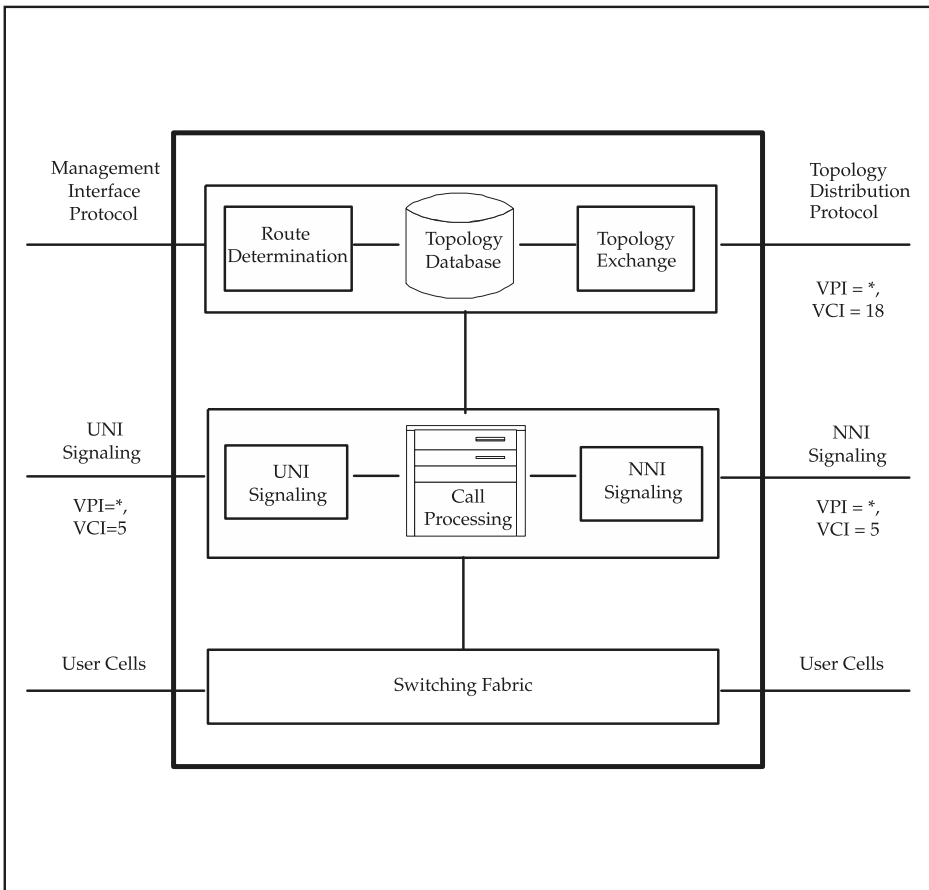


Figure 15-1. PNNI switching system architectural reference model

functions, along with a routing and signaling protocol in the control plane. PNNI protocols operate over dedicated links, or else tunnel over VPs, as denoted by the $VPI=*$ notation in the figure. A topology exchange protocol defines the methods and messages for disseminating information between switches and clusters of switches used in the routing of connections. PNNI employs a recursive, hierarchical design that scales to huge network sizes. A key feature of the PNNI hierarchy mechanism is its ability to automatically configure itself in networks with an address structure that reflects the topology.

The PNNI signaling protocol uses messages to establish point-to-point and point-to-multipoint connections across an ATM network. The PNNI signaling protocol uses ATM Forum UNI signaling specification as a basis, augmenting it with information elements to support source routing, and the ability to crank back to earlier nodes in order to route around an intermediate node that blocks a call request. The PNNI specification also defines SVC-based Soft Permanent Virtual Paths and Channel Connections (SPVPCs and SPVCCs) that allow automatic provisioning and restoration of ATM VCCs and VPCs.

Version 1.0 of the ATM Forum's PNNI specification [AF PNNI 1.0] was released in 1996. It is an interesting standard in that it was less influenced by vendor competition, since everyone was starting with a clean slate instead of developing an approach based upon one vendor's proprietary design. It supported all UNI 3.1 and most UNI 4.0 signaling and traffic management capabilities. Several design objectives were paramount in the development of PNNI 1.0: hierarchical routing enabling scaling to very large networks, support for constraint-based, QoS-aware routing, and automatic discovery of topology and address reachability. Also, support for anycast signaling and group addresses that were important in LAN emulation and IP over ATM applications were important drivers as well. Finally, consideration was also given to interoperation with external domains that did not use PNNI.

Version 1.1 of the PNNI specification [AF PNNI 1.1] fixed errors in the preceding version, rolled up a number of addenda into a single (lengthy) document, and also added some important functions. First, it adds support for all UNI Signaling 4.1 capabilities (except proxy signaling and virtual UNIs). It also adds support for point-to-multipoint SPVPCs, as well as native Frame Relay SPVPCs or FR/ATM interworking SPVPCs. The update also specifies a number of other refinements based on operational experience related to addressing support, topology update triggers, passing of information, and enhanced status signaling and routing.

Network Addressing Philosophy

Network addressing philosophies fall between two ends of a spectrum: flat and hierarchical [Alles 95]. Scalability is an important measure of how nodal memory and processor requirements grow with the number of devices in a network. Devices with flat addressing, such as bridged LANs, must have a routing table with an entry for every other device in the network, and therefore networks with flat addressing scale linearly with the number of nodes. On the opposite end of the spectrum, hierarchical addressing assigns significance to portions of the address. The significance assigned to an address is often geographical or organizational. A familiar example of geographic assignment is that of

the telephone network. The leftmost digits of an international phone number identify the country in which the addressed device resides (e.g., 01 for North America, 44 for the UK) and then the identified country defines the remaining digits. Most countries employ a geographically oriented telephone number address assignment scheme. The Internet primarily utilizes organizational address assignment by assigning address prefixes to organizations (e.g., service providers or large enterprises) and not countries or geographic areas. However, some organizations manage their IP address spaces such that a geographic dimension exists in parallel with the organizational hierarchy. However, no standards dictate this type of address assignment.

An undesirable side effect of hierarchical addressing is low utilization of the available address space. Sparse fill of address space occurs because organizations leave room for growth, or perhaps a network design dictates peer relationships between groups of devices with widely different populations. The ATM Forum's choice of the 20-octet NSAP-based address format described in Chapter 13 for PNNI meets these constraints well, since there is never likely to be a network that approaches a size anywhere near PNNI's theoretical limit of 2^{160} (approximately 10^{48}) nodes. In practice, however, the real number of usable addresses is much less. The PNNI addressing plan provides an unprecedented level of hierarchy, supporting up to 105 levels; however, most implementations implement at most four or five levels. PNNI exploits the flexibility of such a huge address space with the objective of providing an extremely scalable network in the specification of its routing protocols.

A Tale of Two Protocols

As introduced earlier, two separate PNNI protocols operate between ATM switching systems connected by either physical or virtual PNNI links: signaling and routing. The signaling protocol sets up the ATM connection along the path determined by the routing protocol. The routing protocol utilizes two types of addresses—topology and end user—in a hierarchical manner. Through exchange of topology information over PNNI links, every node learns about a hierarchically summarized version of the entire network topology. The distribution of reachability information along with associated metrics, such as administrative cost to reach a particular address prefix over a PNNI link, is similar to that used in the open shortest path first (OSPF) protocol described in Chapter 9.

Given that the source node has a summarized, hierarchical view of the entire network and the associated administrative and quality metrics of the candidate paths to the destination, PNNI places the burden of determining the route on the source. The information about the source-to-destination path is computed at the source node and placed in a Designated Transit List (DTL) in the signaling message originated by the source. Intermediate nodes along the path expand the DTL in their domain, and crank back to find alternative paths if a node within their domain blocks the call. Hence, PNNI DTLs are similar to token ring networks which employ source routing. Furthermore, source routing explicitly prevents loops, and also a standard route determination protocol isn't necessary, simplifying interoperability. The MPLS signaling protocol use of explicit routing resulted from experience with PNNI source routing.

The PNNI signaling protocol defines extensions of UNI signaling through Information Elements (IE) for parameters such as the DTL, Soft PVC (SPVC), and crankback indications. The PNNI signaling protocol utilizes the same virtual channel, VCI 5, used for UNI signaling. The VPI value chosen depends on whether the NNI link is physical or virtual.

The PNNI routing protocol operates at the virtual circuit level to route messages from the signaling protocol through the ATM network toward their destination. This protocol operates over AAL5 on VCI 18 of VPI 0 for a dedicated link, or some other nonzero VPI for a PNNI virtual path tunneled across another PVC ATM network.

Although PNNI builds upon experience gained from earlier routing protocols, its complexity exceeds that of any routing protocol conceived to date. As subsequent sections illustrate, the complexity of PNNI stems from the need to support requirements on scalability, support for QoS-based routing, and the additional complexities of supporting a connection-oriented service with guaranteed bandwidth. We saw in the preceding chapter the need for a separate signaling protocol along with necessary extensions to OSPF and IS-IS in order to provide similar functions for MPLS.

PNNI Routing Hierarchy and Topology Aggregation

PNNI employs the concept of embedding topological information in hierarchical addressing to summarize routing information. This summarization of address prefixes constrains processing and memory space requirements to grow at a rate slower than the number of nodes in the network. At each level of the hierarchy, the PNNI routing protocol defines a uniform network model composed of logical nodes and logical links. PNNI proceeds upward in the hierarchy recursively; that is, the same functions are used again at each successive level. The PNNI protocol defines:

- ▼ Neighbor discovery and link status determination via a Hello protocol
- Topology database synchronization procedures, for example as used at startup time
- Peer-group determination and peer group-leader election to implement the hierarchy
- Reliable PNNI Topology State Element (PTSE) flooding
- ▲ Bootstrapping of the PNNI hierarchy from the lowest level upward

Figure 15-2 depicts the example network used in the remainder of this section to illustrate PNNI. The example denotes addresses using a label of the form a.b.c to denote common address prefixes. In these addresses, “a” represents the world region (where P represents the Pacific Rim, A the Americas, and E Eurasia), “b” represents the next lower level of hierarchy, and “c” represents the lowest level of hierarchy. At the lowest level (e.g., a single site) the addressing may be flat, for example, using LAN MAC addresses. The line from P.1 to the left and the line from E.3.2 to the right indicate a connection between P.1 and E.3.2 in the following.

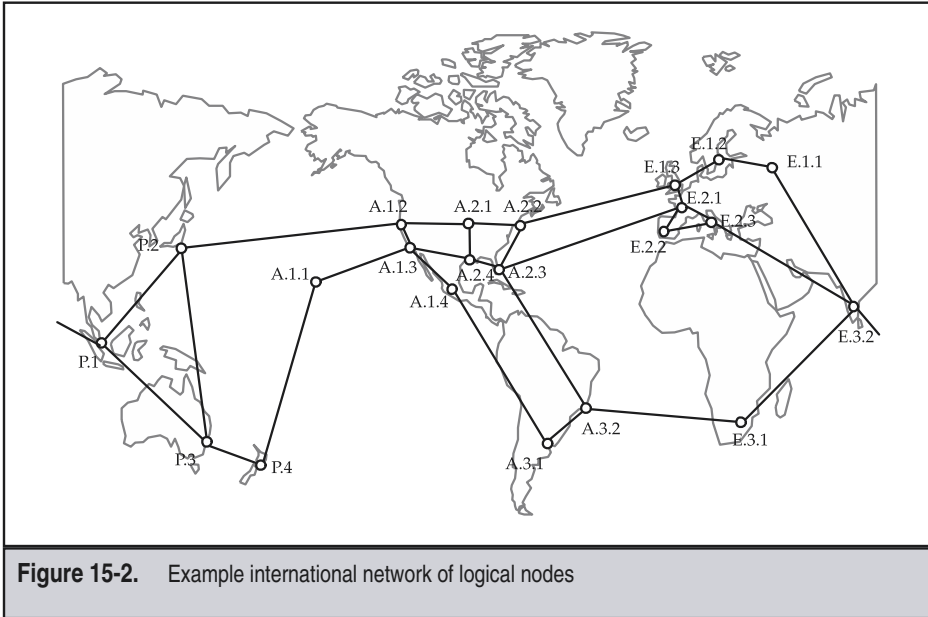
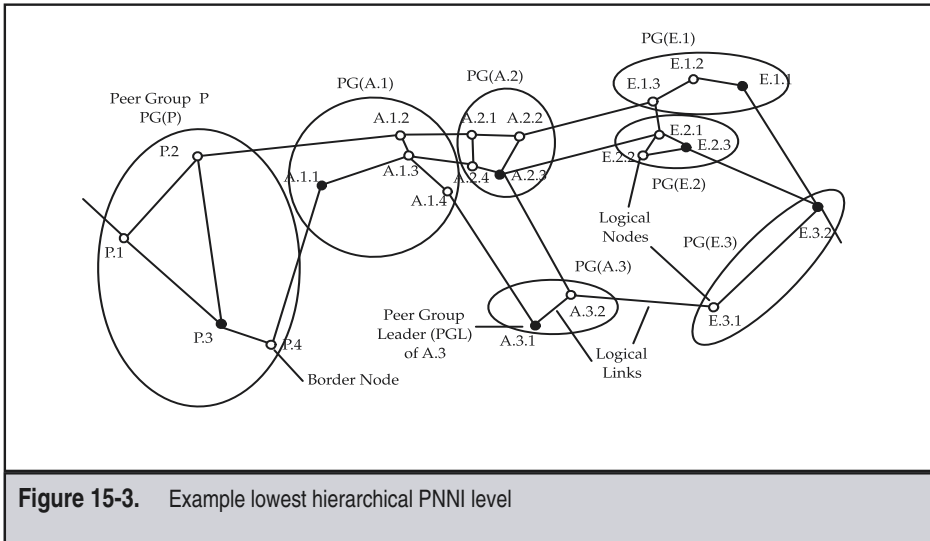


Figure 15-2. Example international network of logical nodes

PNNI Terminology and the Lowest Hierarchical Level

The PNNI routing hierarchy builds from the lowest-level logical nodes and links depicted in Figure 15-3. A lowest-level node may be a physical system, or a network of switches that supports PNNI for external connectivity. The diagram doesn't depict end systems. PNNI has its own vocabulary, which we now introduce. A *logical link* may be either a physical link or a VPC PVC across another ATM network. PNNI defines logical links as either *horizontal links* when they connect logical nodes within a peer group, or as *outside links* if they connect peer groups. PNNI also defines *exterior links* that connect nodes to other networks that don't use the PNNI protocol, for example, a public ATM SVC network. PNNI uses UNI signaling to dynamically signal connection establishment and release requests with these networks. Of course, the PNNI routing protocol terminates at these exterior links.

Several important PNNI identifiers use the ATM End System Address (AESA) defined in Chapter 13. A 19-octet AESA uniquely identifies each logical node. The 20th octet, the selector byte, has other uses described later. By configuration, lowest-level logical nodes are part of a *peer group (PG)* identified by a one-octet level indication and a prefix of at most 13 octets of an AESA. Nodes within a peer group share a common address prefix whose length determines the level in the PNNI hierarchy. For example, the nodes with addresses of the form A.1.x are all in the peer group A.1. A peer group leader election



process selects one node automatically based upon its configured priority. Border nodes have one or more links crossing a peer group boundary, as shown in Figure 15-3.

Dynamically Building the PNNI Hierarchy

When logical nodes (or logical links) transition to the active state, a Hello protocol executes on VCI 18, called the PNNI *Routing Control Channel (RCC)*. Periodically transmitted Hello packets convey the logical node's AESA, node ID, peer group ID, and port ID. The *Hello protocol* detects link failures via a timeout on unacknowledged packets. Once nodes acknowledge each other via the Hello protocol, they begin a database synchronization process. Newly acquainted nodes exchange PNNI Topology State Element (PTSE) headers to determine if they are in synchronization. For efficiency, one or more PTSEs are encapsulated in a PNNI Topology State Packet (PTSP). The protocol resolves mismatches by accepting the latest PTSE. As a result of these information exchanges, neighbor nodes synchronize their databases and hence learn the overall network topology in an efficient manner.

Once database synchronization completes, nodes flood information about their nodal and link attributes and metrics throughout their peer group. PNNI defines a reliable flooding protocol where every PTSE has a unique identifier containing a timestamp. When a node within a peer group receives a PTSP, it floods the PTSEs not already received on all other links except the link on which it received the PTSP containing the new PTSE. Thus, since PTSPs are acknowledged, the reliable flooding protocol ensures that the topology database of every node in the peer group *converges* to a common state. As in other link state protocols, PNNI sends PTSPs at regular, but not too frequent, intervals.

Nodes also send PTSE updates when a significant event occurs, such as a link failure, a large change in allocated bandwidth, or a blocked connection attempt.

Figure 15-4 illustrates how the Hello protocol used by border nodes establishes uplinks that dynamically build the PNNI hierarchy. The resulting higher-level peer group elects a leader, which then floods summarized topology data within its higher-level peer group. PNNI 1.1 defined optional procedures that better controlled the volume of flooded information. Higher-level logical nodes construct logical horizontal links, sometimes collapsing multiple physical links into a single logical link. For example, the parallel links A.1.2–A.2.1 and A.1.3–A.2.4 collapse into the one link A.1–A.2 at the next higher level in the hierarchy. As illustrated in the figure, PG(A) is the *parent peer group* of all nodes with addresses of the form A.x, who are each a *child peer group* of A. Higher-level peer group IDs have a common prefix with the IDs of all nodes within their peer group. For example, all nodes in PG(P) have a common address of the form P.x, where x indicates an arbitrary number. Recall that the line from P.1 to the left and the line from E.3.2 to the right indicate a connection between P.1 and E.3.2.

PNNI identifies nodes using a 22-byte node identifier. For nodes present only at the lowest level, the first octet specifies the level of the node's containing peer group, the second octet uniquely identifies it as a lowest-level node, followed by the node's 20 octet AESA. For *logical group nodes (LGNs)* participating at higher levels in the PNNI hierarchy, the first octet is the same, followed by further identification, for example, the 14-octet peer group ID and a six-octet ESI of the end system implementing the LGN function.

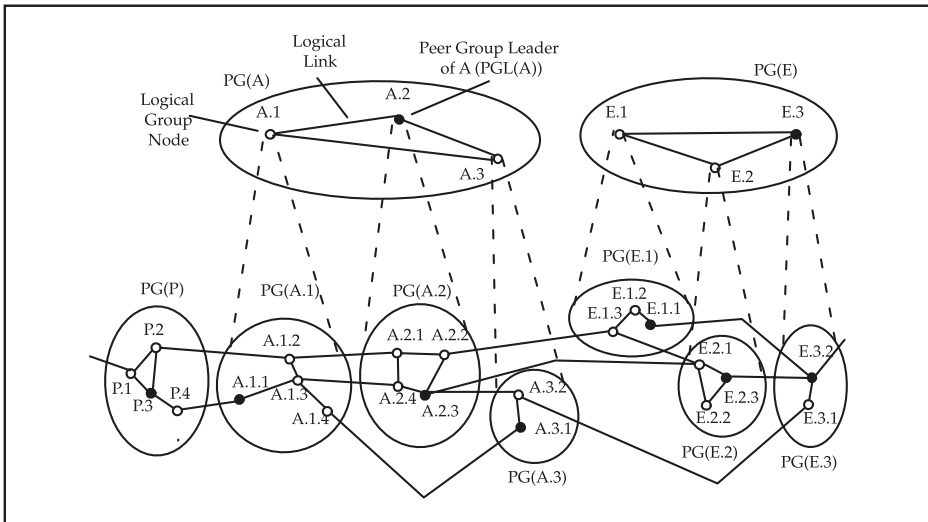


Figure 15-4. Building the PNNI hierarchy

The PNNI protocol also advertises links in PTSPs. Since ATM link attributes may be asymmetric, and actual resource usage can differ in each direction, PNNI identifies links by a combination of a transmitting node ID and a locally assigned port ID. The Hello protocol exchanges these port IDs. PNNI also defines additional logic to handle the case when multiple links connect adjacent nodes.

Topology Aggregation and Complex Node Representation

All network nodes estimate the current network state from the attributes and metrics contained in advertisements from the flooded PTSEs about both links and nodes. Typically, PTSEs include bidirectional information about the transit behavior of particular nodes based upon entry and exit port, taking into account the current internal state. Nodal information often represents an aggregated network (also called a *summarized peer group* in PNNI parlance) and not just a single switch. The complex node representation presents the result of this aggregation in a parent peer group or a lowest-level node. The default representation uses a symmetric star topology centered on a node called the *nucleus* connected to ports around the circumference of the logical node via spokes. Spokes may have either default attributes or exception attributes. The concatenation of two spokes represents traversal of the symmetric peer group.

Unfortunately, the real topology of a peer group is often asymmetric; therefore, the default representation may hide important topology state information. The complex node representation models such differences by using exceptions to represent particular ports whose connectivity to the nucleus is significantly different from the default. Furthermore, exceptions can represent connectivity between two ports significantly better than that implied by traversing the nucleus. An example of an exception is a high-speed, underutilized connection between border nodes of the peer group.

We illustrate these concepts through an example. Figure 15-5a shows the details of Peer Group A.2's available bandwidth for each link. Figure 15-5b shows a possible complex node representation of A.2 using the available bandwidth attribute. The complex node representation need only advertise two values for the link available bandwidth; the default value of 40 is chosen as the minimum of all links in the peer group, with a single exception of 100. Some accuracy is lost in this example, but aggregation achieves a 50 percent reduction in topology data over the nonhierarchical method of advertising the available bandwidth for every link.

Aggregation reduces complexity and increases scalability by reducing the amount of information required in PTSP flooding. PNNI allows the network designer to trade off improved scalability using aggregation against loss of detailed information necessary for optimized source routing. Practically, the exchange of full topology information limits peer group size due to processing, storage, and transmission of topology data in PTSPs. Complex node aggregation reduces topology information advertised up, across, and back down the hierarchy by logical group nodes, as long as the administrator controls the number of advertised exceptions. Although aggregation reduces the volume of topology data, the processing required inserts some additional delay. However, since other parts of the network require less processing time for the smaller volume of aggregated data, the overall convergence time typically decreases.

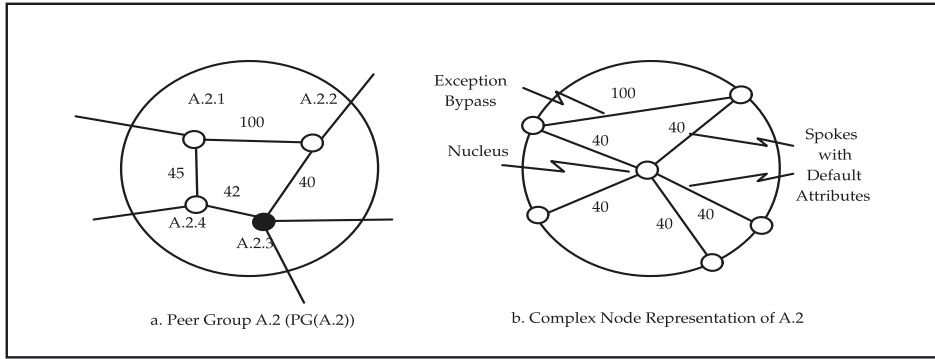


Figure 15-5. Complex node representation example

Completing the PNNI Hierarchy

Figure 15-6 completes the hierarchy for this example, recursively employing the previous concepts and protocols. Peer group leaders formulate summarized versions of their

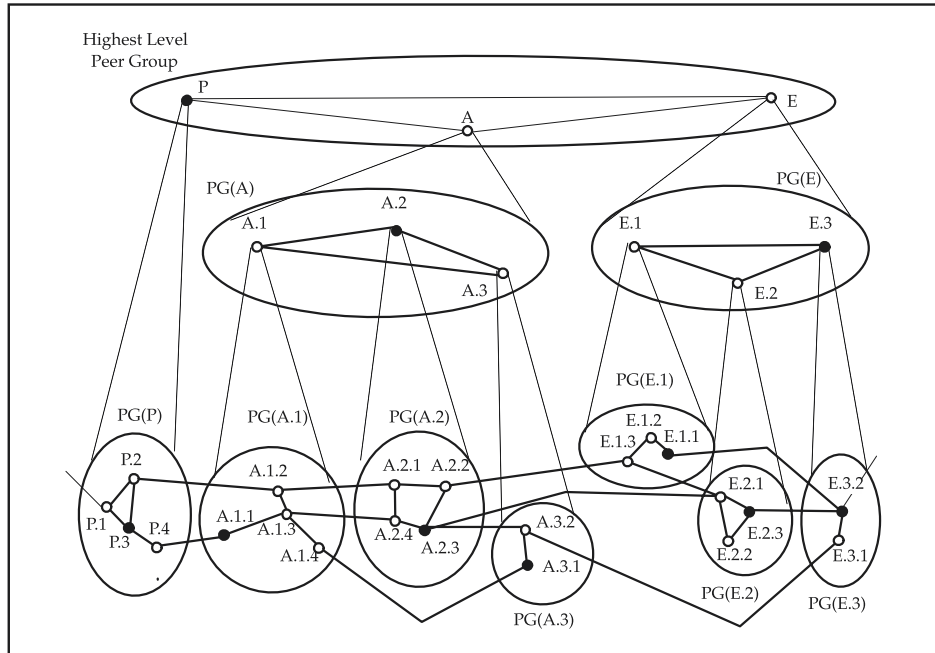


Figure 15-6. Complete PNNI hierarchy for international network example

children using the complex node representation, format these into PTSEs, and flood these in PTSPs within their higher-level peer group so that each node in the higher-level peer groups has a common view of the summarized topology, as illustrated in the figure.

Furthermore, the LGNs flood the higher-layer topology information within their child peer groups so that lower-level topology databases also contain the same view of the summarized overall network topology. This information is essential for PNNI's source routing paradigm to function. Figure 15-7 illustrates the summarized topology view from node A.1.3's point of view. Uplinks, shown as bold dashed lines in the figure, actually bootstrap the PNNI hierarchy. When a border node detects that an adjacent node is in a different peer group, it creates an uplink and advertises topology information via PTSEs that advertise this to its respective peer groups. For example, node A.1.1 is adjacent to the peer group P, while node A.1.2 is adjacent to both peer group P and peer group A.2. Every member of each peer group is then aware of the uplink. There is also an induced uplink from A.1 to P, derived by aggregation of the uplinks from A.1.1 and A.1.2 to P. Similarly, induced uplinks are derived between A.2 and E as well as A.3 and E, which is what enables A.1.3 to compute a route to E. This information also allows peer group leaders to

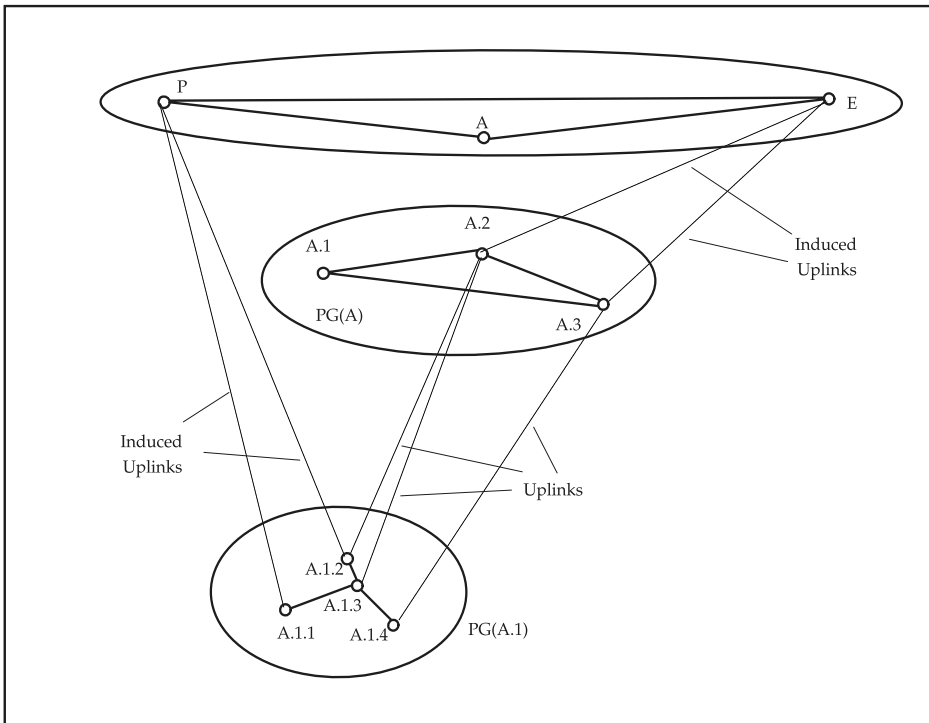


Figure 15-7. Lowest-level node (A.1.3) view of summarized network topology

establish an RCC via an SVC at the next level of the hierarchy. The peer group leaders effectively become logical group nodes at the next level up in the PNNI hierarchy and perform similar functions to logical nodes at the lowest level. A major difference is that in higher-layer peer groups, the Hello and topology synchronization protocols operate over the SVC RCC, and not a physical link.

Note how the information fed down the hierarchy represents an increasingly more summarized view of the remainder of the network. The lowest-level node has a complete view of the topology of its peers; however, information progressively decreases further up the hierarchy as peer groups are aggregated.

Reachability and Scope

In addition to summarized addresses, a number of other elements of reachability information are also carried within a PTSP, as controlled by manual configuration. Routes to external networks, reachable across exterior links, are advertised as external addresses. Peer groups may also include nodes with nonaggregatable addresses, which must also be advertised, for example, registered group and anycast addresses. Generally, none of these types of information can be summarized, since they fall outside the scope of the default PNNI address hierarchy. The scope of advertisement of group addresses is controlled by how the scope of a registered node is mapped to the PNNI hierarchy.

Beyond Connectivity to Quality and Bandwidth

PNNI supports two of the fundamental tenets of ATM via its routing and signaling protocols—guaranteed Quality of Service (QoS) and reserved bandwidth on a per-connection basis. The routing protocol distributes available bandwidth, cost, and QoS metrics across the network hierarchy. The source node utilizes these metrics to choose the best route that meets the requested bandwidth and QoS constraints. The source specifies the preferred route in a Designated Transit List (DTL) in the SETUP message.

PNNI 1.1 supports all TM 4.1 service classes: CBR, VBR-rt, VBR-nrt, ABR, UBR, and GFR (see Chapter 20 for details). ATM switches apply Connection Admission Control (CAC) upon receipt of a SETUP message from the preceding switch. The CAC function admits the connection only if QoS would still be guaranteed for existing connections after admitting the new request. If admitted, a node then forwards the SETUP message onto the next switch in the DTL. If CAC fails, then the switch clears the request back to the preceding switch. CAC is an implementation-dependent function, determined by factors such as the switch architecture, buffer structure, and queuing implementation.

An efficient implementation assures that the source route determined by the origin node specifies a DTL with a high likelihood of success. PNNI meets this objective by flooding QoS and bandwidth PTSEs in addition to reachability metrics. The result is that all nodes develop a summarized view of the support for QoS and available bandwidth across the entire network. Flooding of topology information occurs whenever a significant event occurs, such as when a new node joins the network, when a link or node fails, when available bandwidth changes dramatically, or when CAC denies a SETUP message request.

PNNI defines two types of link parameters: nonadditive link attributes, which are applied to a single network link or node in route determination, along with additive link metrics, which are applied to an entire path of nodes and links. The set of *nonadditive link attributes* in PNNI are:

- ▼ Available Cell Rate (ACR): A measure of available bandwidth in cells per second for each traffic class.
- ▲ Cell Rate Margin (CRM): A measure of the difference between the effective bandwidth allocation per traffic class and the allocation for sustainable cell rate. CRM is the safety margin allocated above the aggregate sustained rate.

The set of *additive link metrics* in PNNI include explicit support for QoS parameters and other parameters useful in selecting a candidate route:

- ▼ Maximum Cell Transfer Delay (MCTD) per traffic class.
- Maximum Cell Delay Variation (MCDV) per traffic class.
- Maximum Cell Loss Ratio (MCLR) for CLP=0 cells, for the CBR and VBR traffic classes.
- Administrative Weight: This value indicates the relative desirability (i.e., cost or shortness) of a network link in a manner analogous to OSPF.
- ▲ Variance Factor (VF): A relative measure of CRM margin normalized by the variance of the aggregate cell rate on the link.

Estimating and Refining Source Routes

Recall that the source PNNI node determines a path across the network based upon the requested QoS and its knowledge of the network state obtained from flooded PTSEs. In a dynamically changing network, the source node has only an imperfect approximation to the true network state. This inconsistency occurs because topology information obtained from PTSE flooding may be out of synch with the actual current network state. Furthermore, the topology aggregation process hides details to improve scalability. Additionally, each node along the path may perform CAC differently than the source node's estimate, which motivates specification of generic CAC to minimize this potential discrepancy. The net result of these considerations is that the source node's best estimate of the ideal path may not result in a successful connection attempt, and this is why the ATM Forum defined the crankback procedure.

Generic Connection Admission Control (GCAC)

The PNNI protocol attempts to minimize the probability of failure in the determination of the first source route by defining a Generic CAC (GCAC) algorithm, which allows the source node to better estimate the expected CAC behavior of nodes along candidate paths based upon additive link metrics advertised in PTSPs and the requested QoS and bandwidth of a connection request. The GCAC algorithm provides a good prediction of a

typical node-specific CAC algorithm. Individual nodes may optionally communicate the stringency of their own CAC using an optional complex GCAC calculation by advertising the Cell Rate Margin (CRM) and Variance Factor (VF) metrics.

The GCAC uses the additive metrics subject to the constraints described previously. Individual nodes (physical or logical) advertise the values of these parameters in terms of their internal structure and current connection status. The GCAC algorithm works best for CBR and VBR connections. GCAC for UBR connections simply determines whether nodes on the candidate path advertise support for UBR. For ABR connections, GCAC checks whether the link or node can support any additional ABR connections and also verifies that the Available Cell Rate (ACR) for the ABR traffic class for the node or link resource is greater than the Minimum Cell Rate specified by the connection. Using GCAC, the source node routing algorithm performs the following functions:

- ▼ Eliminates all nodes and links from the topology database that don't meet the end-to-end QoS or bandwidth constraints of the connection request.
- Applies advertised reachability information against the constrained topology to compute a set of least-cost paths to the destination, for example, using the Dijkstra algorithm.
- Removes candidate paths that fail additive link metrics, such as delay, and do not meet other constraints. If no paths are found, then the source node blocks the connection request. If the routing algorithm finds more than one path, the source node may choose one according to its policy, such as minimum delay, load balancing, or least administrative weight.
- ▲ The source node encodes the selected route to the destination in a Designated Transit List (DTL), which describes the complete hierarchical route to the destination, and it inserts this DTL into a SETUP signaling message sent to the first node identified in the DTL.

Note that when multiple links connect adjacent nodes, connections traverse only one physical link to preserve cell sequence integrity. For multiple links between adjacent nodes with otherwise equal attributes and metrics, implementations may perform load balancing.

Designated Transit Lists (DTLs)

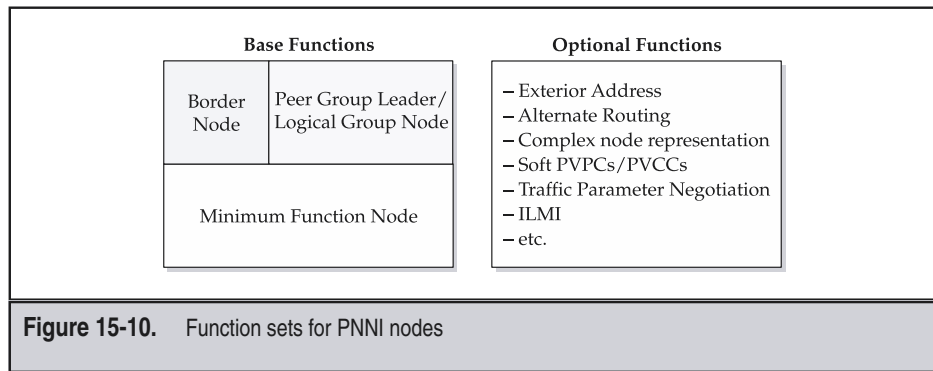
PNNI arranges DTLs in a stack in the SETUP message containing the path elements from the source to the destination. Each DTL is a sequence of abstract nodes along the path from source to destination. A node operates only on the topmost DTL. Figure 15-8 illustrates an example for a connection attempt from a source at address P.3.a to a destination at address E.1.1.b. Each DTL stack entry is a list of logical node IDs, along with a pointer indicating the next element, indicated by underlined type in the figure. Beginning at the source node P.3, the DTL articulates each node up to the border node P.4 in peer group P. The border node P.4 removes the top DTL from the stack and sets the pointer to the higher-level peer group, A, along the path to the destination E, as indicated by the bottom

setting up these connections using a combination of the IETF ATM Management Information Base (ATOMMIB) and the PNNI 1.1 ATM Forum SPVC MIB parameters at the origin node. The ATOMMIB, described in Chapter 27, keys on the source interface, VPI and VCI indices, which the SPVC MIB also uses to identify the source of either a point-to-point or a point-to-multipoint SPVCC (SPVPC). The ATOMMIB also defines the traffic descriptors for both directions of the connection. The SPVC MIB identifies the NSAP-based address of the destination, or target, interface and whether the destination may select any VPI/VCI, or if a particular target VPI/VCI must be used. The SPVC MIB specifies the operational status of the SPVCC (i.e., in progress, connected, or retries exhausted) diagnostics giving the cause for an SPVC release and a set of retry control parameters (e.g., interval, maximum number).

Once the SPVCC status is set to active or the SPVCC is restarted by the restart control, the network automatically attempts to establish the end-to-end connection as if it were an SVC. An additional information element in the PNNI signaling message instructs the destination switch to terminate signaling and not forward any signaling to the destination end system. Furthermore, if the source detects a failed connection, for example, due to an intermediate link failure, it must automatically periodically retry to reconnect the SPVCC/SPVPC. Hence, PNNI soft permanent virtual path and channel connections provide a standard means for a network operator to automatically provisioning and restore VPC and VCC PVCs. PNNI 1.1 enhances the SPVCC capability to provide support for FR over ATM PVCs or FR to ATM interworking SPVCs.

Minimum Interoperable PNNI 1.1 Subset

A complex protocol like PNNI must have a definition of a minimum interoperable subset; otherwise, manufacturers could choose which portions of PNNI they wish to implement, and the resulting networks would fail to meet the goal of multivendor operation. Toward



this end, the ATM Forum PNNI 1.1 specification defines base node subsets and options for three kinds of nodes, as illustrated in Figure 15-10. A minimal function node has only inside PNNI links and forms the mandatory basic function set. A border node also has outside links. A Peer Group Leader/Logical Group Node (PGL/LGN)-capable node participates in higher levels of the PNNI hierarchy. Furthermore, PNNI defines a set of optional functions, as shown in the figure. The PNNI 1.1 specification further details the mandatory and optional aspects for each node type.

BROADBAND INTERCARRIER INTERFACE (B-ICI)

The ATM Forum specified the Broadband Intercarrier Interface (B-ICI) for support of multiple services over PVCs as well as SVCs beginning in the early 1990s. B-ICI version 1.0 defined the PVC mode for the following services over interconnections between carrier networks: Cell Relay Service (CRS), Circuit Emulation Service (CES), Switched Multimegabit Data Service (SMDS), and Frame Relay Service (FRS). B-ICI 1.0 defined support for these services between the Local Exchange Carrier (LEC), Interexchange Carrier (IXC), Independent Local Exchange Carrier (ILEC), and other public ATM network providers. Depending upon regulatory and business arrangements, any carriers could assume any of these roles. B-ICI 1.0 contained information for physical, ATM, AAL, and higher layers as well as OAM functions in advance of ITU-T standards. The adoption of formal standards in most of these areas rendered this document obsolete.

B-ICI specifications 2.0 and 2.1 defined support for UNI 3.1 ATM SVCs in advance of ITU-T standards. The ATM Forum chose the ITU-T's B-ISDN Services User Part (BISUP) protocol stack as the basis for these signaling specifications. An addendum to the B-ICI 2.0 specification, called B-ICI 2.1, added three key features. A call correlation tag provides a means to identify a call across multiple carriers for billing purposes. The addendum also clarified support for Variable Bit Rate (VBR) connections. B-ICI 2.1 also defined support for all NSAP-based address formats in the connections between carriers. Adoption of these functions in formal standards has also made these specifications largely obsolete.

B-ISDN USER SERVICES PART (BISUP)

As introduced in Chapter 13, B-ISDN adapts the N-ISDN User Part (ISUP) at the NNI resulting in a protocol called B-ISUP. The ISUP protocol supports N-ISDN connection capabilities between carrier networks. The ITU-T specifies the B-ISDN ISDN User Part (BISUP) protocol for use at the NNI. The ITU-T BISUP specifications for signaling at the NNI match well with the corresponding ITU-T UNI Signaling standards as summarized in Table 15-1.

Capability Description	ITU-T UNI Standard	ITU-T NNI Standard
On-demand (switched) connections	Q.2931	Q.2761-4
N-ISDN signaling interworking	Q.2931	Q.2660
E.164 address support	Q.2931	Q.2761-4
NSAP-based address support	Q.2931	Q.2726
Root-initiated point-to-multipoint calls	Q.2971	Q.2722.1
Traffic parameter modification during active calls	Q.2963.1	Q.2725.1
Traffic parameter negotiation during call setup	Q.2962	Q.2725.2
Supplementary services	Q.2951	Q.2730
User-user signaling (UUS)	Q.2957	Q.2730

Table 15-1. Mapping of ITU-T UNI and NNI Signaling Standards

B-ICI'S REPLACEMENT: ATM INTER-NETWORK INTERFACE (AINI)

The ATM Forum's replacement for the B-ICI internetwork signaling protocol is called the ATM Inter-Network Interface (AINI) [AF AINI]. This interface provides the benefits of PNNI signaling (e.g., crankback and SPVCs) but does not support the exchange of PNNI routing information. As such, AINI has several potential applications. A principal driver is for interconnection of service provider networks, where there are compelling business reasons not to share topology and state information. Another potential use is between equipment within a single network that does not support PNNI. AINI could also be used for interconnection of PNNI networks over PNNI exterior routes, for example, between separate PNNI domains within a single network. In any event, routing information for AINI ports must be manually configured.

AINI is based on the PNNI signaling protocol, and it defines signaling interworking between PNNI- and B-ISUP-oriented networks, as well as interworking between PNNI networks. Although topology and routing information is not exchanged across an AINI, it does support crankback and alternate routing procedures to respond to failures during connection attempts, albeit with limited cause and diagnostics information. AINI was based upon a subset of the PNNI 1.0 specification (primarily section 6) and parts of other addenda in support of the capabilities summarized in Table 15-2. The AINI specification defines interworking functions (e.g., message mapping and procedures) between PNNI and BISUP networks.

Capability	AINI Specification
Point-to-point calls	M
Point-to-multipoint calls	O
Signaling individual QoS parameters	M(Note)
Crankback	M(Note)
Alternate routing as a result of crankback	O
Associated signaling	O
Negotiation of ATM traffic descriptors	O
Switched Virtual Path (VP) service	O
Soft PVPC and PVCC support	O
ABR Signaling for point-to-point calls	O
Generic Identifier Transport	O
Transport Frame Discard indication	O(Note)
A-INI/PNNI interworking	O
A-INI/B-ISUP interworking	O
Security signaling	O
Transported address stack	O
Generic Application Transport	O
Path and Connection Trace	O
Domain-based rerouting	O

Table 15-2. Mandatory (M) and Optional (O) AINI Capabilities

NOTE: Mandatory(M) for AINI/PNNI interworking, otherwise Optional(O)

As indicated in the table, only support for point-to-point calls with QoS parameters is mandatory. The crankback feature is also mandatory because it provides an alternate routing mechanism necessary in a source-routed network in response to changes in network condition, or a difference in actual resource availability versus that distributed by the routing protocol. For example, if CAC on the other side of an AINI interface blocks a connection attempt, crankback allows the originating PNNI network to try another AINI port to reach that network. What is different for crankback at an AINI versus a PNNI port is that the reachability and metrics for all AINI ports must be manually configured,

whereas PNNI does this automatically. Since the crankback function is not defined at a UNI, the AINI capability is important for resilient interconnection of networks that do not wish to share dynamic routing information.

At publication time, the ATM Forum was working on finalizing version 1.1 of AINI 1.1. This version adds support for UNI 4.1 features and removes support for B-ISUP interworking.

Extended PNNI and AINI Routing and Signaling Capabilities

The ATM Forum has also defined a number of optional PNNI- and AINI-related routing and signaling capabilities, which we summarize in this section.

Generic Application Transport (GAT) and Network Call Correlation Identifier (NCCI)

The ATM Forum has a goal for PNNI of being able to act as a network capable of supporting many applications. Instead of defining a new information element for each application, the protocol designers instead chose to implement a Generic Application Transport (GAT) information element that transports organization-specific information through a PNNI network to support nonstandardized features. There are many examples of things necessary to support ISUP or B-ISUP, as well as Frame Relay-specific parameters. Since the GAT IE can be present in most signaling messages, it can be used by applications to communicate information at various stages of an ATM call setup or release.

The Network call correlation identifier (NCCI) is a means to uniquely identify an ATM call within a network [AF NCCI]. The NCCI can be used by applications that need to correlate different management records generated for a call, for example, a billing application. An NCCI can be associated with a point-to-point or point-to-multipoint, switched or soft permanent virtual channel or path connection call. Note that the NCCI may be nonunique for point-to-multipoint calls if the call enters a network node on multiple branches.

Path and Connection Trace

Trace features are essential for the operation and maintenance of a network. Historically, this function was done by proprietary methods, which complicated management of a multivendor network. In response to this need, the ATM Forum defined a path and connection trace specification [AF TRACE] for PNNI networks. The features provide mechanisms to determine the sequence of logical nodes and links that a connection traverses. The path trace applies to new connections, while connection trace applies to an existing connection. As such, they have some unique functions, but they use some common concepts that we first describe.

The trace is initiated at a trace source node and terminates at a trace destination node. Note that a connection may have been initiated upstream from the trace source node, and conversely the connection may progress beyond the trace destination node. The underlying paradigm of the trace feature is gathering route information normally contained in

the (expanded) DTL of connection establishment messages exchanged along the path. The trace procedures use a Trace Transit List (TTL) information element (IE) to capture this information and return it to the trace source node. This procedure is necessary in a hierarchical network, because the completely expanded DTL may never be present in a single signaling message.

A path trace discovers the path that a connection takes and to help diagnose problems along the way, for example, an excessive setup failure rate. Path tracing uses normal connection establishment messages and procedures, but adds a TTL IE to the original SETUP message. A path trace can be initiated as a separate test connection or on selected connection attempts, as determined by administrative configuration. The first case is useful for troubleshooting, while the second is useful to confirm a user complaint. The path trace feature also requires new procedures in addition to the standard PNNI call and connection control procedures.

On the other hand, a connection trace collects information on existing connections. It requires two new messages, TRACE CONNECTION and TRACE CONNECTION ACKNOWLEDGE, which contain the TTL IE also used for path trace. One use of a connection trace would be to discover the actual route of an existing connection to troubleshoot a performance problem, for example, to determine the cause of excessive cell error rate reported by users. Another use could be to achieve diversely routed connections.

Domain-Based Rerouting (DBR)

The ATM Forum's domain-based rerouting feature [AF DBR] defines the means to re-route segments of an existing point-to-point connection in support of three real-world operational scenarios. DBR works with either a UNI or AINI at a source and destination node that may be anywhere along the connection path. When an AINI is involved, rerouting can occur in a coordinated manner across multiple networks. Rerouting may be done without involving the end systems using the connection, or it may be done explicitly in response to an end system request. Nodes must first negotiate support for DBR during call establishment before invoking these procedures. Rerouting services cannot be renegotiated once a call has already been established.

The first case is that of *hard rerouting* (also called break-before-make), which is triggered by a failure event that causes the existing connection segment to be released by the network back to the source node. The source node then tries to reestablish the connection to the destination node. Note that the ability of the connection to transfer cells may be impacted for many seconds using this procedure. When it is used for an SVC, the end user is not signaled regarding the release or a successful reestablishment attempt unless the re-route attempt of a connection attempt fails, which then results in SVC call release signaling to inform the user of the failure.

As discussed in the preceding chapter, there is a need for an ongoing reoptimization of long-duration connections in a constraint-based routed network, so that the path taken by connections can be maintained at a near optimal state. The other modes of DBR address this requirement. A source node can use *asymmetric soft rerouting* (called make-before-break) procedures for optimizing the path or other administrative needs in a

relatively nondisruptive manner. The source node accomplishes this by establishing a new connection segment to the destination while the initial connection segment is still active, and then switching the traffic onto the new connection segment, often resulting in loss of a negligible amount of traffic. The source node then releases the initial connection segment. *Symmetric soft rerouting* (also called make-before-break) procedures can be performed by either the source or destination node while an initial connection segment is still active. This has use in the application where a nondisruptive handoff must be made in a wireless networking environment.

The ATM Forum specification also defines the interactions necessary when these approaches are used together. For example, hard rerouting can be used to restore the call from failures, while soft rerouting can be used for optimization. In another example, when symmetric soft rerouting service is in use, the source and destination nodes could attempt to initiate a soft rerouting operation at the same time. In this case, the source node-initiated soft rerouting operation has priority over that initiated by the destination node.

Explicitly Routed Calls

There are circumstances in which a network operator needs to explicitly indicate the exact sequence of nodes and links for a connection, for example, in order to ensure that a pair of SPVCs has no node or link in common to achieve diversity. These may be logical or lowest-level physical nodes or links. Connections that are diverse in this way are not subject to a single point of failure and can be used with protection switching at the endpoints, as described in Chapter 28. PNNI 1.1 defines the explicitly routed call mechanism to support this need. An operator must explicitly specify the path, since PNNI routing typically has no knowledge of the underlying transmission infrastructure. Explicitly routed calls carry this information in the DTL stack of the SETUP message; however, all DTL information elements contain only lowest-level Node IDs and Port IDs. This may well include Node IDs from multiple peer groups.

Other Extensions

A number of other extensions to PNNI and in many cases AINI have also been defined in support of signaling security, loop detection, signaling congestion control, and call processing priority. We summarize each of these in the following. Signaling support [AF CS 116] for ATM Forum defined Security [AF SECURITY] adds a security services IE to call establishment and release messages. This IE allows inband exchange of messages between security agents, who can use additional information to decide whether to accept or reject a call or else provide useful diagnostics. Note that this security feature applies only to the signaling messages and not the data for the connection.

Congestion can occur with resources that implement the signaling protocol, such as a processor or an internal bus. As such, the ATM Forum specifies detailed guidelines for detecting congestion, as well as an IE for a crankback message that can be used to indicate congestion [AF CS 181]. For example, signaling congestion could occur due to a large number of connections that need to be rerouted from a particular source node. During this period, other connection attempts may use a procedure defined in the

specification to process congestion indication information to try another path to avoid the congested node.

In a manner similar to IISP, whenever manual routing is used, for example in AINI or B-ISUP, there is a potential for a routing loop of connection establishment messages to form. To address this issue, the ATM Forum defines a remaining hop count (RHC) IE [AF CS 176] and B-ISUP is defining a hop count IE to detect such loops. This hop count is similar in operation to the TTL field in an IP or MPLS packet header, except that each hop is a network. Specifically, the originating network sets the hop count to an initial value, and each time an AINI or B-ISUP interface is traversed, a node decrements its value. If the hop count reaches zero, then the call is released.

As is true in many aspects of life, some calls are more important than others. Recognizing this fact, the ATM Forum added support for the ITU-T Recommendation Q.2959 Call Processing Priority IE and procedures for use at the UNI, PNNI, or AINI [AF CS 182]. The IE, which is optional in SETUP and ADD PARTY messages, indicates the relative priority for access to call processing resources for that call. The UNI priority can be transported transparently across a PNNI or AINI for delivery to the destination. A separate priority IE can be used in PNNI or AINI signaling message. Once a call is established, the priority applies to the release phase of the call as well. This is useful in several important cases of network restoration. For example, if a SPVPC implements a VP that is used to carry many VC PVCs, then reestablishing the VP shared by many VCs should have higher priority than other per VC calls. Similarly, some VCs should have higher priority because they are used to reestablish signaling to build the PNNI hierarchy.

REVIEW

This chapter detailed the ATM signaling and routing protocols involved at the Network-Network Interface (NNI). We first summarized the simplest NNI protocol, namely the ATM Forum's Interswitch Signaling Protocol (IISP). The text then gave an in-depth walkthrough of the most sophisticated routing protocol developed to date: the ATM Forum's Private Network-Network Interface (PNNI). The text continued by summarizing historical considerations with signaling protocols employed between carrier networks at the Broadband Intercarrier Interface (B-ICI) using the BIDSN User Services Part (B-ISUP). We then gave an overview of the successor to B-ICI, called the ATM Internetwork Interface (AINI), which implements the signaling portion of PNNI, but not the routing. The chapter concluded with a summary of recently defined ATM Forum extensions to PNNI and AINI.



PART IV



ATM and MPLS Support for Networking Applications

Chapter 16 begins with an in-depth look at support for video and voice over ATM and MPLS. The subjects covered include interworking with voice, narrowband ISDN, circuit emulation, and video.

Next, Chapter 17 covers ATM and MPLS interworking with Frame Relay and other wide area networking data communication protocols, such as HDLC and SMDS. This chapter also describes the cost-effective, low-speed, frame-based ATM protocols. Chapter 18 then covers support for LAN protocols. In particular, we summarize the ATM Forum LAN Emulation (LANE) protocol and the multiprotocol bridging and routing over ATM encapsulation technique. This chapter also discusses some of the approaches being considered in the IETF to provide Ethernet over MPLS. Finally, Chapter 19 covers the important topic of ATM and MPLS support for IP in the context of a virtual private network (VPN). Regarding ATM, topics covered include classical IP over ATM, multiprotocol over ATM (MPOA), and IP multicast over ATM. The previous part covered the generic forwarding and control plane protocol support of MPLS for IP. This chapter summarizes the current state and direction of work in the IETF regarding an emerging set of provider-provisioned VPN (PPVPN) standards.

CHAPTER 16

Enabling Voice, TDM, and Video Over ATM and MPLS

This chapter covers the protocols supporting voice and video applications over ATM and MPLS. At the time of writing, more standards for ATM existed than did for MPLS, and therefore the amount of material in this chapter reflects this state of affairs. The following sections describe support for voice, circuit emulation, and video over ATM and MPLS. The story begins with an overview of packet voice network architectural and performance considerations, highlighting the principle application of trunking, which the text details for voice over ATM and MPLS. Next, we summarize the key characteristics of emulating synchronous TDM circuits over an asynchronous ATM or MPLS network. We then move on to explore the status of ATM and MPLS support for video, summarizing the state of digital video coding and delivery over packet-switched networks.

PACKET VOICE NETWORKING

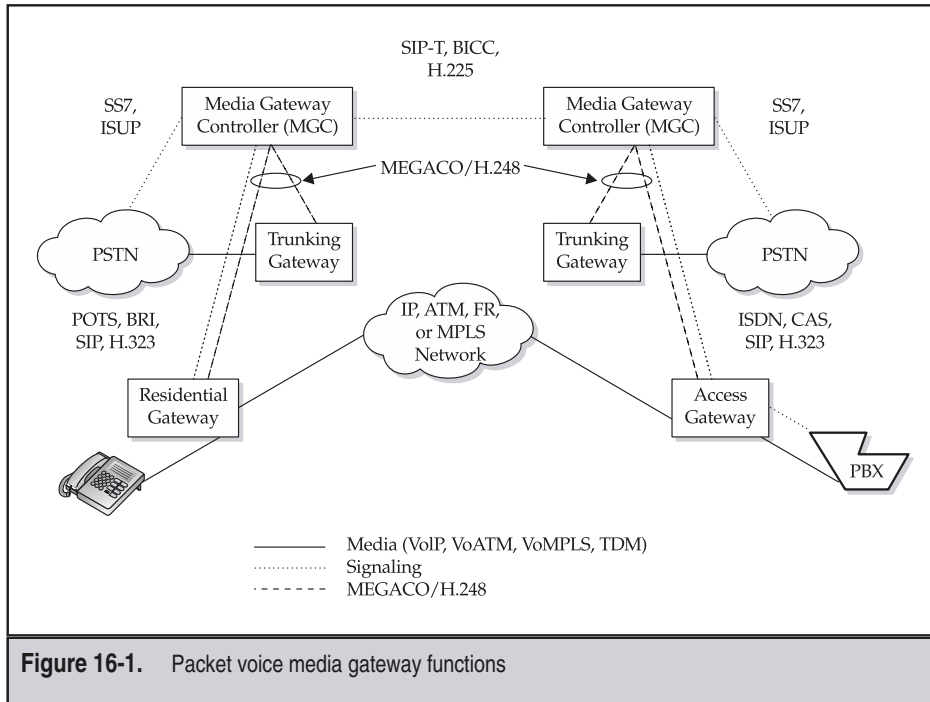
An important business driver for packet voice is cost reduction. Service providers and enterprises reduce costs by combining voice and data applications on expensive transmission facilities, not transmitting the packet samples corresponding to periods of silence, and reducing the transmission rate required for voice through advanced coding schemes. Combining voice and data effectively requires the network to provide some degree of bandwidth reservation, as well as controlled delay and loss for voice packets with respect to their data packet counterparts, as covered in Part 6. The silence suppression feature reduces the transmission requirement significantly, and modern coding schemes convey voice at bit rates much lower than the standard 64 Kbps Pulse Code Modulation (PCM) invented 50 years ago.

Voice over ATM (VoATM), voice over MPLS (VoMPLS), voice over Frame Relay (VoFR), and voice over IP (VoIP) are all forms of packet voice. Each was developed to meet a specific set of needs, and the focus of this book will be on VoATM and VoMPLS. However, where parallels or significant differences exist, we will also summarize important aspects of VoIP and VoFR. The Frame Relay Forum's FRF.1101 implementation agreement describes how Frame Relay supports the multiplexing of many voice conversations over a single DLCI. The ATM Forum's Voice and Telephony Over ATM (VTOA) Working Group generated a number of early standards, some of which the ITU-T later adopted, which is our focus in this chapter. For a historical view of the VTOA work, see Reference [McDysan 98] or the ATM Forum Web site. The MPLS Forum defined an application for trunking voice between points with a number of connections [VoMPLS 1.0]. Also called IP telephony, voice over the Internet Protocol (VoIP) is expected to be the principal protocol used for access and residential gateways, as well as trunking gateways that provide access to the PSTN. The principal application of VoATM and VoMPLS appears likely to be providing toll-quality trunking as either a replacement for or a means to continue growth for native PSTN services. Therefore, this chapter focuses on trunking of voice in a service provider environment using VoATM and VoMPLS. We also summarize the standards involved in using VoATM to provide ATM-based integrated voice and data access, for example, over a Digital Subscriber Line (DSL) facility.

First, let's take a look at the overall architecture and protocols involved in packet voice systems that emulate telephone calls and internetwork with the existing telephone systems.

General Network Architecture

Packet voice targets several deployment scenarios involving phone-to-phone communications, trunking between public switched telephone network (PSTN) switches, inter-PBX communication, and PBXs to branch office sites, as well as PC-to-phone connections. Figure 16-1 illustrates these representative deployment scenarios along with the protocols involved with them. The separation of the control plane from the user or switching plane is most mature in the area of telephony, as shown in the center of the figure, where a media gateway controller (MGC) can interface to one of several classes of gateways via either the media gateway control (MEGACO) protocol (see RFCs 2805 and 3015) or ITU-T Recommendation H.248. This separation provides a service provider the flexibility to choose the MGC vendor separately from the gateway vendor. Shown at the bottom of the figure are the other two classes of media gateways, called the residential and access gateways. Although, in theory, these gateways could be controlled via MEGACO/H.248, much of the focus is on separation of control and switching in service provider networks



to provide trunking between parts of the PSTN, or as a network gateway to provide a means for enterprise access and residential gateways to reach the PSTN. As shown in the center of the figure, the so-called bearer network could be IP, ATM, FR, or MPLS for the carriage of packet voice.

A number of control plane signaling protocols are involved in packet voice, as shown by the relationships represented as dashed lines in Figure 16-1. From the residential gateway, this includes plain old telephone service (POTS) and the N-ISDN Basic Rate Interface (BRI), as described in Chapter 6. From the enterprise access gateway, the signaling protocols include Primary Rate Interface (PRI) from N-ISDN and TDM-oriented channel-associated signaling (CAS), also as described in Chapter 6. Both the residential and enterprise gateways may also employ VoIP signaling protocols, principally, the Session Initial Protocol (SIP) [RFC 2543], with some extensions to support ATM [RFC 3108] and ITU-T Recommendation H.323. As shown in the upper-middle portion of Figure 16-1, the principal control plane protocols involved between MGCs are the Bearer Independent Call Control Protocol (BICC) defined in ITU-T Recommendation Q.1901 and the support of H.323-type devices as specified in H.225. Finally, the MGC has control protocols that work with PSTN switches, typically SS7 and its ISDN User Part (ISUP), that coordinate placement of voice and ISDN traffic on the bearer connection between the PSTN and the trunking gateway. Also, the SIP for Telephones (SIP-T) protocol extension was in draft state at the time of writing. As an example of these important control plane functions, we summarize the operation of BICC later in this chapter.

Media Gateway Functions

Let's take a look inside the packet media gateway to better understand its operation. Figure 16-2 illustrates the basic functions implemented by a media gateway: voice packetizing, telephony and signal conversion, and the packet network interface. The voice packetizing function converts from analog (or digital) speech to packets according to one of the voice coding standards described in the next section. Since some of these coding schemes can't convey DTMF tones, call progress tones, fax, or modem signals, a separate telephony signal processing conversion unit converts these information streams to a separate set of packets, as shown in the figure. The packet network interface function takes the voice and telephony signal packet streams and affixes packet headers for relay across the packet network to the destination gateway. This function also takes other packet data streams and multiplexes them with the voice and signaling packets in a prioritized manner to ensure voice quality.

Packet Voice Encoding Standards

Table 16-1 shows the major voice coding names, the defining ITU-T standard, the peak transmission bit rate, and the algorithmic encoding delay [Kostas 98]. Over a high-quality network, most users do not detect much difference between 64 Kbps PCM and the lower bit rate coded voice signals in subjective listening tests. Note that the lower bit rate standards carry human voice only—they do not carry DTMF tones, fax, or modem signals.

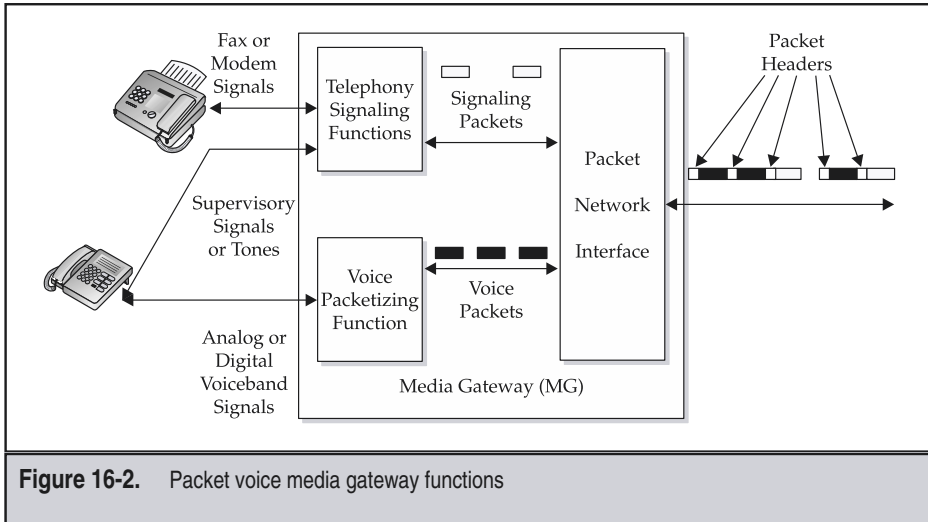


Figure 16-2. Packet voice media gateway functions

As we shall see, VoATM and VoMPLS define other means that are necessary to carry these types of signals, which are normally carried by traditional PCM voice networks. As shown in the table, the low bit rate G.723.1 coder uses less than 10 percent of the peak transmission bandwidth that the current telephone network does! However, this isn't the complete story, since packet-based voice transmission adds additional overhead, a topic that we cover in Part 8.

Acronym	Name	ITU-T Standard	Peak Bit Rate (Kbps)	Algorithmic Delay (ms)
PCM	Pulse Code Modulation	G.711	64	0.125
ADPCM	Adaptive Delta Pulse Code Modulation	G.726	16, 24, 32, 40	0.125
LD-CELP	Low Delay–Code Excited Linear Prediction	G.728	16	2.5
CS-ACELP	Conjugate Structure–Algebraic CELP	G.729	8	10
MP-MLQ	Multi Pulse–Maximum Likelihood Quantizer	G.723.1	5.3, 6	30

Table 16-1. Voice Coding Techniques, Standards, and Peak Bit Rates

Quality Considerations

The issue of quality is important in voice. Since packet voice inserts an algorithmic delay, as described in the previous section, and also encounters other delays when traversing a packet-switched network, delay is the key quality impairment. This occurs because PCM processes voice traffic on a sample-by-sample basis, while a low bit rate vocoder processes voice traffic on a frame-by-frame basis. The vocoder utilizes the correlation structure with nearby voice samples and frames to achieve information compression and a low bit rate. The algorithmic delay of a vocoder results from the encoding computation time, as well as the time waiting for samples to form frames. In summary, the contributions to overall round-trip delay include

- ▼ Voice encoding algorithmic delay (see Table 16-1).
- Packetization delay—the time to fill a packet with samples for transmission over the packet network. Typically, this is between 5 and 20 milliseconds.
- Time required to transmit the packets on the access lines, which is longer for slower-speed access lines.
- Switching and queuing delay encountered by voice packets traversing the packet network, which can be as low as a few milliseconds across an ATM or IP QoS network.
- Propagation delay, which depends on the distance between the communicating parties.
- ▲ Playback buffer delay, which is usually equal to a small multiple of the packetization delay.

All of these delays quickly add up and can create some additional requirements for support of packet voice. Placing calls to the legacy telephones presents a commonly encountered requirement. If any impedance mismatch exists between the 4 wires and the 2 wires in the analog hybrid transformer circuit leading to the legacy telephone, then an echo of the talker's signal at a reduced level returns. Called *talker echo* [Vickers 96], this phenomenon—in which the speaker hears a delayed version of his or her own voice—is quite annoying if the round-trip delay exceeds approximately 50 ms [ITU-T G.131]. If the total round-trip delay exceeds 50 milliseconds, then ITU-T Recommendation G.131 requires echo cancellation. Currently, most packet voice gateway functions implement echo cancellation.

Furthermore, if the one-way delay exceeds 100–200 ms, then the conversation is not very interactive. Some packet voice communication configurations can have one-way delays that exceed these values, particularly for PC-to-PC packet voice. This situation is similar to that of satellite communications where people may begin speaking at the same time. Once people get used to this effect, the connection is usable. As an additional benefit, people actually need to be more polite to let the other party finish speaking! As in the day when satellite communication was cheaper than terrestrial microwave transmission, cost-conscious users adapt to save money.

With improved voice codecs, lost or delayed voice packets are now the most significant source of voice fidelity impairments. Unlike data applications, voice applications cannot simply retransmit lost packets—it simply takes too long. Therefore, packet voice gateways account for lost packets in one of two ways. First, the packets contain sequence numbers so that the decoder can interpolate between lost samples, inserting interpolated data or noise using a technique called *packet loss concealment*. G.723.1 uses this method to operate acceptably when losing 10 percent of the packets while adding 40 ms of delay when compared with the G.729 coding and packetization delays [Kostas 98]. However, if the decoder does too much interpolation, then fidelity suffers. Another method involves the voice encoder inserting redundancy in the packets. However, this approach increases the bandwidth requirements and also increases delay. Of course, the best solution is to reduce loss to acceptable values. As studied in Part 5, ATM keeps absolute delay and delay variation to minimal values for CBR and rt-VBR service categories. MPLS is capable of using the IP Diffserv standard to support comparable levels of QoS, as described in Part 5. The Frame Relay Forum Implementation agreement FRF.11 adds the notion of prioritization for voice. The IP protocol adds the RSVP protocol to reserve bandwidth and QoS, as described in Chapter 8. Voice over IP also uses the Real Time Protocol (RTP) to sequence number voice packets and aid in the playback process, as described in Chapter 8.

Finally, the receiving gateway must handle variations in the packet interarrival times through a playback buffer dimensioned to handle the expected range of packet delay variation. Some contemporary Frame Relay and IP networks experience greater variations in packet interarrival times, necessitating larger playback buffers and increasing the end-to-end delay.

VOICE TRUNKING USING ATM AND MPLS

Figure 16-3 illustrates the trunking scenario examined in this section for VoATM and VoMPLS. As discussed earlier in this chapter, this is one of the most relevant cases involving the deployment of packet voice switching over ATM and MPLS. A voice over packet (VoPacket) switch is logically composed of a media gateway controller (MGC) and a media gateway (MG). If these are separate devices, then the MGC must use the H.248 or MEGACO protocol to control and communicate with the MG. Starting at the top of the figure, the MGC signals with telephone or N-ISDN switches using control plane protocols such as Signaling System 7 (SS7) or channel-associated signaling (CAS) (see Chapter 6), and also signals with other MGCs using a protocol like BICC [Q.1901] or SIP-T. Shown in the bottom of the figure, MGCs initiate signaling procedures on the MGs. In response, the MGs use either ATM or MPLS signaling via User-Network Interface (UNI) with ATM or MPLS switches to set up a VoATM or VoMPLS connection between the VoPacket switches. As covered in the previous part, ATM or MPLS switches use a Network to Network Interface signaling and routing protocol to establish an end-to-end ATM virtual channel (VC) or a pair of unidirectional label switched paths (LSPs) in each direction. The end result is an equivalent voice or TDM circuit connection between telephone/ISDN switches, as implemented by this set of VoATM or VoMPLS trunking protocols.

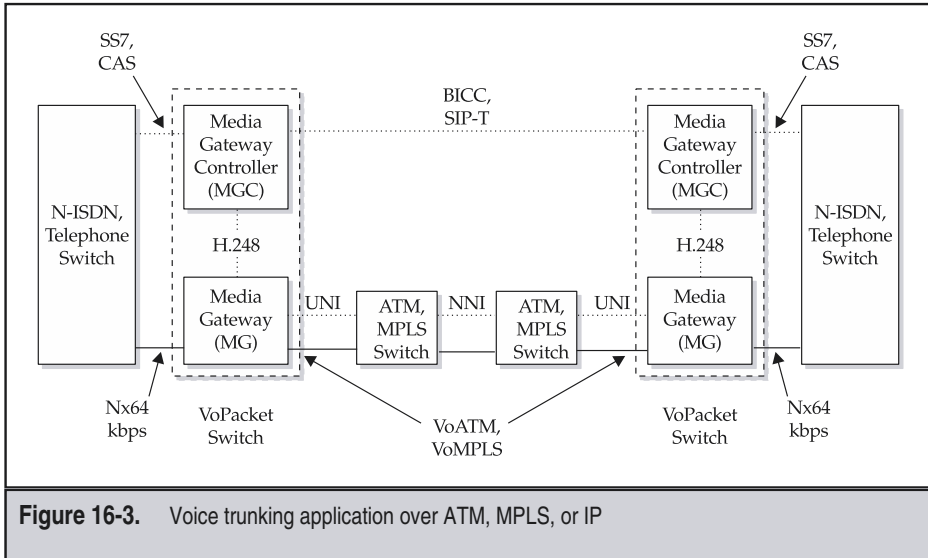


Figure 16-3. Voice trunking application over ATM, MPLS, or IP

Since this book focuses on ATM and MPLS, we cover the specifics of only VoATM and VoMPLS, but many of the concepts are similar for VoIP. Since, at the time of writing, only ATM control plane protocols for establishing a voice connection were standardized, we cover only the VoATM signaling case, as described in the next section. Next, we describe both the VoATM and VoMPLS data plane protocols.

Voice over ATM (VoATM) Trunking

This section covers important aspects of trunking voiceband connections over an ATM network. We begin with an overview of the signaling protocol exchange, then offer some details on the AAL2 VoATM encapsulation and operation, and conclude with some considerations involved in using ATM QoS and AAL types for specific types of voiceband connections.

VoATM Trunk Signaling

The call flow diagram of Figure 16-4 summarizes a number of the more significant signaling message exchanges that occur between the telephone switches, VoPacket switches, and ATM switches for the case of the VoATM trunking configuration in Figure 16-3. The high-level example covered here is backward establishment of a backbone network bearer connection, as defined in the BICC protocol [ITU Q.1901]. Recommendation Q.1901 uses generic functional terms, since it describes a general protocol architecture instead of the more implementation-specific examples described in the previous section. The following are the functional equivalents in terminology: an MGC implements the

Q.1901 call service function (CSF), while the MG implements the bearer control function (BCF). We use the implementation-oriented terminology in the following example with reference to Figure 16-4. The trunking proceeds from an ingress N-ISDN switch on the far left to an egress N-ISDN switch on the far right. Starting in the upper left-hand corner of the call flow, the ingress N-ISDN switch initiates call setup with an SS7 initial address message (IAM), which indicates the called telephone network party address A. The ingress VoPacket switch determines the egress VoPacket switch and sends a BICC IAM message over the SS7 network, adding its ATM address X to the telephone network address in the IAM message. This message also contains a request for backward connection, ATM bearer-specific parameters (e.g., traffic and QoS parameters), the AAL type, and other parameters needed to properly configure the VoATM connection.

Upon receipt of the BICC IAM, the egress VoPacket switch initiates an ATM SVC using the SETUP message to ATM address X to the egress ATM switch over the UNI, as shown in Figure 16-4. The egress ATM switch uses an NNI to complete the ATM SVC

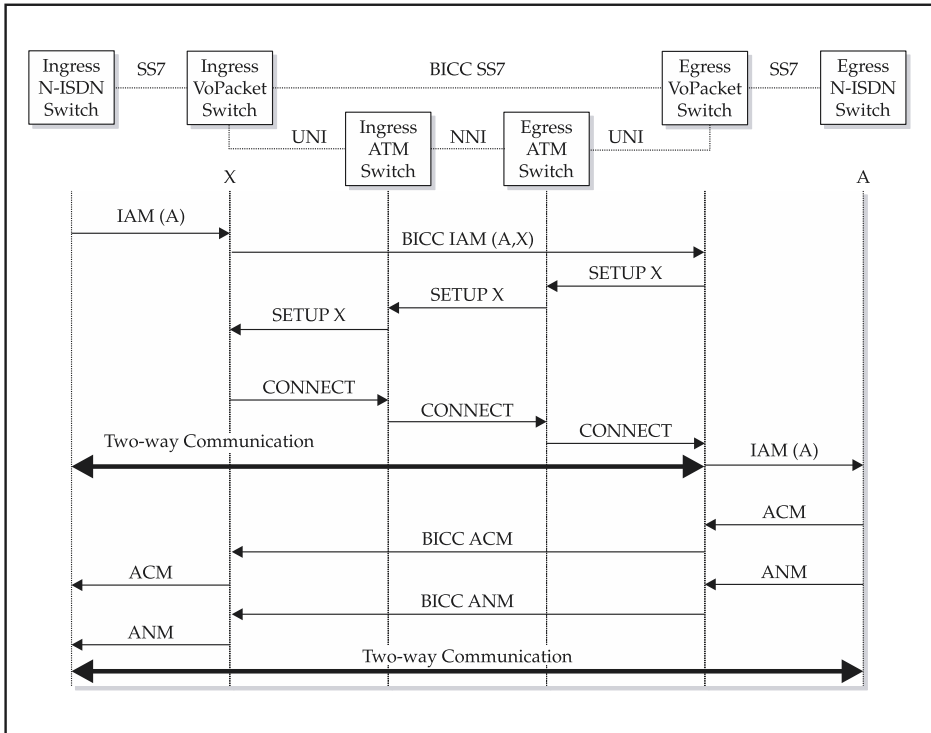


Figure 16-4. VoATM trunking control plane signaling

over the ATM network until the SETUP message arrives at the ingress ATM switch UNI associated with address X. In the usual handshake for an ATM SVC (see Chapter 13), the ingress VoPacket switch confirms connection establishment using the CONNECT message. Now, a two-way communication path is established from the ingress N-ISDN switch through to the egress VoPacket switch, as shown by the thick arrow in the middle of the figure. The motivation for this backward setup configuration is that, in some cases, the ingress N-ISDN/telephone switch may need to receive inband call progress tones or CAS indications from the egress N-ISDN/telephone switch in order to properly perform telephony-related call processing. Continuing with the example, the egress VoPacket switch progresses the call attempt to the egress N-ISDN switch using an SS7 IAM message containing the called telephony address A. Once the egress N-ISDN switch has reached the called party and the call is advancing, it communicates this fact using the address complete message (ACM). The egress VoPacket switch uses the BICC ACM to relay this information back to the ingress VoPacket switch, which, in turn, passes the SS7 ACM back to the ingress N-ISDN switch. Once the egress N-ISDN switch determines that the called party has answered, it returns the SS7 answer message (ANM). The VoPacket switches use the BICC ANM message to convey this fact back to the ingress N-ISDN switch, at which time there is an end-to-end, two-way voice communication path established, as shown at the bottom of Figure 16-4.

The preceding example covered only a single case of the procedures and options for BICC defined in ITU-T Recommendation Q.1901, which the interested reader should consult for further detail. Other cases include forward call setup along with use of the SS7 continuity (COT) tone test, negotiation of voice codec parameters, and reuse of idle bearer connections, as well as the messages and procedures involved in releasing a connection. Also, the ITU-T Q.1912 series of recommendations contain further detail on the encoding of the BICC messages and further descriptions of the parameters involved.

Some signaling experts regard this signaling example as unnecessarily complex, since it involves additional BICC and ATM signaling messages to advance a call between two telephone switches. The alternative that had been considered prior to BICC required translation of the N-ISDN signaling messages and parameters into equivalent B-ISDN signaling messages and parameters, as was envisioned in the general architecture described in ITU-T Recommendation I.580. However, after pursuing this approach for many years and seeing it significantly complicate B-ISDN/ATM signaling, the members of the ITU-T instead chose the approach described previously, which left existing N-ISDN/telephone switches unchanged and instead placed the additional signaling burden on the VoPacket switches. Since these switches are newer and have significantly faster (and less expensive) processors, this is a reasonable trade-off. Of course, if the bearer network is VoIP, then there is no signaling protocol to establish a bearer connection, and some simplification results here. However, without a bearer signaling protocol, it is difficult to perform admission control and ensure QoS, except on a statistical basis.

ATM AAL2 Narrowband SCS

Now we cover the AAL2 service-specific convergence sublayer (SCS) that supports packetized voiceband connections. ITU-T Recommendation I.366.2 defines an AAL2

SSCS specifically for narrowband services, which makes use of the AAL2 common part sublayer (CPS) summarized in Chapter 12, as detailed in ITU-T Recommendation I.363.2. This SSCS carries the content of a single narrowband call over an AAL type 2 connection, with a bearer capability of voice, voiceband data, circuit mode data, or frame mode data. Normally, the signaling protocol indicates the bearer capability, as described later in this section. The SSCS also defines secondary messaging for interleaving of other information over the connection, for example, dialed digits, channel-associated signaling bits, alarms, and loopback commands. See [ITU I.366.2] for more details or information on these other services. An AAL2 connection carries SSCS protocol data units as CPS packets using one of the two packet formats shown in Figure 16-5. The channel identifier (CID) specifies the particular AAL2 connection. The SSCS makes explicit use of the CPS user-to-user indication (UUI) field and implicit use of the Length Indicator (LI) in the CPS-Packet header, as shown in the figure. The AAL2 CPS header error control (HEC) field detects errors in the CID, UUI, and LI. The CPS packet payload is of variable length up to a maximum of 45 octets. Type 1 SSCS packets are unprotected and are used for transferring audio encoding and other information. Type 3 packet SSCS packets have a six-bit message type and a ten-bit cyclical redundancy check (CRC). The CRC-10 is identical to that used for operations, administration, and maintenance (OAM) ATM cells, as described in Part 7.

The value of the five-bit UUI field defines the AAL2 narrowband SSCS packet content, as indicated in Table 16-2. Type 1 packets with codepoints 0 through 15 support transfer of

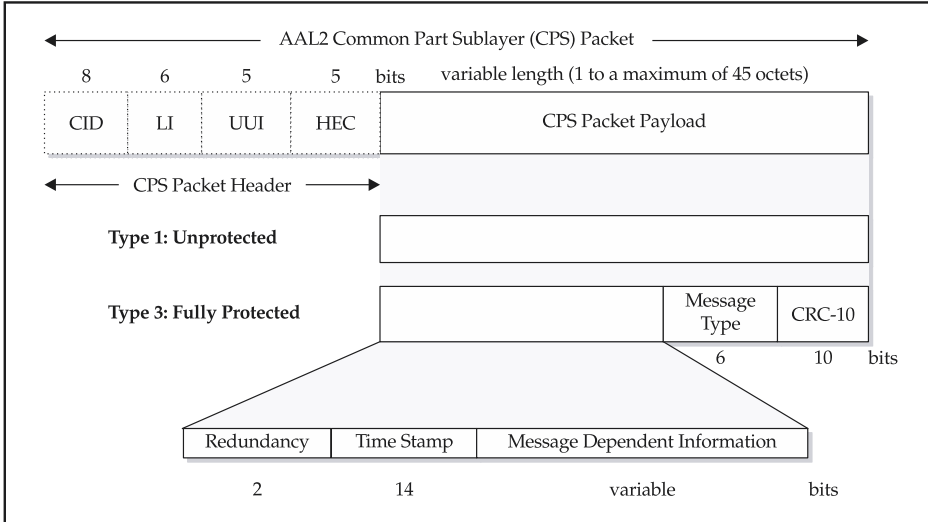


Figure 16-5. AAL2 narrowband SSCS packet payload types and CPS field usage

all of the ITU-T audio encoding formats described earlier in this section. This includes the capability to support silence suppression, which has the benefit of reducing the required transmission capacity and hence reducing cost for trunking or access loop emulation as described earlier. The circuit mode data service supports $N \times 64$ Kbps narrowband ISDN-type services. The protocol also provides a facsimile remodulation and demodulation service for encoding schemes that cannot directly support carriage of facsimile tones. In support of these services, the SSCS performs sequence numbering and absorption of delay variation introduced by the underlying network. An optional frame mode service, as detailed in ITU-T Recommendation I.366.1, provides an equivalent service to that of the AAL5 common part convergence sublayer (CPCS) described in Chapter 12.

Most type 3 packets share the additional structure shown at the bottom of Figure 16-5, which applies to dialed digits, channel-associated signaling bits, and facsimile demodulation control, as well as user state control packets. OAM functions, such as alarms, do not use this format nor the associated functions, and instead are patterned on ATM-layer OAM cells, as described in Part 7. The 14-bit time stamp field has units of one millisecond per increment. The transmitter-inserted value allows a receiver to accurately reproduce relative timing of events separated by a short interval, such as audio playback or dialed digits, as well as handle delay variation over a time scale comparable to the playback buffer. A full cycle of the time stamp counter takes slightly less than 16.4 seconds. Type 3 packets requiring error correction employ triply redundant transmission, as indicated by the two-bit redundancy field. Such packets are sent three times, with a fixed interval between transmissions, the interval depending upon the information stream. For example, it is 5 ms for dialed digits and channel-associated signaling bits. Each copy of a redundant

UI Codepoints	Packet Content and Function
0–15	Audio, circuit mode data, demodulated facsimile image data in a type 1 packet; may optionally be used for sequence numbering
16–23	Reserved for future assignment
24	Type 3 packets except OAM packets
25	Nonstandard extensions for vendor proprietary features
26	Framed mode data, final packet
27	Framed mode data, more to come
28–30	Reserved as specified in I.363.2 for AAL2 CPS
31	OAM packets

Table 16-2. AAL2 Narrowband SSCS UII Codepoint Determination of Packet Content

packet contains the same content, except the redundancy field, which is set to values 0, 1, and 2 for the three successive redundant transmissions. Redundant packets have the same time stamp, indicating the playout time, and hence the receiver can detect duplicates. A redundancy value of 3 indicates no use of triple redundancy, for example, when used as a long-term refresh of CAS bits. These messages may occur periodically, but at a much longer interval.

What follows the redundancy field and time stamp in a type 3 packet is message-dependent information, as determined by the message type field. Table 16-3 shows the information streams indicated by the six-bit message type field for type 3 packets, along

Information Stream	Message Type	Packet Content and Function
Dialed Digits	000010	Series of digits, digit type (DTMF, MF, R2), signal level
Channel-associated signaling (CAS)	000011	A, B, C, D CAS bits in the TDM frame (see Chapter 6)
Facsimile demodulation control	100000	ITU-T T.30 preamble
	100001	Echo Protection Tone (EPT) to disable echo cancelers
	100010	Training signal indication
	100011	Indication of local facsimile signal termination
	100100	ITU-T T.30 data
OAM	000000	Alarm Indication Signal (AIS), Remote Alarm (or Defect) Indication (RAI/RDI), Loopback command
User state control	000001	Control transition states for voice, voiceband data, circuit mode, and facsimile demodulation
Rate control	000100	Means for a receiver to request a rate from transmitter
SSCS synchronization	000101	Means for a transmitter to send reconfiguration requests

Table 16-3. AAL2 Narrowband SSSC Message Types and Packet Content

with a brief summary of the packet format, content, and purpose. Dialed digits and CAS messages support transfer of this information for audio encodings that cannot directly carry the tones, dial pulses, or CAS bits. Facsimile demodulation supports ITU-T Recommendation T.30 transfer of decoded facsimile data for audio encodings that cannot transfer the required tones. OAM provides a means to indicate alarms for network management as well as command loopbacks for diagnostic purposes. User state control allows an AAL2 transmitter to request that a receiver utilize a different, separately communicated configuration. Finally, rate control allows a receiver to request that a transmitter reduce the audio encoding rate, for example, if it encounters congestion.

The principal function of the SSCS is to transfer type 1 packets (UUI codepoints 0–15) from one information stream (i.e., audio, circuit mode data, or facsimile data). The information stream being transferred can be changed by type 3 messages, and the two directions of the connection can be set to different states. A profile is a mapping of both the UUI and Length fields from the AAL2 CPS header that tells the receiver of a type 1 audio packet how to interpret the packet content. A profile mapping defines the audio encoding standard and how the samples are placed into the payload, sequence numbering parameters, and the time interval between audio packet samples.

The AAL2 narrowband SSCS may use the low-order bits specified in the profile of the AAL2 CPS packet header UUI field as a sequence number for type 1 packets containing audio encoding information. The use of the sequence numbering is mandatory at the transmitter but optional at the receiver. At each time interval, the transmitter increments the sequence number according to the modulus up to the all-ones value and then wraps around to start again at zero. The entire UUI codepoint range 0–15 is used to encode $N \times 64$ Kbps circuit mode data with sequence numbers modulo 16. A large sequence number modulus aids in the detection of lost or late packets, and improves the integrity of the circuit mode data service. This sequence numbering makes AAL2 narrowband SSCS potentially applicable to link layer transport technologies other than ATM that may not always preserve packet sequence order, such as MPLS or IP.

VoATM and N-ISDN Relationships to AAL and QoS

The ITU-T and ATM Forum standards for transporting VoATM and N-ISDN traffic are

- ▼ Circuit emulation and full N-ISDN 64 Kbps compatibility over a Constant Bit Rate (CBR) ATM service category connection using AAL1
- ▲ Voice with silence suppression and other variable rate encodings over a real-time Variable Bit Rate (rt-VBR) ATM service category connection using AAL2

Each of these techniques has advantages and disadvantages. While CBR guarantees bandwidth and provides the best quality, it does not free up bandwidth during periods of voiceband inactivity. Use of the rt-VBR service category offers a more cost-effective alternative for human voice, since the ATM network doesn't use any bandwidth during periods of silence. The unused bandwidth during these inactive intervals is then available to other, lower-priority ATM service categories.

Most of the audio encoding standards support constant rate transmission and can operate over AAL1 using the CBR service category. Fax and modem modulation are best supported by the CBR service category, or by transcoding to the native bit rate using methods employed by the AAL2 narrowband SSCS described earlier.

Support for variable rate, packetized voice and other audio encodings normally works well using the rt-VBR ATM service category, which improves efficiency by not transmitting cells during periods of silence. The protocol converting voice to ATM cells detects silent periods and doesn't generate any cells during these intervals. An integrated voice/data multiplexer sends lower-priority data traffic during these silent intervals. Typically, the gaps in normal human conversation last several seconds. Telephone engineers have a great deal of experience on how to make silence suppression yield high fidelity speech transmission from work on undersea cable and satellite communication systems. These gaps in human speech constitute over 60 percent of the average conversation, allowing for approximately a 50 percent bandwidth savings when using rt-VBR compared with CBR. Chapter 25 provides more details on the statistical multiplexing of voice.

Table 16-4 compares the advantages and disadvantages of placing voice traffic over AAL1 on a CBR VCC versus AAL2 on a rt-VBR VCC. With the statistical multiplexing inherent in rt-VBR, there is a possibility of loss. Loss of voice signal due to cell loss may sound like a click, or, occasionally, it may render a single syllable (or even an entire word) unrecognizable. In summary, rt-VBR is more efficient than CBR, but at the expense of a small degradation in quality.

Voice over MPLS (VoMPLS) Trunking

The first document produced by the MPLS Forum is a voice over MPLS bearer transport implementation agreement [VoMPLS 1.0], where, as described earlier, bearer transport is ITU-T parlance for trunking. The protocol defines the encapsulation of packet voice directly over an MPLS LSP in a highly efficient manner. Patterned after a document produced by the Frame Relay Forum [FRF 11.1], the protocol has many similarities to the AAL2 narrowband SSCS described earlier [I.366.2]. Both the MPLS and FR Forum documents are implementation agreements in the sense that they define a subset of

Attribute	AAL 1 and CBR	AAL2 and rt-VBR
Cost of VC	Highest	Lower
Bandwidth Usage	Less efficient	More efficient
Voice Quality	Effectively lossless	Possibility of loss

Table 16-4. Comparison of ATM CBR to rt-VBR Service Category Support for Voice

the functions in Recommendation I.366.2, as well as certain options and specifications that are clarified in the interest of interoperability. Okay, let's take a look inside the VoMPLS protocol.

Figure 16-6 illustrates how VoMPLS audio-encoded primary subframes are multiplexed into a single packet on an MPLS label switched path (LSP). The mandatory (M) top-level label (called the outer label in [VoMPLS 1.0]) indicates the destination VoMPLS trunk switch. An optional (O) stacked label (called an inner label) can be used to create multiple logical trunk groups between VoMPLS trunk switches to increase the number of connections, or to support multiple signaling sessions. What follows is one or more VoMPLS *primary subframes*, the total number being limited by the maximum transmission unit (MTU) size of the MPLS LSP. Each primary subframe has further structure, as shown at the bottom of Figure 16-1.

Values of the 8-bit channel identifier (CID) in the range 0–247 identify VoMPLS user channels, and value 248 is used for peer-to-peer management; while value 249 is reserved for signaling, with values 250–255 reserved for future standardization. This encoding essentially combines the CID and part of the UUI functions of the AAL2 narrowband SSCS. The next field is the payload type, for which values 0–192 are used for voice and audio encoded information, values 224–255 are used for control payloads (as described in the text that follows), and values 193–233 are reserved. This is similar to the AAL2 narrowband SSCS message type along with the UUI. Next is an 8-bit counter, which operates effectively as a sequence numbering function. The 6-bit length field indicates the number of 32-bit words in the variable-length encoded voice or audio information portion of the subframe. The protocol requires alignment on a 32-bit word boundary, and, therefore, a PAD field of between 0 and three octets completes the subframe. The 2-bit PAD length (PDL) field indicates the length of the PAD so that the receiver can remove it.

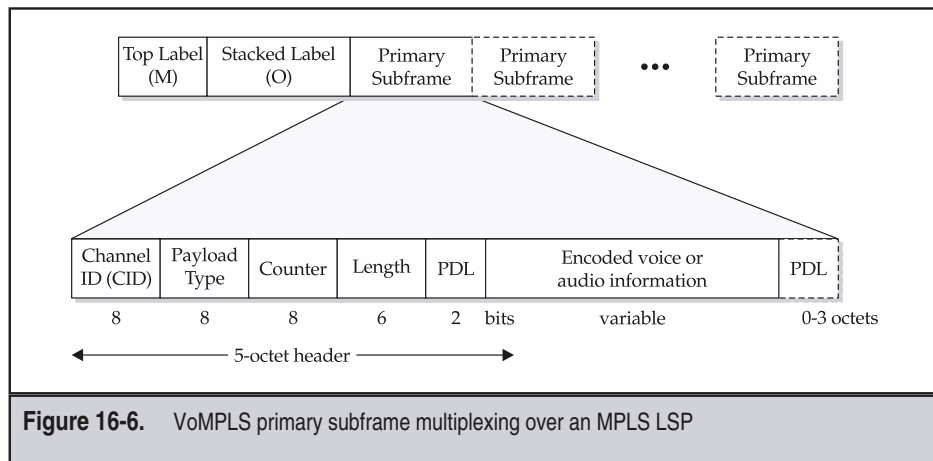


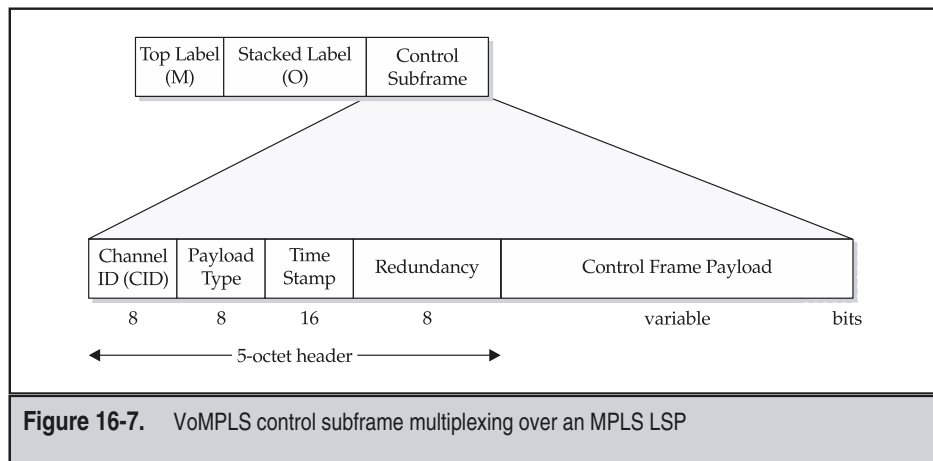
Figure 16-6. VoMPLS primary subframe multiplexing over an MPLS LSP

Figure 16-7 illustrates how a single VoMPLS control subframe is sent in a packet over an MPLS LSP. The CID and payload type fields are identical to that of a primary subframe, as described previously. Version 1.0 of the MPLS Forum implementation agreement only defines support for dialed digit and channel-associated signaling as payload type 240 and 241, respectively. The 16-bit time stamp and redundancy fields are analogous to those used in the AAL2 narrowband SSCS. That is, the time stamp field helps the receiver play back digit and CAS information with the correct temporal spacing, while the redundancy field helps ensure that this critical signaling information is reliably delivered.

Although a VoMPLS primary subframe header is four octets as compared with the three-octet AAL2 narrowband SSCS type 1 packet, the fact that multiple VoMPLS primary subframes can share the same MPLS LSP topmost label should make for relatively efficient trunking. Also, the MPLS Forum implementation agreement supports a very simple set of signaling protocol functions, and avoids the additional complexity of facsimile and N×64 Kbps support defined in the AAL2 narrowband SSCS. Of course, for high-quality VoMPLS trunking, the MPLS LSP should either be traffic engineered for high performance, use a separate LSP for each QoS class (i.e., L-LSP), or else support IP QoS using the EXP bits in the MPLS header (i.e., E-LSP), as described in Chapter 21.

BROADBAND LOCAL LOOP EMULATION USING AAL2

The access network is often cited as the principal bottleneck for broadband services. As described in Chapter 11, the Digital Subscriber Line (DSL) transmission method running over existing twisted pairs will play an important role in connecting small business sites and consumers to a range of network services. An ATM Forum specification defines a



loop emulation service that employs the AAL2 narrowband SSCS [AF VMOA 145] to meet this market need by providing an efficient mechanism to carry voice, voice-band data, and facsimile, as well as frame- and circuit-mode traffic. As shown in Figure 16-8, an ATM access network supports the means for voice-related services to share a broadband DSL access line with Internet data, for example, using IP over AAL5. The ATM Forum specification defines the use of AAL2 over an ATM virtual connection (a PVC, an SPVC, or an SVC) that is separate from the connection used to access the Internet, as shown in the figure. This VC supports the voice- and facsimile-related services using either common channel or channel-associated signaling with the telephony device located at the customer premises equipment (CPE). A higher-performance QoS class would typically be assigned to the AAL2 VC for the voice-related services than the AAL5 VC assigned to support access to the Internet. As discussed in Parts 5 and 6, this is precisely the type of application where connection-oriented QoS (like that provided by ATM) applies, since the voice- and data-related traffic are on separate virtual connections.

The specification describes the voice-related functions performed at the customer premises interworking function (CP-IWF) and central office IWF (CO-IWF) in terms of the interworking with traditional analog and digital telephony and the conversion to AAL2 narrowband SSCS—specific messages and protocols. Therefore, by examining the operation of each of these components in further detail, we can understand a great deal about the service. Let's begin with the CP-IWF, since it is simpler.

As shown in the left-hand side of Figure 16-9, the physical interface of the CP-IWF maps signaling from time slots or analog telephony signals to AAL2 SSCS channels. The

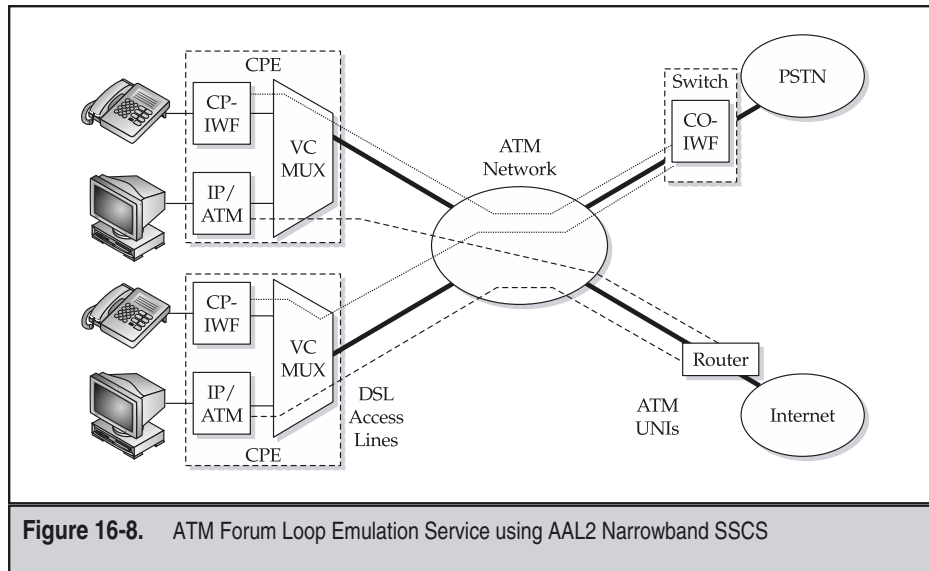


Figure 16-8. ATM Forum Loop Emulation Service using AAL2 Narrowband SSCS

SSCS user function processes individual streams from the physical telephony interface. For voice calls, it is a codec with functions indicated in the profile associated with the AAL2 connection, such as coding algorithm, bit rate, and silence suppression. For facsimile, the SSCS user function could be the facsimile demodulation/remodulation capability. The SDUs from the SSCS user are passed to an AAL2 SSCS (either I.366.1 or I.366.2), with the resulting AAL2 packets transferred to the AAL2 CPS function for multiplexing into ATM cells for transfer. The signaling interworking function converts between analog, CAS, or CCS signaling on the telephony side of the CP-IWF into the CAS or CCS signaling protocols used in the AAL2 narrowband SSCS, which are then carried by the AAL2 CPS over an ATM VCC back to the CO-IWF. When a network uses SVCs to support loop emulation, the signaling interworking function interacts with VCC management function, which uses the signaling AAL (SAAL) to initiate the establishment and release of SVCs between the CP-IWF and the CO-IWF. The procedures for ATM cells arriving from the network destined for the CP-IWF telephony physical interface are the mirror image of those described previously, which we describe for the CO-IWF next.

The functions in the CO-IWF for processing voice and facsimile are similar to those in the CP-IWF, but typically serve a much larger number of equivalent connections. Starting from the lower left-hand corner of Figure 16-10, cells from ATM VCCs connected to many CP-IWF devices deliver AAL2 CPS packets to the AAL2 SSCS function. The extracted encoding packets are then passed to an SSCS user function, which is either a voice codec with a profile specific to the CP-IWF connection, or the facsimile demodulation/remodulation capability. As defined in the ATM Forum specification, a CO-IWF may operate in one of two modes. In nonconcentrated mode, the timeslots for the TDM connections between the

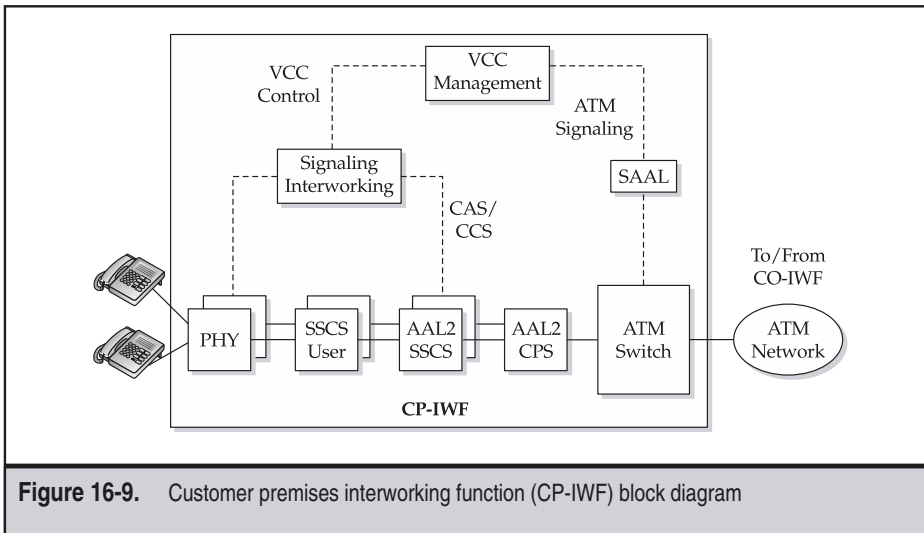


Figure 16-9. Customer premises interworking function (CP-IWF) block diagram

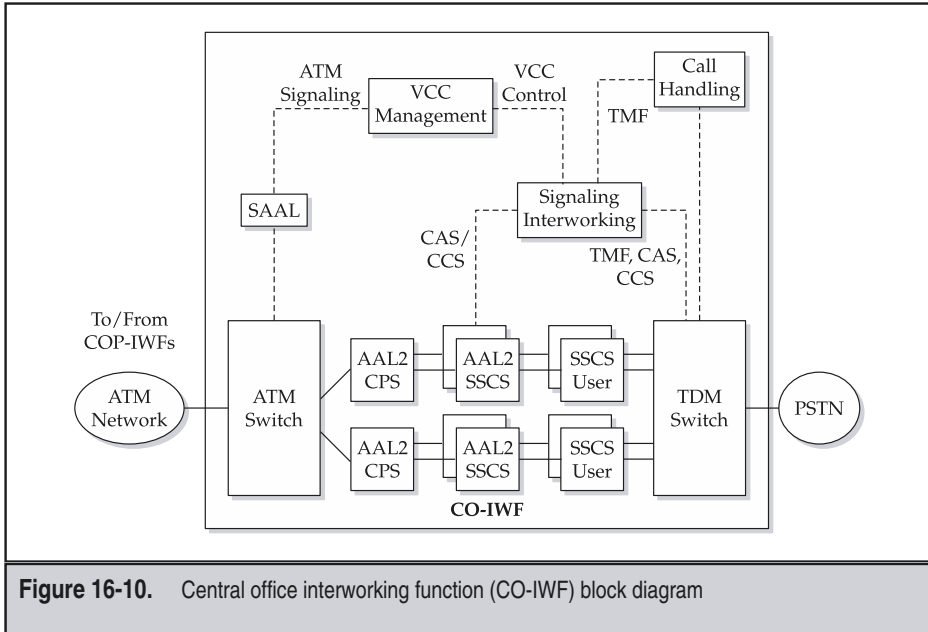


Figure 16-10. Central office interworking function (CO-IWF) block diagram

CO-IWF and the telephone switch are statically allocated to a specific AAL2 SSSC channel ID. In concentrated mode, the CO-IWF dynamically allocates timeslots to AAL2 channels in response to CAS or CCS signaling received from a CP-IWF. For nonconcentrated access mode, the call handling function is null, with signaling and bearer information passed according to the fixed mapping of time slots to AAL2 channels.

Now, let's look at control plane signaling in the CO-IWF with reference to Figure 16-10. In networks where SVCs implement connections between the CP and CO IWFs, the VCC management function uses ATM signaling information to determine how to interact with the signaling interworking function to initiate the establishment and release of VCCs. The signaling interworking function converts between the CAS or CCS signaling on the AAL2 narrowband SSSC and the telephony side of the CO-IWF. This signaling includes end-to-end signaling in CAS or CCS format to and from the CP-IWF, as well as a local timeslot management function (TMF) in the concentrated mode. The signaling interworking passes TMF signaling to the call handling function, which controls the TDM switch to set up and tear down connections between specific telephony timeslots and AAL2 channels. Finally, on the right-hand side of the figure, the signaling interworking function inserts and extracts CCS or CAS signaling to and from the telephony side of the CO-IWF associated with the individual 64 Kbps channels via the TDM switch.

Although this technology is well suited to the efficient support of voice in the upstream of an asymmetric DSL link, the efficiency for support of IP over AAL5 in the downstream

direction is not as good as with a frame-based protocol, as analyzed in Part 8. Furthermore, there are other impediments to the use of this technology, an important one being that of regulation. In some regulatory environments, telephony and the Internet are considered separately, with telephony often subject to regulation, but Internet access often unregulated. For example, voice may be regulated such that an incumbent service provider would be required to resell the voice access to competitors, which can be a disincentive for an incumbent provider to deploy a technology such as this.

CIRCUIT EMULATION USING ATM AND MPLS

Closely related to the support for voice, but in a more traditional manner, is the support for the direct emulation of TDM circuits. This section covers the standardized support by ATM for circuit emulation, and summarizes some of the current direction of emerging MPLS-based techniques to emulate some attributes of TDM circuits using a specific set of protocols operating over a pseudo-wire. Chapter 6 described the international digital multiplex hierarchy, of which the latest set of standards is called SONET/SDH. These standards define very specific means of handling error conditions, sending alarms, supporting management system communication, and collecting performance statistics. TDM circuits also have some stringent performance requirements, including a low bit error rate (BER), a high percentage of error free seconds (EFS), expectations of nearly continuous availability, and rapid restoration. Some of these performance attributes are challenging to meet when emulating a circuit over a packet-switched network (PSN), since loss of a single packet creates a burst of errors, which makes achieving a low BER or EFS challenging. Furthermore, ATM and MPLS signaling and routing protocols detect failures and restore service in a time frame that is on the order of seconds versus tens of milliseconds in SONET/SDH ring restoration systems, which reduces overall availability and decreases EFS.

AAL1-Based Circuit Emulation Service (CES)

The ATM Forum specification for Circuit Emulation Service (CES) [AF CES 2.0] defines the means for ATM-based networks to employ AAL1 [ITU I.363.1] to emulate, or simulate, synchronous TDM circuits over the asynchronous infrastructure of ATM networks. CES defines support for two types of emulated circuits:

- ▼ Unstructured DS1/E1/J2 (1.544/2.048/6.312 Mbps)
- ▲ Structured DS1/E1/J2 supporting $N \times 64$ Kbps (i.e., fractional DS1/E1/J2)

Chapter 12 described AAL1 and gave examples for both of these services at the detailed protocol level. This section provides a more deployment-oriented perspective of AAL1-based circuit emulation. Figure 16-11 illustrates the generic CES reference model. On each end, TDM equipment such as a PBX, a multiplexer, a switch, or a CSU/DSU (not shown) connects to the CES Interworking Function (IWF) via a standard TDM physical

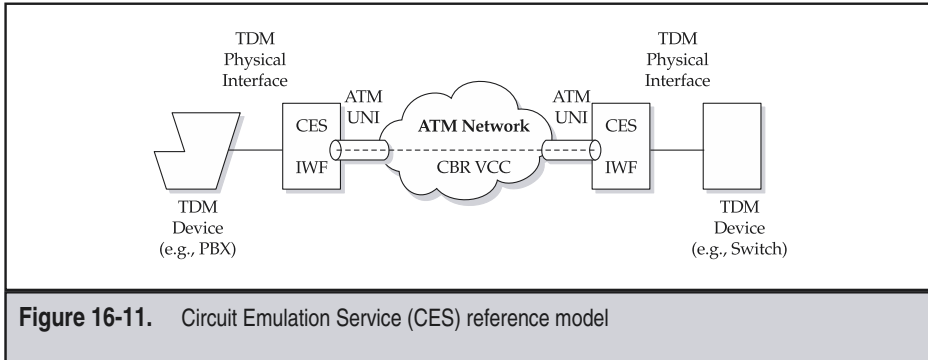


Figure 16-11. Circuit Emulation Service (CES) reference model

interface connector and protocol. The IWF implements the AAL1 SAR and CS sublayer functions defined in Chapter 12. The CES specification also defines a logical circuit emulation capability, which has no physical interface. This technique enables an efficient electronic interface to a TDM cross-connect function inside a piece of equipment. CES mandates use of the Constant Bit Rate (CBR) ATM service category and associated quality of service for the Virtual Channel Connection (VCC) that interconnects CES IWFs. This choice means that the playback buffers in the IWF underflow or overflow infrequently. The specification also details handling for error cases, such as lost cells, buffer underflow and overflow, and TDM alarm processing and performance monitoring. CES also defines the necessary parameters to set up ATM SVCs to support circuit emulation. These include the ATM traffic descriptor, QoS parameter, AAL1 parameters, and broadband low-layer information. The ATM Forum's specification contains a MIB and identifies several other MIBs that have been defined to support circuit emulation. It also contains an impairment analysis that maps a typical DS1 TDM circuit's 10^{-6} BER and 99.5 percent EFS performance objectives into ATM performance parameters that would yield equivalent performance.

Unstructured Mode Circuit Emulation

Figure 16-12 illustrates an example application and key functions of the unstructured mode CES Interworking Function (IWF). As described in Chapter 12, AAL1 unstructured data transfer (UDT) mode operates in either synchronous mode, where the IWF provides timing to the TDM equipment, or asynchronous mode, where the IWF accepts timing from attached equipment and transfers this timing to the destination IWF. Asynchronous timing transfer uses either the Synchronous Residual Time Stamp (SRTS) method or adaptive clock recovery. Timing transfer is critical for many legacy TDM networks, specifically TDM multiplexer networks. The unstructured service provides a clear channel pipe at a bit rate of 1.544 Mbps for DS1, 2.048 Mbps for an E1, 6.312 Mbps for J2, 44.736 Mbps for a DS3, or 34.368 Mbps for an E3. This means that the CES IWF supports bit streams with nonstandard framing, such as that used by some video codecs and encryptors, in addition to standard, framed DS1, E1, and J2 signals used by multiplexers and PBXs. One disadvantage of

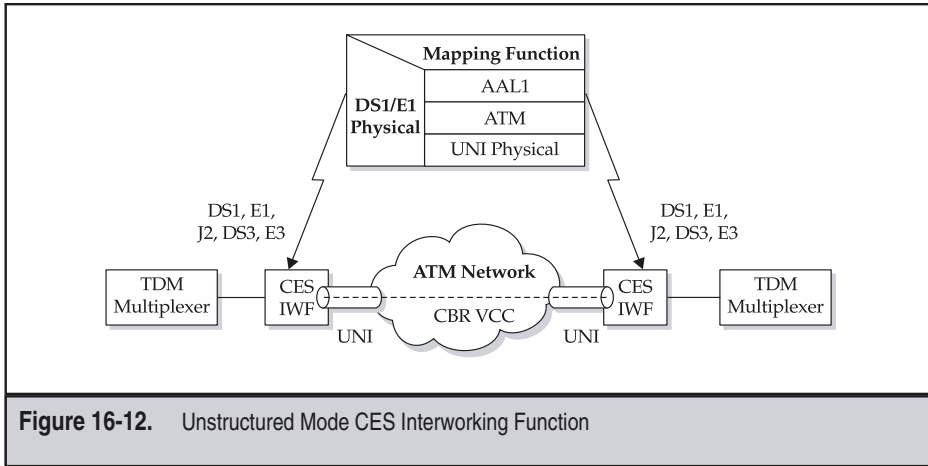


Figure 16-12. Unstructured Mode CES Interworking Function

devices that don't use standard framing is that the CES IWF cannot support standard DS1, E1, and J2 performance monitoring or alarm handling.

ATM Forum specification VTOA-119 defines another form of unstructured service support for low-speed serial interfaces (e.g., EIA-449 and V.35) found on some older DTE and DCE data communications equipment, as described in Chapter 5. The data rates involved range from 75 to 2048 Kbps but can be higher. This service can be useful to connect telemetry devices, control system signals, or special-purpose devices, like encryptors.

Structured Mode Circuit Emulation

Figure 16-13 illustrates an example application and key functions of the structured CES IWF, which always provides timing to the connected TDM equipment. As described in Chapter 13, an accurate clock must be provided at each interworking function (IWF) interface for proper AAL1 structured data transfer (SDT) mode operation [ITU I.366.1]. In turn, the CES IWF usually then provides this clock to attached devices, such as a PBX or a telephone switch, as shown in the figure. The structured capability supports combinations of $N \times 64$ Kbps bearer channels, where $1 \leq N \leq 24$ for a DS1, $1 \leq N \leq 31$ for an E1, and $1 \leq N \leq 96$ for a J2. These services are often called fractional DS1, E1, or J2. Optionally, several emulated groups of $N \times 64$ Kbps circuits may occupy the same DS1, E1, J2, or logical interface, as illustrated in the figure by the two instances of AAL1, one for each CBR VCC in the example. In this example, the CES function performs a mapping of the 64 Kbps time slots from the DS1/E1/J2 TDM transmission pipe into separate AAL1 Segmentation and Reassembly (SAR) functions. The structured service must maintain 125 μ sec frame integrity. That is, timeslots received in the same frame at the CES IWF source must be delivered at the destination in the same frame and in order.

Structured CES also specifies support for channel-associated signaling (CAS), commonly used by PBXs to indicate off-hook and on-hook conditions. Since the structured

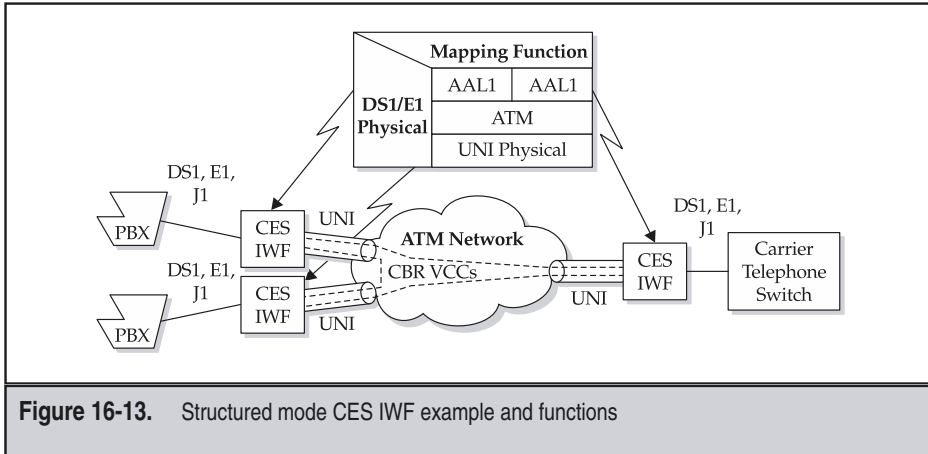


Figure 16-13. Structured mode CES IWF example and functions

mapping of the individual time slots does not convey TDM framing information end-to-end, the CAS information is encoded and transported separately, requiring additional overhead. The specification also requires that the CES IWF report alarms received from any of the connected devices.

Circuit Emulation over MPLS

Unlike the ATM Forum specifications, the IETF Pseudo Wire Emulation Edge-to-Edge (PWE3) working group does not make achieving performance levels comparable to that of a TDM circuit a firm requirement. This means that the target circuit-oriented application should be willing to accept BER, EFS, and/or availability performance less than that which would typically be delivered by a TDM circuit. This occurs because of impairments caused by the underlying IP or MPLS network, such as outages of varying duration due to software, circuit, or hardware failures, periodic reoptimization, packet loss, excessive delay variation, and restoration times that may be significantly greater than those achieved by TDM systems. ATM networks had some of the same challenges to achieve performance comparable to TDM. In fact, several operational networks have achieved the performance needed to meet the ATM Forum-specified goals summarized earlier. We expect that a well-run, traffic engineered, probably QoS-enabled MPLS or IP network could achieve circuit emulation performance comparable to that on an ATM network; however, a TDM circuit would likely still have superior performance.

At the time of writing, there were several proposals on how best to support an emulated circuit over an MPLS or an IP tunnel. One suggestion was to use AAL1, AAL2, and the AAL2 SSCS protocols directly, since none of them actually require ATM, and they could instead be carried over an MPLS or IP tunnel with sufficiently low loss and delay variation. Another proposal is to use the time stamp and sequence number properties of

the Real-Time Transfer Protocol (RTP) over the User Datagram Protocol (UDP), as described in Chapter 8. This should work well, since critical components of AAL1 and AAL2 are timing transfer and sequencing. However, some capabilities are missing from RTP/UDP, such as CAS, along with TDM alarm support and performance monitoring. Furthermore, RTP/UDP requires 20 bytes of overhead per packet, making for a difficult trade-off of the efficiency of a long packet versus the loss of significant interval of the signal if a packet is lost. Finally, there were also several proposals that borrowed parts from other protocols, like RTP and the ATM adaptation layers, or simply proposed new ways of emulating a synchronous TDM circuit over an asynchronous network. An objective of some of these proposals was to emulate a circuit in a highly efficient manner.

VIDEO OVER ATM AND PACKET NETWORKS

This section covers an overview of video coding standards along with a summary of video over ATM and packet-switched networks. At the time of writing, there was no work in progress to specifically support video over MPLS. However, as described in Chapter 12, there is an ATM Forum specification for carrying ATM or AAL5 over MPLS, and therefore any video over ATM service that uses these protocols could operate over MPLS. Also, a number of video over IP implementations can operate over MPLS, ATM, Ethernet, or a variety of other link layer networks, as long as the packet transfer performance met the levels necessary for the video application.

Commonly Used Video Coding Standards

Much of the videoconferencing today is TDM or IP based. However, the guaranteed capacity of ATM networks along with controlled latency and delay variation is used in a number of high-performance video applications.

Table 16-5 summarizes the encoded bit rate requirements and compression ratio relative to broadcast quality video for the set of compressed video encoding standards discussed in this section. The reference for uncompressed broadcast quality video is defined by the North American National Television Standards Committee (NTSC). ITU-T Recommendation H.261 defines encoding for two-way audio and video at rates of $p \times 64$ Kbps, where $1 \leq p \leq 30$. A common use of this standard is for small-screen N-ISDN-based videoconferencing by terminals with characteristics defined in ITU-T Recommendation H.320. The quality of this encoding at 1.5 Mbps is comparable to that of an analog VHS VCR. Although ITU-T Recommendation H.321 defines the means for a native ATM-based terminal to operate with an H.320 terminal, a more commonly encountered usage of ATM in support of H.261 video is that of $p \times 64$ Kbps circuit emulation using either AAL1 or AAL2, as described earlier in this chapter.

The Motion Photographic Experts Group (MPEG) version 2 standard, called MPEG-2, is the one that we focus on in this section. The encoded bit rate is dependent upon the amount of motion in the original video image, as well as options selected. At data rates between 3 and 15 Mbps, MPEG-2 achieves broadcast quality, while encoded bit rates of

Standard/Format	Encoded Bit Rate	Compression Ratio
Uncompressed NTSC broadcast-quality video	140 Mbps	1:1
H.261	p×64 Kbps, up to 2 Mbps	70:1 to 2100:1
H.262/MPEG-2	3–30 Mbps	5:1 to 45:1

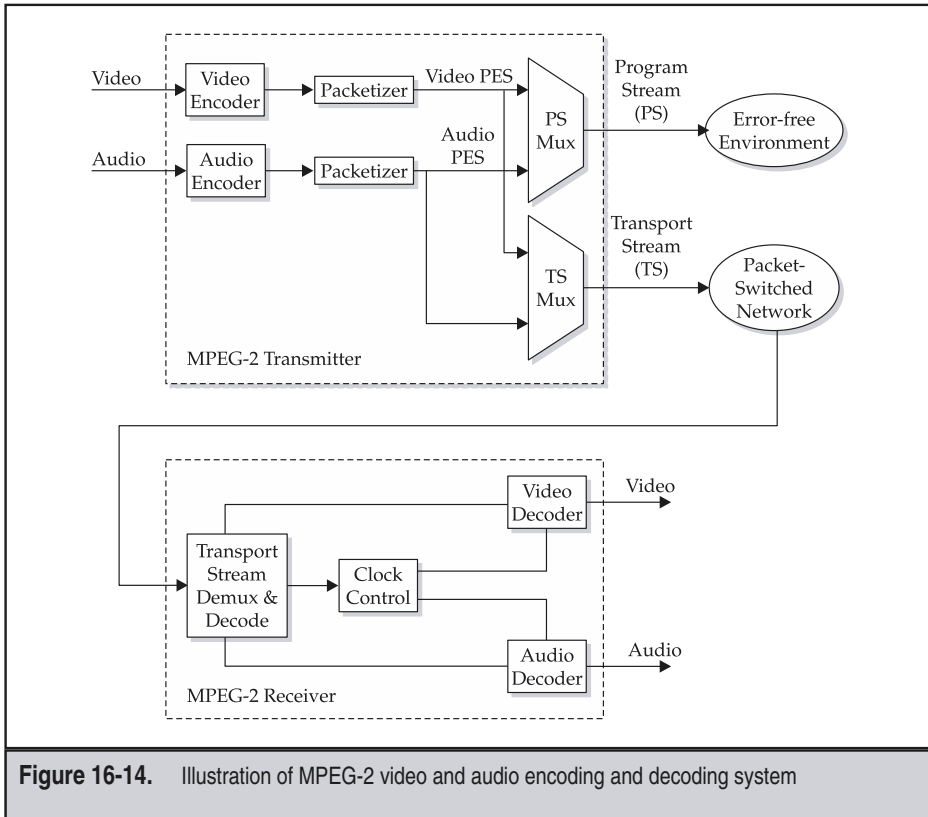
Table 16-5. Bit Rates and Compression Ratios for Video Coding Standards

up to 30 Mbps are capable of supporting high-definition television (HDTV). The next section summarizes some important aspects of MPEG-2.

MPEG-2 Video Over ATM and Packet Networks

ITU-T Recommendation H.222 (identical to ISO/IEC standard 13818-1:2000) defines the characteristics of the overall multimedia MPEG-2 transport stream, described next with reference to Figure 16-14. The drawing shows an MPEG-2 transmitter and receiver connected via a packet-switched network. The MPEG-2 coding standard for video is defined in ITU-T Recommendation H.262 (identical to ISO/IEC standard 13818-2:2000). The MPEG-2 coding standard for audio is defined in ISO/IEC standard 13818-3:2000. Next, the transmitter packetizes the encoded video and/or audio streams into the corresponding Packetized Elementary Stream (PES). A program stream (PS) multiplexer combines two or more PES inputs into a single program stream, which is intended for error-free environments, such as those used for manipulation of MPEG-2 media streams in software. The transport stream (TS) is designed for environments such as packet-switched networks where errors or impairments can occur, as shown in the right-hand side of the figure. TS packets are 188 octets in length.

The bottom part of Figure 16-14 illustrates the functions performed at an MPEG-2 receiver. First, the TS packets are demultiplexed and decoded into the constituent audio and video streams and delivered to their respective decoders. In support of the synchronization of the video and/or audio playback, the MPEG-2 receiver usually recovers the information related to clocking and relative timing of the media streams and provides this to the decoders. The MPEG-2 receiver removes delay variation introduced by the packet-switched network through the use of playback buffering. MPEG-2 relies on a packet-switched network providing some means of recovering the Program Clock Reference (PCR) to control the playback buffer. In ATM networks, a QoS class—for example, real-time Variable Bit Rate (rt-VBR)—provides low cell delay variation to meet this requirement. The IP protocol suite employs the timestamp field in the Real-Time Transport Protocol (RTP) header, as described in Chapter 8, to meet this need.



Most video coding schemes rely on the fact that after a scene change, the transmitter need only send the difference between subsequent images and the initial image of a video sequence. Specifically, in MPEG-2, an intra-coded picture (I-picture) specifies the “bootstrap” image for a subsequent series of predictive-coded pictures (P-pictures) that use information from previous I- or P-pictures, or bidirectionally predicted pictures (B-pictures) that use information from previous or subsequent pictures [ITU H.261, Orzessek 98]. The reason for doing predictive coding is that a P-picture is typically 30–50 percent the size of an I-picture, while a B-picture is often only half the size of a P-picture. Of course, the actual sizes depend upon the video source. However, if the network loses the beginning of an I-picture, then an entire video scene may be disrupted. As studied in Chapter 22, MPEG-2 also defines the means to prioritize critical information essential to reproduction of the basic image over that of finer-grained detail information.

The ATM Forum’s Audiovisual Multimedia Services Video on Demand (VOD) version 1.0 [AF VOD 1.0] specification defines support for audio, video, and data over ATM

on the basis of ITU-T Recommendation H.222.1. ITU-T Recommendation I.375.2 defines a similar concept. The ATM Forum VOD specification defines user signaling for connection control on the basis of the UNI signaling specification. The VOD specification defines how connection control, video, audio, data, and user control streams are encoded and multiplexed over up to three ATM VCCs using AAL5.

The ATM Forum VOD specification and ITU-T Recommendation J.82 define the means for carrying an MPEG-2 transport stream (TS) packets over a single VCC using the standard AAL5 protocol [ITU I.363.5] as described in Chapter 12. The video and/or audio information stored in the MPEG2 SPTS format results in packets that are 188 octets long as indicated at the top of Figure 16-15. The use of $N = 2$ SPTS packets per AAL5 common part convergence sublayer (CPCS) Protocol Data Unit (PDU) results in an efficient packing of the MPEG-2 stream into the payload of exactly eight ATM cells, as shown at the bottom of the figure.

ITU-T Recommendation J.82 also defines support for a constant rate MPEG-2 stream over AAL1 [ITU I.363.1], as described in Chapter 12. Figure 16-16 illustrates the encapsulation of MPEG-2 over unstructured data transfer mode AAL1. Since the AAL1 convergence sublayer (CS) payload is 47 octets, a 188-octet MPEG-SPTS packet fits in exactly four AAL1 CS-PDUs, as shown in the middle of the figure. The AAL1 segmentation and reassembly (SAR) PDU adds one octet of overhead per CS-PDU, resulting in a series of 48-octet SAR-PDUs, which fit precisely into the payload of four ATM cells per MPEG-2 TS packet, as shown at the bottom of the figure. The recommended means of timing transfer is use of the adaptive clock recovery mechanism described in Chapter 12. Delay variation is handled by using the ATM QoS class for constant bit rate traffic. Since AAL1 also has the option for Forward Error Correction (FEC), this method is useful for networks

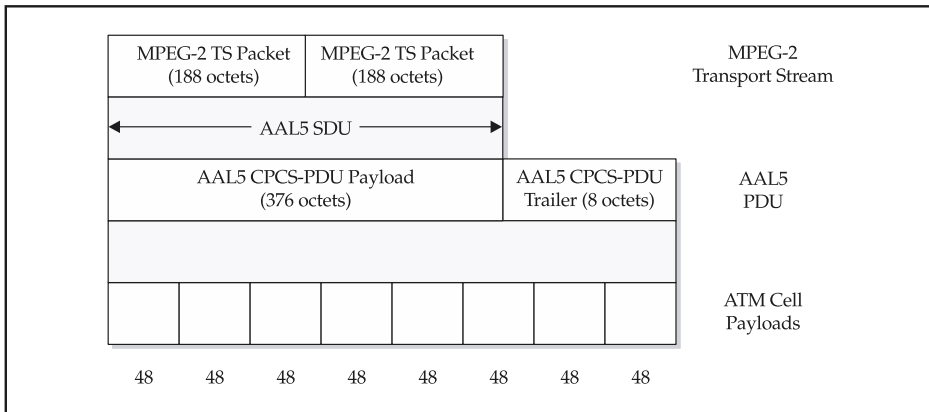
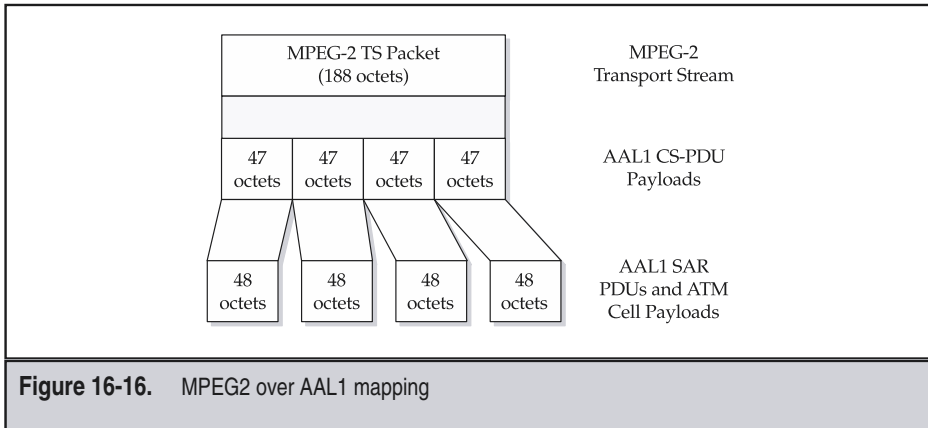


Figure 16-15. MPEG2 SPTS over AAL5 mapping



with loss and/or errors. A disadvantage of this mode is that the transmission is a constant rate, and therefore the compression ratio must be designed for the worst-case video encoding rate.

At the time of writing, work in the IETF pseudo-wire emulation efforts is also considering the standard 188-octet MPEG-2 packet as a particular size and type of “cell” for which services like clock recovery would be provided.

QoS Considerations Related to Video

Target end-to-end requirements for video over packet-switched networks are

- ▼ Packet loss ratio across the network on the order of one in a billion
- Cross-network transfer delay determined by whether the application is interactive or broadcast, as discussed in Chapter 20
- ▲ Packet delay variation determined by whether the application is interactive or broadcast, as discussed in Chapter 20

These are key attributes to look for when selecting a switch or service provider for high-performance video over packet applications. Many modern switches and routers can meet these requirements, if properly configured and operated. Network designers must pay close attention to the aggregation of the preceding impairments in larger networks to meet the stringent objectives required for high-performance video.

Video coding is also sensitive to errors and loss, since the error may cause the loss of some or all of the content in an I-picture. In order to protect against loss of such critical information, video coding often employs error-correction techniques. For example, the MPEG-2 video-coding standard requires a bit error rate of 10^{-11} to deliver broadcast-quality video [Orzessek 98].

REVIEW

This chapter covered packet-switched network support of voice, circuit emulation, and video. First, the text described the business drivers and general architecture for voice over packet networks. We then described how control signaling protocols operating between telephone and ATM switches can establish equivalent voice circuits over an ATM network. The text then covered specifics of how AAL2 can be used for VoATM trunking, and a similar MPLS Forum standard that defines support for VoMPLS trunking. The discussion then covered circuit emulation Service in support of $N \times 64$ kbps TDM circuits using structured mode, and support of standard North American, European, and Japanese digital signals (e.g., DS1, E1, and J2) in the unstructured mode. The chapter concluded with an overview of the N-ISDN and MPEG-2 standards commonly used in commercial video over packet network applications. The discussion focused on the high-performance MPEG-2 set of standards, and in particular on the methods defined for carrying video in this format over AAL1 and AAL5.

CHAPTER 17



Connection-Oriented Protocol Support

This chapter covers the application of ATM and MPLS for support of popular wide area connection-oriented data networking protocols, specifically with a focus on Frame Relay. The text does this by describing ATM's support for WAN data protocols, specifically Frame Relay (FR) and SMDS over ATM, and the state of standards and architectural principles for a variety of link layer protocols operating over MPLS or an IP-based tunnel. The chapter begins by defining the general concepts of interworking, logical access, trunking, and physical access over ATM and MPLS. The text then provides specific examples of interworking Frame Relay and ATM services, trunking of FR over ATM, and logical access to SMDS via ATM. We also give a summary of frame-based ATM protocols, namely, the ATM Data Exchange Interface (DXI), the Frame-Based UNI (FUNI), and Frame-Based ATM. The chapter concludes with some architectural principles involved in carrying link layer, or layer 2, protocols over a tunneled network, such as MPLS or IP. We give a few examples of prestandard protocols in this area to help clarify these concepts.

INTERWORKING, ACCESS, AND TRUNKING

This section looks at how the commonly used data communication protocols interact. The types of interactions fall into three categories: direct service interworking, access to one protocol via another, and the trunking (or network interworking) of one protocol over another. Figure 17-1 illustrates the relationships defined in standards between the major WAN protocols: Frame Relay, ATM, MPLS, and IP. The preceding edition of this book contained a similar analysis that included X.25 and SMDS [McDysan 98]; however, we omit these protocols in this edition in the interest of brevity. The notation of A and B defined at the bottom of the figure connected via a particular line or arrow style defines the relationships between protocols A and B. The following narrative speaks to each of these relationships. Figure 17-1a illustrates true interworking of the services provided by the protocol with a thick, solid curved line. Note that standards define true interworking only between Frame Relay and ATM. The analogy of service interworking with human languages occurs when each has exactly the same semantics, or meaning. Thus, as in languages, true service interworking occurs only when protocols have similar semantics. For some protocols, as is the case in some languages, some concepts simply don't translate. FR and ATM interwork because they are both connection-oriented protocols possessing similar status indication methods, as well as connection establishment and release procedures, as studied in Part 2. On the other hand, IP is connectionless, and hence it cannot directly interwork with a connection-oriented protocol, since the basic semantics differ. A similar situation occurs in human languages, where different cultural concepts don't translate. Figure 17-1a also illustrates logical access, via directed arrows. Many standards define how to access one protocol via another. For example, a user connected to an ATM or FR network can access an IP network. Furthermore, although the means are not completely standardized, some vendors and service providers are considering ways to use MPLS to access IP, as well as define service interworking between ATM and MPLS. The head of the arrow in the figure indicates the protocol that is being accessed. Note from the figure that standards exist to access IP via all of these protocols.

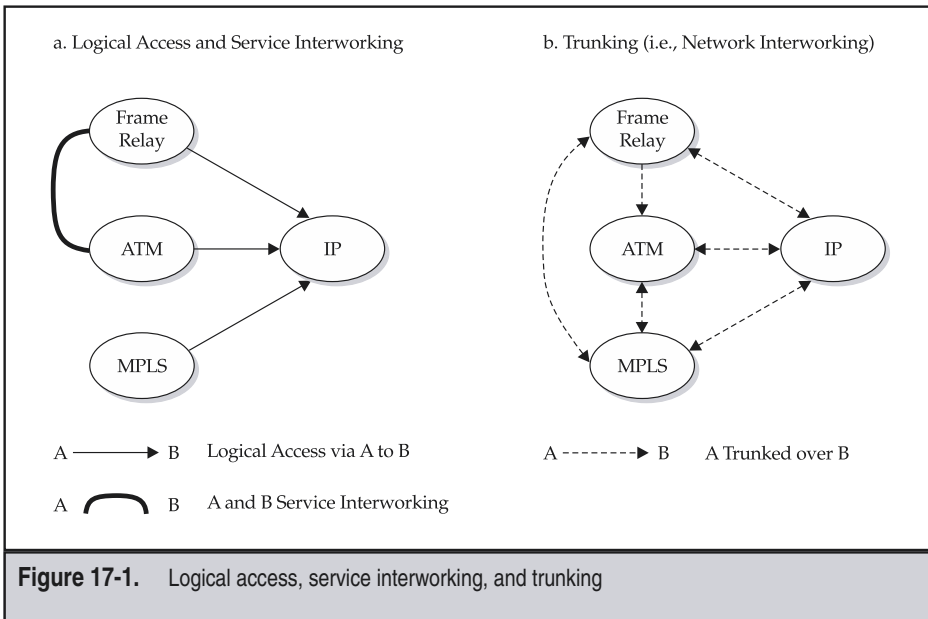


Figure 17-1b illustrates another key concept of protocols, namely that almost every one can be trunked (or equivalently, tunneled) over another. The directed dashed arrows in the figure indicate the cases in which one protocol can be trunked over another. Note from the figure that ATM, MPLS, or IP can potentially be the trunking protocol for any of the other protocols. This reflects the original ITU-T vision for B-ISDN, of which ATM is the foundation. The addition of MPLS and IP as trunking protocols reflects the emerging requirement from by the IETF pseudo-wire emulation efforts. Of course, the trunking protocol must support the QoS required by the trunked protocol in a standard manner. The business consequence of protocol trunking standards is that some carriers and enterprises trunked multiple services over a common ATM infrastructure and will soon have the opportunity to use MPLS and/or IP trunking infrastructure to achieve similar goals. Of course, each WAN protocol has the capability of separate physical access, which may be either dedicated or switched.

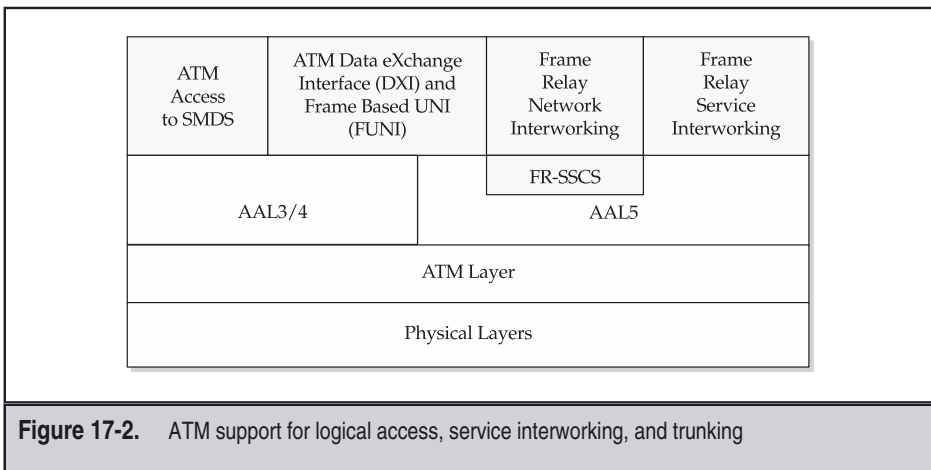
Table 17-1 summarizes the differentiating attributes of the logical access, service interworking, and trunking functions. The first is the network context of whether the function applies at the User-Network Interface (UNI) or the Network-Node Interface (NNI), as defined in Chapter 10. The next important attribute is whether the protocols supported must be the same, or whether they can be different. If they can be different, then usually some form of translation or mapping of specific protocol functions is necessary, as is done for service interworking only. Finally, encapsulation or tunneling is required when carrying one protocol over another, as is the case for access and trunking.

Function	Network Context	Protocols Supported	Translation Required	Tunnel Required
Access	UNI	Must be the same	No	Yes
Service Interworking	UNI or NNI	Different	Yes	No
Trunking	NNI	Must be the same	No	Yes

Table 17-1. Summary of Important Attributes for Access, Service Interworking, and Trunking

The limited interworking, access, and trunking rules summarized in Figure 17-1 mean that a network designer cannot arbitrarily interconnect protocols using ATM and MPLS but must take some care in the design. Beware of other texts or papers that claim ATM or MPLS provides the ultimate translation capability between all protocols—it is simply an incorrect statement. The Internet Protocol (IP) is the only standard protocol that meets this need today, since it can be accessed via FR, ATM, MPLS, Ethernet, SMDS, and X.25, as well as dedicated TDM or dial-up access.

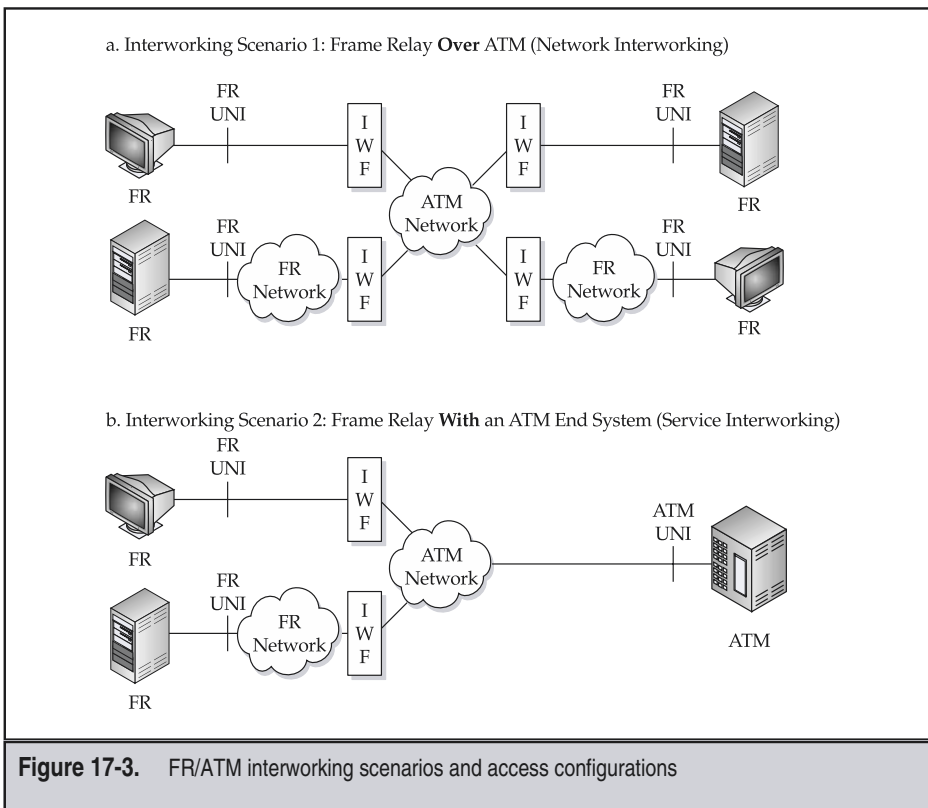
The next sections cover the interworking, trunking, and access protocols for Frame Relay and SMDS with respect to ATM, as illustrated in Figure 17-2. We then cover the frame-based protocols designed in support of ATM. Chapter 19 covers the subjects of access via ATM to IP and IP trunking over ATM. The material at the end of the chapter covers the emerging protocol support for various types of connection-oriented services over MPLS.



OVERVIEW OF FRAME RELAY/ATM INTERWORKING

The Frame Relay service is a multibillion dollar market, and despite premature reports of its demise, it continues to grow at a healthy pace. Some Frame Relay users need higher-speed connections and applications that require multiple service categories. In response to this need, some customer networks migrated their largest locations to ATM, which required interworking between a few large ATM sites and many smaller Frame Relay–connected sites as described later. In addition, some carrier networks and large enterprises trunked multiple services over ATM, including Frame Relay and IP. In response to these business needs, ITU-T Recommendation I.555, the ATM Forum B-ICI specification, and three Frame Relay Forum (FRF) Implementation Agreements (IAs) specify interworking between Frame Relay and ATM.

ITU-T Recommendation I.555 labels these types of FR/ATM interworking as scenario 1 and 2 as shown in Figure 17-3. In scenario 1, FR is interworked (or trunked) *over* ATM, while in scenario 2, an FR end system interworks directly *with* an ATM end system.



scenario 1, FR CPE either directly interfaces to an interworking function (IWF) via an FR UNI or connects via a Frame Relay network. The access configuration in which an ATM end system connects directly via an ATM UNI, which then connects to an IWF, applies to service interworking in scenario 2 only. The Frame Relay Forum details these scenarios in three implementation agreements: FR/ATM Network Interworking [FRF.5], FR/ATM Service Interworking [FRF.8.1], and finally FR/ATM Service Interworking with the FRF.8.1 paradigms, in which the system also performs control signaling interworking [FRF.18]. FRF.5 and FRF.8.1 both concern permanent connections (i.e., PVCs), whereas FRF.18 provides support for switched connections (i.e., SVCs and SPVCs). Let's now explore the network and service interworking scenarios in more detail.

FRAME RELAY/ATM NETWORK INTERWORKING

Figure 17-4 illustrates further details of the FR/ATM network interworking, or trunking, protocol. The FR to ATM network interworking function (IWF) converts between the basic Frame Relay functions and the FR Service-Specific Convergence Sublayer (FR-SSCS) defined in ITU-T Recommendation I.365.1, operating over the Common Part of AAL5 (see Chapter 12). The network IWF also conveys the FR status signaling for a FR UNI port across the one or more VCCs corresponding to the Frame Relay Data Link Connection Identifiers (DLCIs). In the FR/ATM network interworking scenario, the upper layer protocols must be compatible.

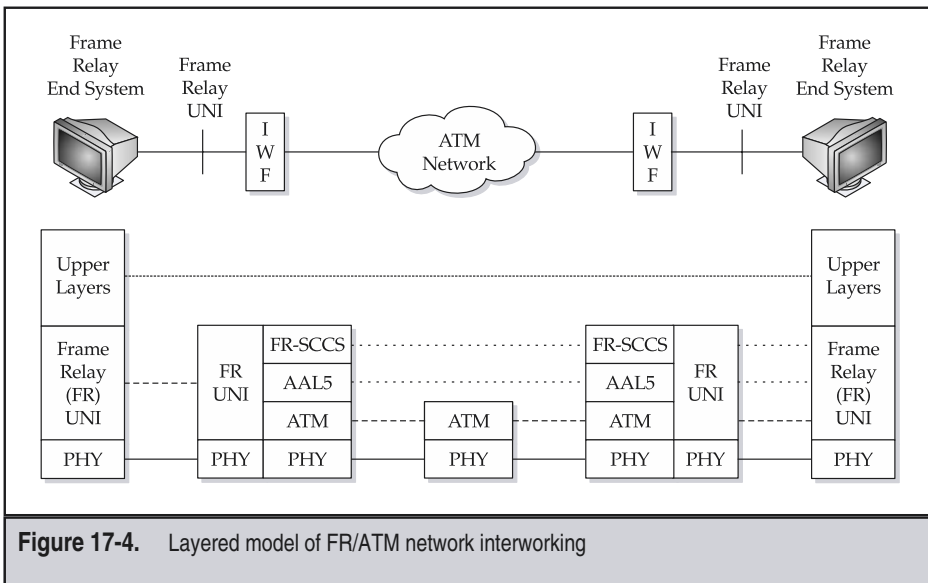
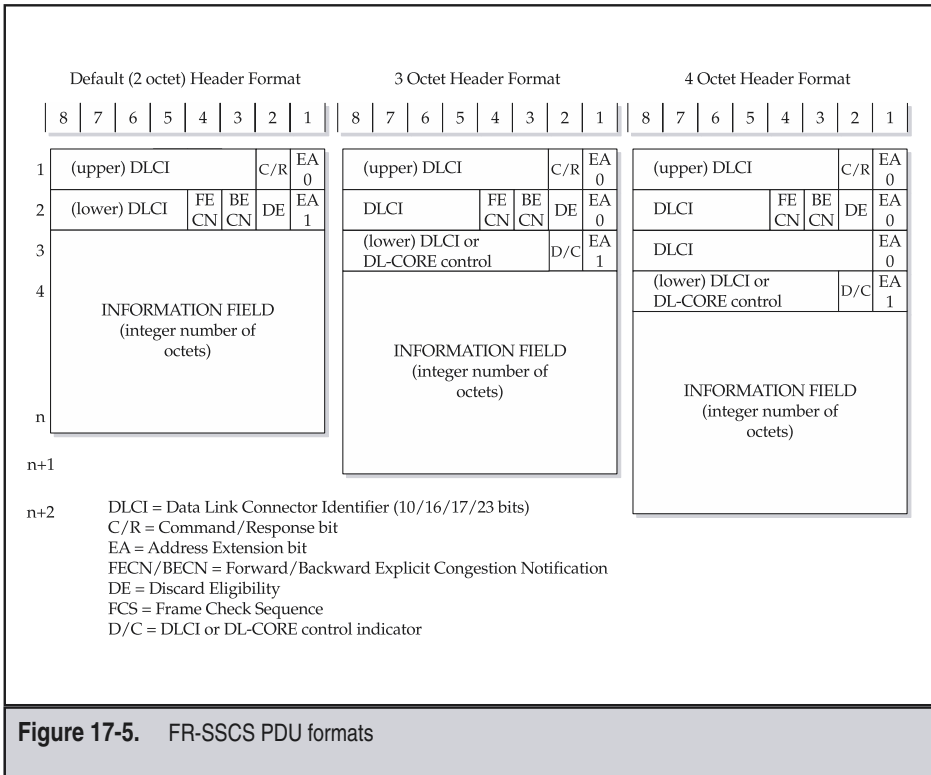


Figure 17-4. Layered model of FR/ATM network interworking

FR Service-Specific Convergence Sublayer (FR-SSCS)

Figure 17-5 depicts the Frame Relay Service-Specific Convergence Sublayer (FR-SSCS) PDU format (essentially the FR frame summarized in Chapter 7) with inserted zeros and the trailing CRC both removed. Frame Relay supports either 2-, 3-, or 4-octet addressing as indicated in the figure. The FR-SSCS supports multiplexing through the use of the DLCI field, with the ATM layer supporting connection multiplexing using the VPI/VCI fields in the cell header. There are two methods of multiplexing FR connections over ATM: many-to-one and one-to-one. Many-to-one multiplexing maps many FR logical connections identified by the FR DLCIs over a single ATM Virtual Channel Connection (VCC). One-to-one multiplexing maps each FR logical connection identified by DLCI to a single ATM VCC via VPI/VCI at the ATM layer. Many-to-one multiplexing is best suited to efficiently trunking FR over ATM, since this method efficiently carries the status signaling between FR networks. However, the many-to-one mapping is optional in the standard, and therefore many vendors don't implement it.



Status Signaling Conversion

The FR to ATM interworking function (IWF) converts between the Q.922 core functions and the FR Service-Specific Convergence Sublayer (FR-SSCS) defined in I.365.1, the AAL5 Common Convergence Sublayer (CPCS), and the Segmentation and Reassembly (SAR) sublayers from ITU-T I.363.5, as shown in the left-hand side of Figure 17-6a. The IWF must also convert between the Q.933 Annex A PVC status signaling for a single, physical FR UNI port and the one or more VCCs that correspond to the DLCIs. The FR-SSCS Protocol Data Unit (PDU) is the CPCS SDU of the AAL5 Common Part as described in Chapter 12. Figure 17-6b illustrates the FR/ATM interworking protocol of an ATM end system. This function is identical to the right-hand side of the FR/ATM IWF. The ATM end system must support Q.933 Annex A Frame Relay status signaling for each FR/ATM network interworking VCC as indicated in the figure. If the FR/ATM IWF and the end system both support many-to-one multiplexing, then the encapsulated FR status signaling channel contains the status for more than one DLCI. In the case of one-to-one multiplexing, each VCC carries the user DLCI and the status signaling DLCI, which reports on the status of a single DLCI.

Congestion Control and Traffic Parameter Mapping

The interworking function (IWF) either maps or encapsulates control and addressing functions along with the associated data. For example, in the FR-to-ATM direction, the

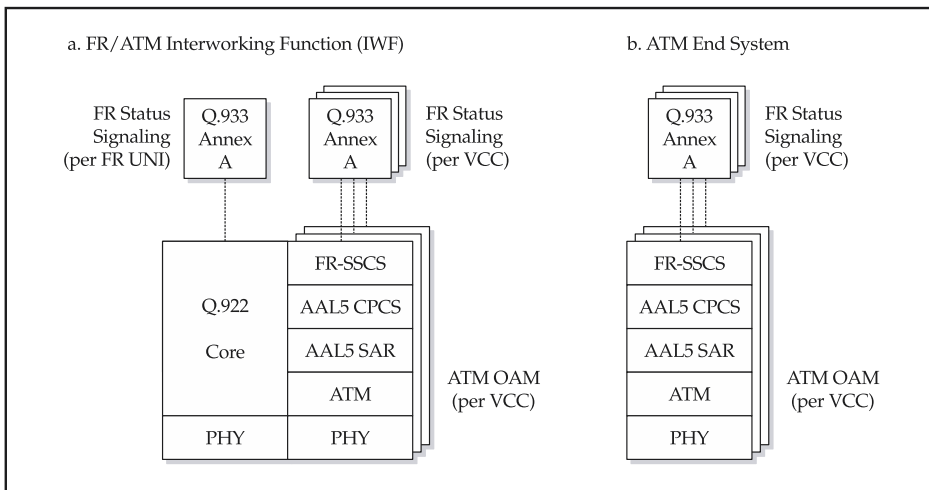


Figure 17-6. FR to ATM interworking control plane protocol stacks

IWF encapsulates the DLCI, DE, FECN, and BECN fields in the FR-SSCS PDU. The IWF maps the FR DE bit to the ATM CLP bit and also maps the FR FECN bit to the ATM EFCI bit. The IWF encapsulates the FR BECN bit in the FR-SSCS. The IWF maps the Frame Relay FCS to (that is, it replaces it by) the AAL5 CRC function.

In the ATM-to-Frame Relay direction the CLP bit may be logically ORed with the DE bit as a configuration option on a per-DLCI basis. The IWF may also map an EFCI indication in the last cell of a reassembled frame to the BECN bit. The IWF also checks the AAL5 CRC and recomputes the FR FCS for delivery to a Frame Relay UNI. The FR-SSCS PDU carries the encapsulated FECN, BECN, and DE bits intact.

As described in Chapter 7, the FR traffic parameters include access line rate (AR), committed burst size (Bc), excess burst size (Be), and measurement interval (T). These FR traffic parameters define a Committed Information Rate (CIR) and an Excess Information Rate (EIR). As described in ATM Forum B-ICI specification 2.0 and FRF.18, the interworking function is configured to map these FR traffic parameters to the ATM traffic parameter in terms of Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS). Usually, Frame Relay utilizes the ATM nrt-VBR service category; however, other categories may also be used in conjunction with prioritized FR service. Annex D of ITU-T Recommendation I.555 provides a similar mapping to FR traffic parameters to that of the Statistical Bit Rate (SBR) ATM Transfer Capability (ATC).

FRAME RELAY/ATM SERVICE INTERWORKING

The ATM Forum worked closely with the Frame Relay Forum to develop a FR/ATM service interworking specification [FRF.8.1]. Figure 17-7 illustrates the user plane protocol stacks for FR/ATM service interworking. Note that a Frame Relay end system directly communicates with an ATM end system in this scenario. The figure illustrates three possible implementations of the interworking function (IWF). Case a in the figure illustrates ATM network implementing the IWF, while case c shows the IWF implemented in the Frame Relay network. Case b illustrates a design employing a separate interworking function. A fourth case (not illustrated in the figure) is also possible where the same multiservice switches implement both the Frame Relay and ATM protocols.

Mapping between different multiprotocol encapsulations standards for FR and ATM, as specified in RFCs 2427 (which obsoletes RFC 1490) and 2684 (which obsoletes RFC 1483), respectively, is an optional protocol translation function. Unfortunately, the multiprotocol encapsulation formats for Frame Relay and ATM differ. RFC 2427 specifies a Network Level Protocol ID (NLPID), Subnetwork Attachment Point (SNAP) format for Frame Relay, while RFC 2684 specifies a LAN-compatible Logical Link Control (LLC) SNAP format. RFC 2684 handles the case in which the IWF does not provide a protocol translation function by requiring that the ATM end system use the NLPID/SNAP multiprotocol encapsulation specified in RFC 2427.

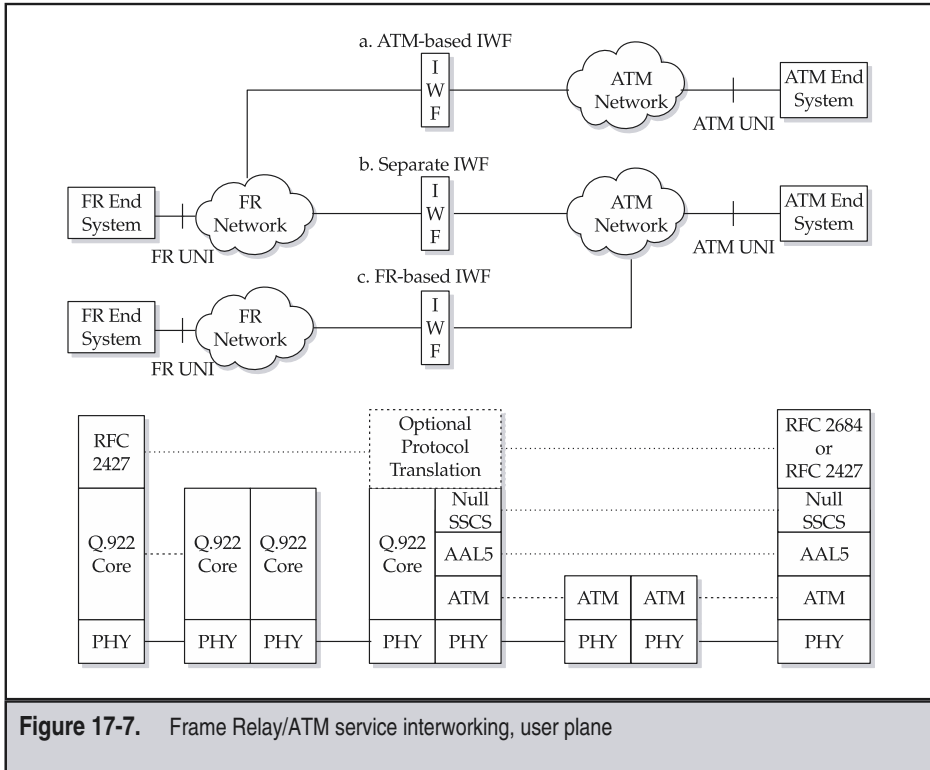


Figure 17-7. Frame Relay/ATM service interworking, user plane

Status Signaling Interworking

Figure 17-8 illustrates the mapping between FR status signaling and the ATM fault and status indications. If the Frame Relay status signaling active bit indicates a failed DLCI, then the IWF generates an ATM OAM cell indicating an Alarm Indication Signal (AIS) fault as defined in Chapter 28. The Frame Relay new bit causes the IWF to generate an ATM ILMI status change trap. In addition, ITU-T Recommendation I.555 defines an optional means to use ATM OAM Continuity Check (CC) cells on the ATM side of the connection in response to a new PVC, specifies the use of ATM OAM loopback cells to confirm that the ATM PVC is active, and generalizes ILMI-specific references to that of a generic network management system (NMS). These mappings communicate the semantics of end-to-end DLCI and VCC status to the FR and ATM end systems, respectively.

FR/ATM service interworking interconnects end users and was not designed as a trunking protocol. FR/ATM network interworking was designed for carrier interconnection of FR services over ATM [AF BICI 2.0, FRF.18]. The mapping of status signaling to

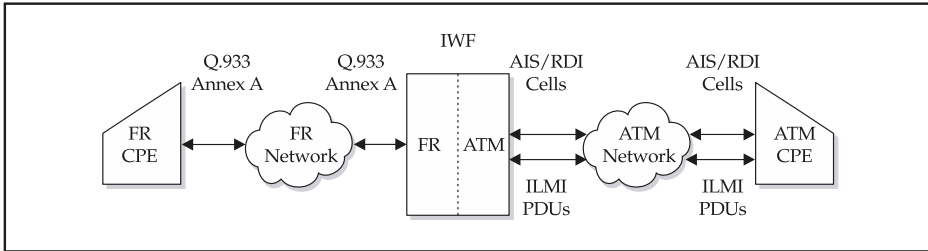


Figure 17-8. Frame Relay/ATM status signaling interworking

ATM OAM cells and ILMI traps is less efficient than the many-to-one multiplexing mode of FR/ATM network interworking described previously.

Address Resolution Protocol Interworking

The Frame Relay Forum's FR/ATM Service Interworking specification also specifies procedures for mapping the Address Resolution Protocol (ARP) [RFC 826] and Inverse ARP (InARP) [RFC 1293] between their Frame Relay [RFC 2427] and ATM counterparts (i.e., the PVC portions of RFC 2225 detailed in Chapter 19) when operating in the optional translation mode.

The use of these encapsulation mappings allows the interworking function to recognize and perform special processing on these ARP and InARP packets. As described in Chapter 9, IP devices, such as routers or workstations, automatically determine the link layer address using the ARP protocol. ARP encapsulation mapping allows IP end systems to automatically determine the correct FR DLCI and ATM VPI/VCI corresponding to a FR/ATM Service Interworking PVC on which to transfer IP packets. In order to perform the ARP and InARP mapping functions, the interworking function contains a table with the following information statically configured in each row for each of the four FR/ATM Service Interworking PVCs, as shown in Figure 17-9:

- ▼ IWF Frame Relay port number (P1)
- Frame Relay DLCI number on the Frame Relay port (aa, bb, cc, dd)
- ATM port number on the IWF corresponding to the FR PVC (P1, P2)
- ▲ VPI/VCI on ATM port corresponding to the FR PVC (rr, ss, tt, uu)

We now give an example of the ARP protocol with reference to Figure 17-9. Note that a full mesh of PVCs interconnects the four CPE devices in the example. Furthermore, the IP addresses are all on the same subnet as defined in Chapter 9. Therefore, when FR-CPE with IP address IP_a wants to send a packet to the device IP address IP_x but doesn't know the correct link layer address, it sends out an ARP packet on all the PVCs. The FR/ATM

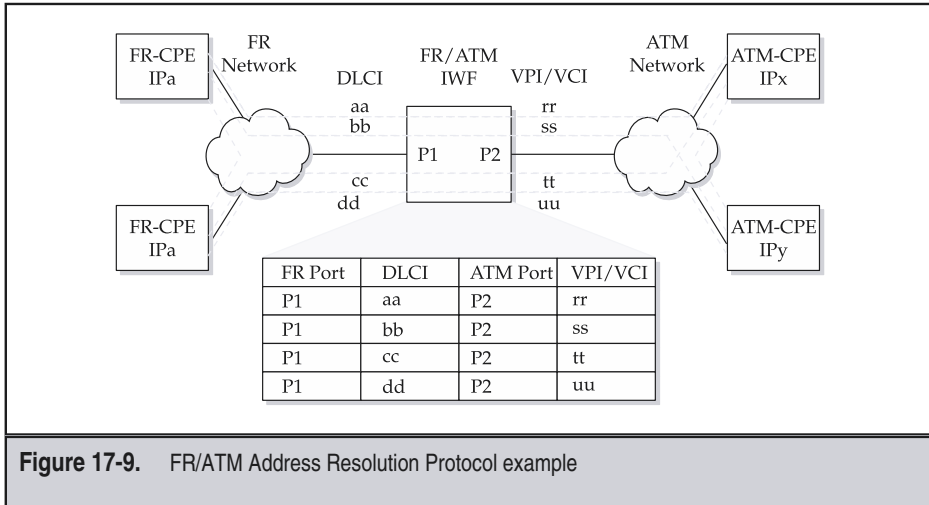


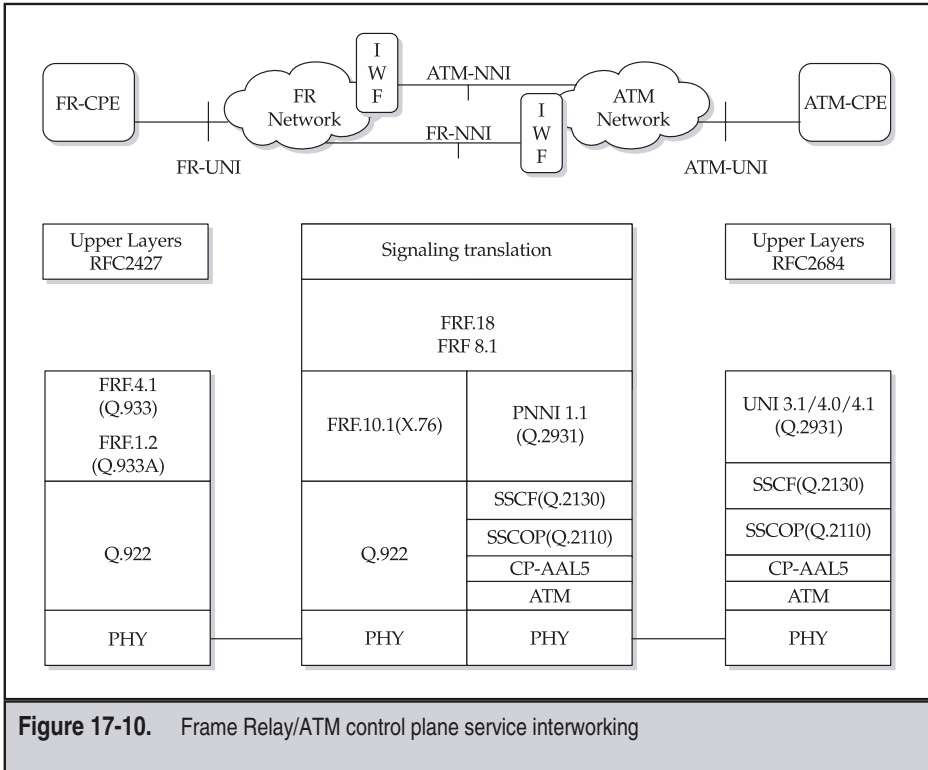
Figure 17-9. FR/ATM Address Resolution Protocol example

IWF receives the ARP packet on DLCIs aa and bb on port P1, converts it into the correct ATM format, and sends it out on ATM port P2 on VPI/VCI rr and ss. The ATM-CPE with IP address IPx responds to the ARP request with an ARP reply in the ATM format on VPI/VCI rr on port P2, which the FR/ATM IWF converts into the Frame Relay format and transmits on Frame Relay port P1 on DLCI aa. Now, FR-CPE with address IPa sends IP packets to address IPx on DLCI aa, which the FR/ATM IWF transmits to the ATM-CPE with address IPx on VPI/VCI rr on ATM port P2.

The Frame Relay Forum's FR/ATM Service Interworking specification [FRF.8.1] also specifies SVC addresses as an optional, yet desirable addition to the just-described mapping in support of FR/ATM SVC Interworking.

FR/ATM SVC Service Interworking

So far we have looked at Frame Relay/ATM Interworking only for permanent virtual connections. The FRF.18 specification defines the means for both switched virtual Frame Relay connections (SVC) and switched permanent Frame Relay virtual connections (SPVC) to be established via an interworking function to a corresponding ATM SVC or SPVC. Figure 17-10 shows the control plane protocols involved with the interworking function placed at either a FR NNI (FRF.10.1) or an ATM NNI (PNNI). Although Frame Relay and ATM SVCs have not seen great commercial adoption by end users, the SPVC functions are widely used by service providers for provisioning PVCs. Unfortunately, many of the SPVC implementations have proprietary components, and the objective of FRF.18 is to support standard interworking of SPVC and SVC features between FR and ATM networks. In the SPVC application, the protocol provides a resilient NNI between Frame and ATM networks, where if the NNI link breaks, connections can use a



completely different path and switch to reestablish connectivity. If a Frame Relay UNI port is directly attached to an ATM node, this specification also gives the standard for implementing a Frame Relay UNI with Frame Relay/ATM SVC interworking capability. We now describe the set of protocols implemented in the Frame Relay CPE (FR-CPE), FR/ATM control plane IWF, and the ATM-CPE, as shown in Figure 17-10. In the FR-CPE, FRF.4.1 provides the Frame SVC/SPVC signaling that connects to the FRF10.1 signaling in the IWF. Within the IWF, FRF.18 performs FR/ATM signaling interworking, which generates ATM UNI 4.1 SVC signaling toward the ATM CPE, which establishes the ATM leg of the connection.

Note that an FR-FR SVC or SVPC can be implemented over an ATM network by repeating the mirror image of this protocol sequence at the destination IWF, resulting in a form of operation that can be viewed as instances of FR/ATM service interworking connected “back to back.” Originally, the FRF.8.1 PVC specification did not describe support for such “back to back” operation, since this was already defined in FRF.5, as described earlier. However, some implementations do in fact use FRF.8.1 methods “back to back,”

although this configuration does have some drawbacks. A significant issue is that the mapping to and from FR new and active bit status signaling to ATM OAM and ILMI described earlier may not detect all failure conditions (e.g., if the IWF fails), and therefore FR status signaling may not be supported end-to-end.

Another drawback is that the dynamic mapping of the traffic parameters from FR to ATM at the originating port will likely not be identical at the terminating end, where the parameters are mapped from ATM back to FR, as can be seen from the calculation to determine the parameters in [B-ICI 2.0, FRF.18]. Furthermore, the OAM cells used for ATM SVCs and SPVCs are not supported in FR standards. Therefore, FRF.18 specifies that the IWF apply the appropriate processing and terminate ATM OAM cells. If the specification is followed to the letter, then the net result is that the end-to-end FR status signaling is not supported. This is an area where further standardization is necessary for SPVCs to have status signaling and OAM behavior that is comparable to that of PVCs.

FRF.18 also supports address translation of FR NSAP addresses into ATM AESAs, as well as selective translation of FR information elements to a corresponding equivalent ATM information element, if there is a corresponding one, and vice versa. The incomplete mapping can cause some difficulties, especially if "back-to-back" FR ports are used, and therefore the generic application transport information element (GAT), defined in both FRF.10.1 and PNNI 1.1, can be used to transparently transport information elements that cannot be mapped or for other proprietary contents that may be needed at both ends. FRF.18 also suggests a mapping of the four FR service classes as defined in ITU-T Recommendation X.146 to ATM service categories, a potentially useful convention that supports the FR QoS semantics across an ATM network. Unfortunately, FRF.18 only suggests this use, and therefore implementations supporting both FR and ATM QoS may not interoperate.

FR/ATM Interworking Applied

Large enterprise networks typically have a few locations that serve as major traffic sources and sinks. Typical applications are large computer centers, large office complexes with many information workers, campuses requiring high-tech communication, server farms, data or image repositories, and large-volume data or image sources. These large locations have a significant community of interest among them; however, the enterprise usually also requires a relatively large number of smaller locations needing at least partial, lower-performance access to this same information. The smaller locations have fewer users and generally cannot justify the higher cost of equipment or networking facilities. Generally, cost increases as performance, number of features, and flexibility increase.

Some users have deployed hybrid networking solutions using high-speed ATM at the larger sites and interworking with lower-speed Frame Relay at the many smaller locations. These lower-speed access sites require more efficient access rather than high performance, and thus Frame Relay access through low-end routing and bridging products is often more cost effective than ATM. This is because the cost per bit per second generally decreases as the public network access speed increases. For example, the approximate ratio of DS1/DS0 and DS3/DS1 tariffs is approximately 10:1, while the speed difference is

approximately 25:1. This means that a higher-speed interface can be operated at 40 percent efficiency at the same cost per bit per second. Conversely, the lower-speed interface costs 3.5 times as much per bit per second, and therefore efficient use of capacity is more important.

What does a typical virtual enterprise network using FR and ATM look like? Figure 17-11 illustrates an ATM-based interworking network cloud connecting a few large ATM sites to many smaller Frame Relay sites. Such a network has many smaller sites and few larger sites, which is typical of large enterprises, such as corporations, governments, and other organizations. Principal needs that drive the need for ATM are multiple levels of service characterized by parameters such as throughput, quality, and usage-based billing.

An issue with FR/ATM interworking arises if the application requires end-to-end QoS support. Since the ATM designers started with QoS as a fundamental requirement, while Frame Relay was initially conceived as a data-only service; FR/ATM interworking doesn't necessarily guarantee end-to-end QoS [Sullebarger 97]. Therefore, if QoS is critical to your application (for example, voice or video), check to see that the Frame Relay service provider and equipment supplier support your needs.

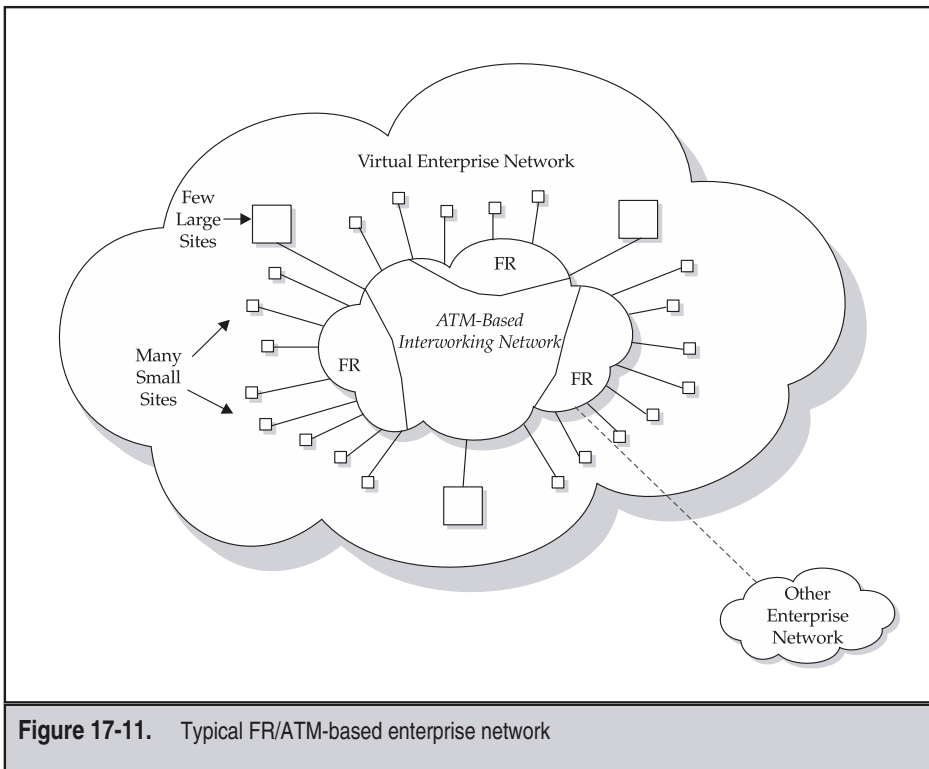


Figure 17-11. Typical FR/ATM-based enterprise network

ATM ACCESS TO SMDS

As summarized in Chapter 8, Bellcore created the Switched Multimegabit Data Service (SMDS) in the early 1990s to bridge the interval until ATM matured. In Europe, a similar service called Connectionless Broadband Data Service (CBDS) emerged to fill a similar need. As ATM became available in the mid 1990s, many users wanted to move to ATM but didn't want to write off their investment in SMDS. At the same time, service providers wanted a means to more efficiently trunk multiple services over a common network, and ATM seemed the logical choice. The ATM Forum's B-ICI 1.1 specification first defined how to transport SMDS between switches and even between carriers. The ATM Forum, SMDS Interest Group (SIG), and European SIG (E-SIG) jointly specified how a user should access SMDS across an ATM UNI to support customers who want to share ATM access lines between SMDS and access to other ATM and ATM-based interworking services. This helped accomplish the migration of customers from SMDS to ATM. However, as FR, ATM, and IP services became available, most customers migrated away from SMDS. We present a brief summary of this technology as an example of service interworking and a link layer networking migration approach.

Figure 17-12 depicts the access configuration and logical placement of functions for accessing SMDS features over an ATM User-Network Interface (UNI) [SIG SMDS-ATM]. An ATM end system accessing SMDS over ATM must format either an AAL3/4 or AAL5 CPCS PDU containing the SMDS Connectionless Service (SIP_CLS) PDU, as shown on the left-hand side of the figure. The ATM network performs a Usage Parameter Control (UPC) function to emulate the SMDS access class. The ATM network relays the

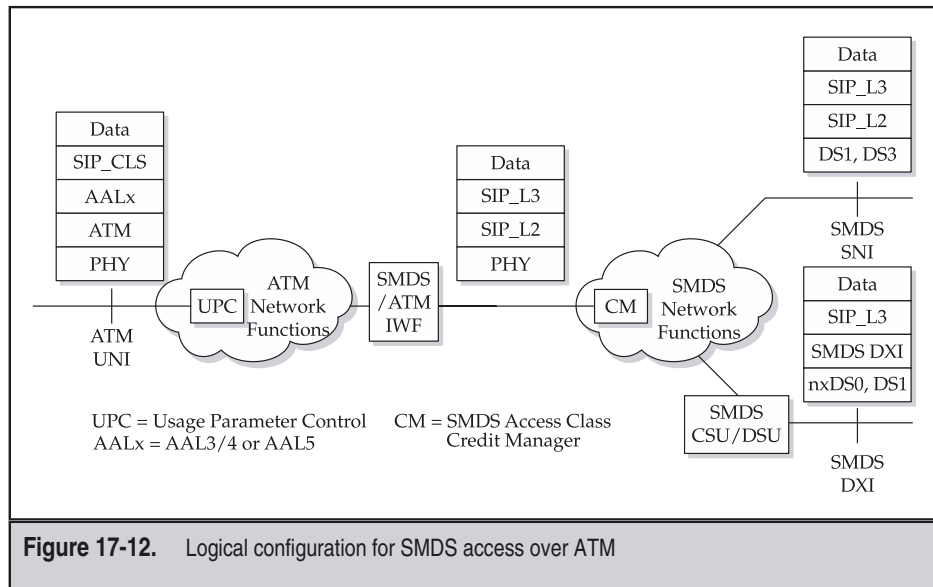


Figure 17-12. Logical configuration for SMDS access over ATM

cells to an SMDS interworking function (IWF), which may be implemented in a centralized, regionalized, or distributed manner. The SMDS/ATM IWF converts the AAL stream into the SMDS Level 2 and 3 protocol stack and passes this to an SMDS network, which implements the SMDS service features, including access class enforcement via the Credit Manager (CM). The mapping between the SMDS access class Credit Manager parameters and the ATM traffic parameters is a key function. The ATM Forum version 3.1 UNI specification described a means for doing this using the peak and sustainable cell rates or only the peak rate. The SMDS network can interface to a subscriber using the SMDS Subscriber Network Interface (SNI) [Bellcore SMDS] or the SMDS Data Exchange Interface (DXI) [SIG DXI], as shown in the figure. SMDS DXI was the predecessor to the ATM Forum's DXI, described in the next section. The SMDS and ATM DXI protocols were often used to interconnect multiprotocol routers across dissimilar link layer networks.

FRAME-BASED INTERFACES SUPPORTING ATM

Initially, the ITU-T and the ATM Forum focused on the specification of high-speed interfaces for ATM. However, the expense of high-speed lines in carrier networks and the costs of early ATM devices drove the need for lower-speed interfaces, or means to utilize existing hardware via software changes only. In response to this need, the ATM Forum defined two types of low-speed, frame-based ATM interfaces to support such connections to ATM networks. Later, as it became clear that the predominant traffic type would be IP variable-length packets and in recognition of the fact that ATM AAL5 was rather inefficient, the ATM Forum developed two specifications for more efficient transport of AAL5 frames over SONET/SDH or Ethernet links. We discuss these protocols after first covering the older ones designed for lower-speed links and legacy devices.

The ATM Data Exchange Interface (DXI) protocol allowed early adopters to utilize ATM with existing routers and data equipment with serial interfaces using a separate piece of equipment to convert between the ATM DXI protocol and an ATM UNI, similar to early implementations of SMDS. The ATM Forum then adapted the ATM DXI protocol to a WAN interface, calling this specification the ATM Frame-Based UNI (FUNI). A main advantage of FUNI is that it eliminates the external CSU/DSU required in the ATM DXI specification. This section describes the context, operation, and application of these two important protocols.

ATM Data Exchange Interface (DXI)

Many users have asked the following question: what if I want the capabilities of ATM over the WAN, but I can't afford the cost of a DS3 or OC-3 access line? The answer could be the ATM Forum-specified ATM Data Exchange Interface (DXI) [AF DXI], which supports either the V.35, RS449, or HSSI DTE-DCE interface at speeds from several Kbps up to and including 50 Mbps. ATM DXI specifies the interface between a DTE, such as a router, and a DCE, usually called an ATM CSU/DSU, which provides the conversion to an ATM UNI, as illustrated in Figure 17-13. Like the SMDS DXI on which the Forum patterned the ATM DXI specification, the context is a limited-distance DTE-DCE interface.

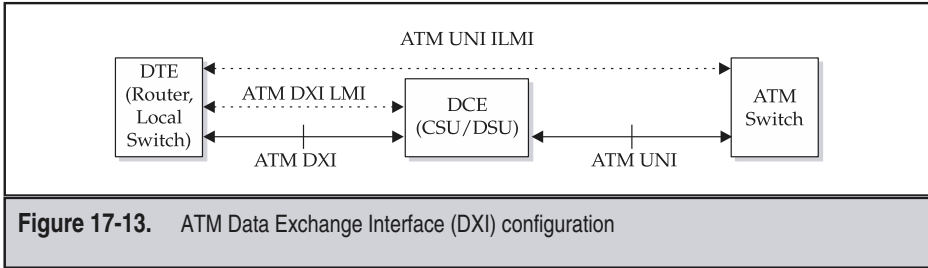


Figure 17-13. ATM Data Exchange Interface (DXI) configuration

The DTE manages the ATM DXI interface through a Local Management Interface (LMI), while the CSU/DSU passes the ATM UNI Interim Local Management Interface (ILMI) Simple Network Management Protocol (SNMP) messages through to the DTE.

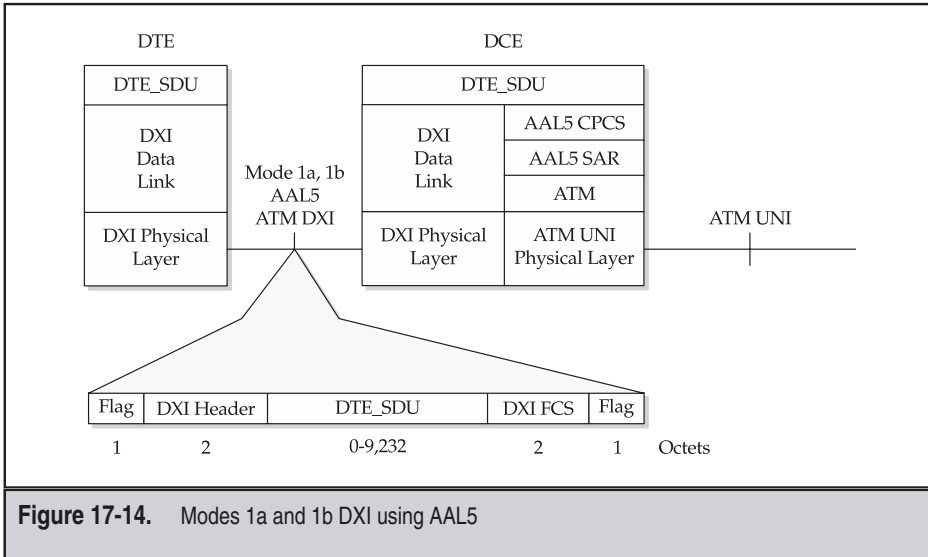
Table 17-2 summarizes the key attributes of the various modes of the ATM DXI specification. All modes support AAL5, while some modes also support AAL3/4. The maximum number of Virtual Channel Connections (VCCs) differs in the various modes, as does the maximum frame size, due to the address bits in the DXI header and the length of the Frame Check Sequence (FCS). Mode 1a is the most widely used ATM DXI interface.

ATM DXI: Mode 1a and Mode 1b

ATM DXI Mode 1 supports two implementations. Both mode 1a and mode 1b define DCE support for AAL5, as shown in Figure 17-14. The protocol encapsulates the DTE Service Data Unit (SDU) in the AAL5 CPCS and then segments it into ATM cells using the AAL5 Common Part Convergence Sublayer (CPCS) and Segmentation and Reassembly (SAR) sublayer functions defined in Chapter 12. The two-octet DXI header defined later in this section prefixes the DTE SDU. The two-octet Frame Check Sequence (FCS) is the

Characteristic	Mode 1a	Mode 1b	Mode 2
Maximum number of VCCs	1023	1023	16,777,215
AAL5 Support	Yes	Yes	Yes
AAL3/4 Support	No	Yes	Yes
Maximum DTE SDU Length			
AAL5	9232	9232	65,535
AAL3/4	N/A	9224	65,535
Bits in FCS	16	16	32

Table 17-2. Summary of ATM DXI Mode Characteristics



same as that used in Frame Relay and HDLC, and hence much existing DTE hardware supports mode 1 through software changes only.

Mode 1b adds support for the AAL3/4 CPCS and SAR on a per-VCC basis. The DTE must know that the DCE is operating in mode 1b AAL3/4, since it must add the four octets for both the CPCS PDU header and trailer. This decreases the maximum-length DTE SDU by eight octets. The same two-octet DXI header used for the AAL5 VCC operation is employed.

ATM DXI: Mode 2

Mode 2 uses the same interface between DTE and DCE, regardless of whether the VCC is configured for AAL5 or AAL3/4, as shown in Figure 17-15. The DTE must place the DTE SDU inside the AAL3/4 CPCS header and trailer, and then the DCE performs the appropriate function, depending upon whether the VCC is configured for AAL3/4 or AAL5. The DCE operates the same as in mode 1b for a VCC configured for AAL3/4, performing the AAL3/4 SAR on the AAL3/4 CPCS_PDU, as shown in the top part of Figure 17-15. The DCE must first extract the DTE_SDU from the AAL3/4 CPCS_PDU for a VCC configured to operate in AAL5, as shown in the bottom half of the figure. The net effect of these two transformations is that a mode 2 DCE can interoperate with a mode 1 DCE. The mode 2 DXI frame has a four-octet header and a four-octet FCS, which usually requires new hardware. Because the FCS is longer, the maximum DTE_SDU length can be larger. The 32-bit FCS used in the DXI is the same as that used for FDDI and AAL5.

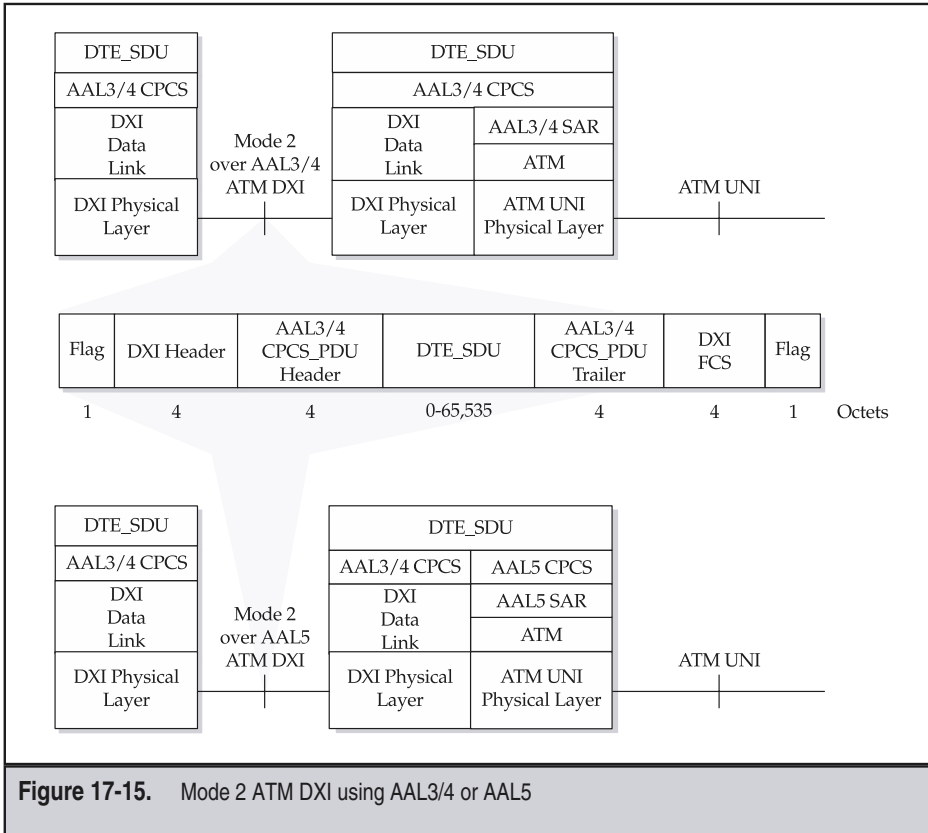


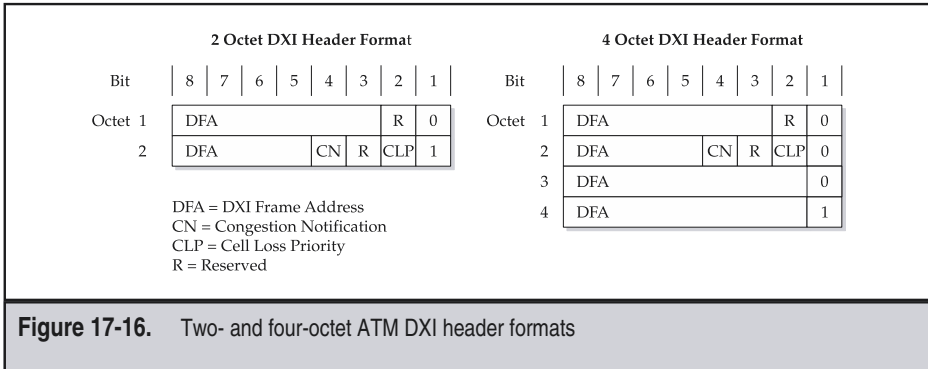
Figure 17-15. Mode 2 ATM DXI using AAL3/4 or AAL5

ATM DXI Header Formats

Figure 17-16 illustrates the details of the two- and four-octet DXI header structure. The DCE maps the DXI Frame Address (DFA) into the low-order bits of the VPI/VCI. The protocol maps the Congestion Notification (CN) bit from the last ATM cell of the PDU's Payload Type (PT) congestion indication (see Chapter 11). The DTE can set the CLP bit so that the DCE will in turn set the CLP bit in the ATM cell header with the same value, thus allowing the user to mark the cells from selected PDUs as a low loss priority. Note that a great deal of similarity exists between these DXI formats and the Frame Relay Service-Specific Convergence Sublayer (SSCS) formats covered earlier in this chapter.

Local Management Interface (LMI) Summarized

The DXI Local Management Interface (LMI) defines a protocol for the exchange of SNMP GetRequest, GetNextRequest, SetRequest, Response, and Trap messages between the



DTE and the DCE. The LMI allows the DTE to set or query (Get) the mode of the DXI interface as either 1a, 1b, or 2. The LMI also allows the DTE to set or query the AAL assigned on a per-VCC basis as indexed by the DXI Frame Address (DFA). A shortcoming of the current LMI is that the DCE does not communicate the ATM UNI status to the DTE.

Frame-Based User-Network Interface (FUNI)

The ATM Forum specified the Frame-Based User-Network Interface (FUNI) version 2.0 specification [AF FUNI 2.0] so that CPE without ATM hardware, such as many currently deployed routers, could interface to ATM networks with only minor software changes because it obviated the need for an expensive external ATM DXI converter. FUNI provides low-speed, WAN ATM access protocol rates of $N \times 64$ Kbps, DS1, and E1.

Figure 17-17 illustrates how frame-based CPE sends frames using the FUNI data link protocol to a network-based ATM switch, which then segments the frames into standard ATM cells using the AAL5 protocol. The same ATM switch reassembles cells transmitted by ATM CPE and delivers frames to the FUNI user. Thus, FUNI users communicate transparently across an ATM network with either other FUNI users (FUNI-to-FUNI) or ATM UNI users (FUNI-to-ATM UNI). It supports VPI/VCI multiplexing, SVC Signaling, network management, traffic policing, ATM OAM functions, and support for Variable Bit Rate (VBR) and Unspecified Bit Rate (UBR) traffic.

FUNI Frame Formats

The FUNI specification requires operation over AAL5 in DXI Mode 1a and makes operation over AAL3/4 in Mode 1b an option. It doesn't use Mode 2 but instead defines two additional modes, numbered 3 and 4. Table 17-3 summarizes the key attributes of the various FUNI modes. These include whether interoperable implementations must implement particular modes, the AAL employed, the number of octets composing the header, the maximum payload length, the number of cyclical redundancy check (CRC) octets in the trailer, the maximum number of user-definable connections, and the target interface speed for the specified mode. In the interest of brevity, this section covers only the details of the required Mode 1a mapping.

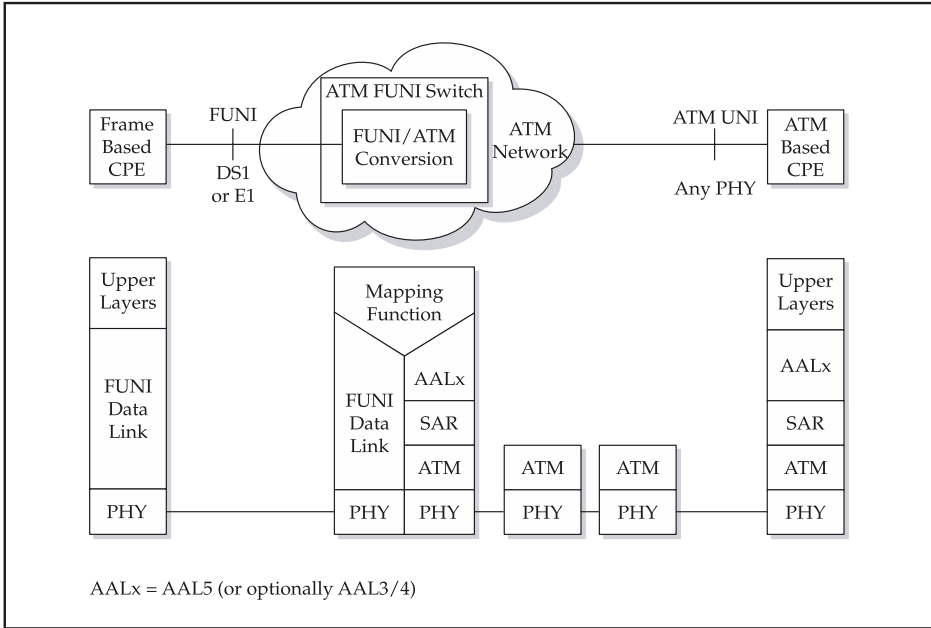


Figure 17-17. ATM Frame-Based User-Network Interface (FUNI)

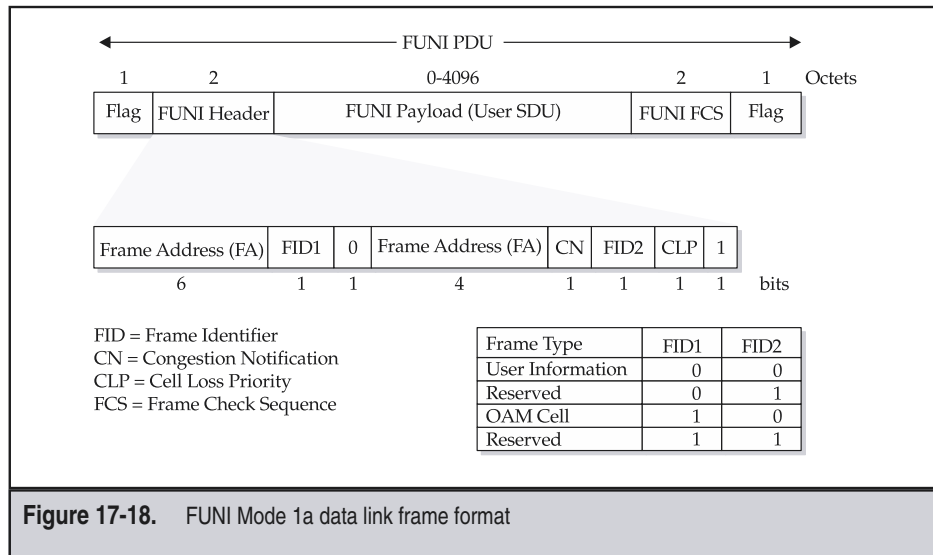
Mode	Implementation	AAL	Header Octets	Maximum Payload Octets	CRC Octets	Maximum User VCCs	Speed Range Required
1a	Required	5	2	4096	2	512	≤ DS1/E1
1b	Optional	3/4	2	4096	2	512	≤ DS1/E1
3	Required	5	2	9232	4	512	≤ DS1/E1
4	Optional	5	4	9232	4	16,777,216	≤ DS1/E1

Table 17-3. Frame-Based UNI (FUNI) Modes

A FUNI PDU has a header and a trailer delimited by HDLC flags as illustrated in Figure 17-18. The FUNI header contains a ten-bit frame address in the same bit positions as the ATM DXI Mode 1a header. Note, however, that the ATM DXI and ATM FUNI frame address fields map to different sets of VPI/VCI bits in the ATM cell header. The two Frame Identifier (FID) bits determine whether the FUNI frame contains either user information (i.e., data, signaling, or ILMI) or an OAM cell. The format of the FUNI header is identical to the ATM DXI Mode 1a header, except that the two bits reserved in ATM DXI are used by FUNI. The other bits operate identically to those in ATM DXI Mode 1a. The Congestion Notification (CN) bit maps to the Explicit Forward Congestion Indication (EFCI) payload type value in the ATM cell header as described in Chapter 11. The Cell Loss Priority (CLP) bit maps to the corresponding bit in the ATM cell header. The additional two bits (0 and 1) serve an address extension function analogous to that used in ATM DXI. The mapping of the FUNI frame address fields into the VPI/VCI bits in the ATM cell header allows FUNI to support a limited form of Virtual Path Connection (VPC) service. This mapping supports at most 16 VPCs or 1024 VCCs. The FUNI specification further limits the total required number of connections to 512 for Mode 1a operation.

FUNI versus ATM DXI and Frame Relay

Two key functional differences separate FUNI and DXI. FUNI provides improved access line utilization compared with cell-based access of an ATM DXI CSU/DSU. For example, at a typical packet size of 300 bytes, FUNI is 15 to 20 percent more efficient than ATM. The second difference is that FUNI supports $N \times 64$ Kbps rates, while the lowest speed supported by ATM DXI is DS1/E1.



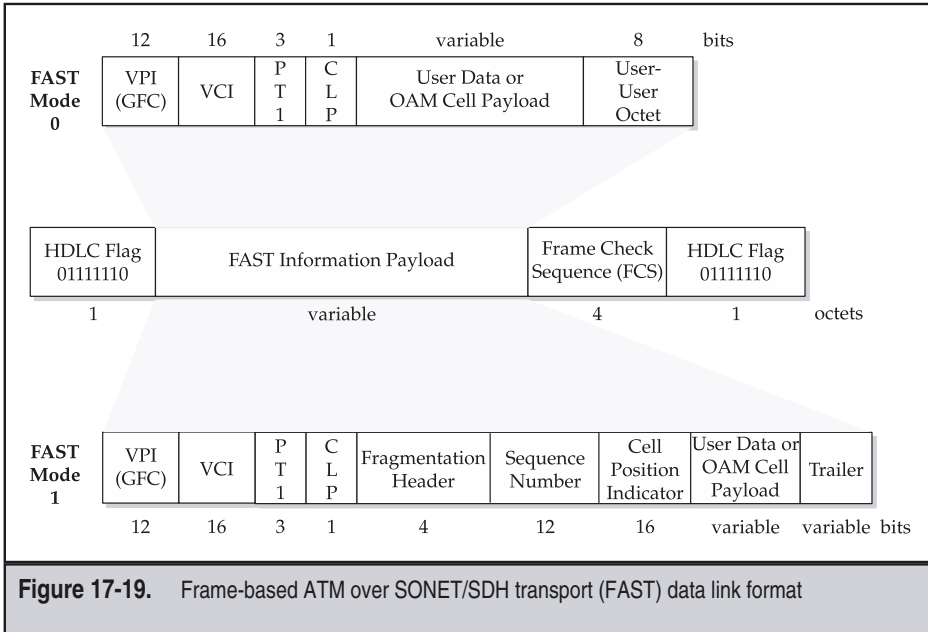
The observant reader will notice a great deal of similarity between FUNI and Frame Relay. A paper published by the Frame Relay Forum in 1995 [FRF FUNI] analyzed the relative advantages and disadvantages of FUNI versus Frame Relay, since these technologies overlap. The principal benefits of FUNI were the potential to better leverage ATM QoS capabilities; support longer frames using AAL5 instead of HDLC; and eliminate conversion between FR and ATM signaling, protocol encapsulation, and traffic parameters. On the other hand, Frame Relay had a large installed base of user and network equipment, as well as equipment and service available from multiple suppliers. In this case, the advantages of the newcomer, FUNI, were not compelling enough over the incumbent Frame Relay to cause a significant demand for FUNI, although there are some vendor implementations and service providers that offer it. The lack of market adoption occurred despite efforts in RFC 2363 to specify a more IP-friendly version of PPP over FUNI, as well as definition of support for multimedia applications [AF SAA-119].

Frame-Based ATM over SONET/SDH Transport (FAST)

In response to the market adoption of variable-length packets as a large portion of traffic, the ATM Forum specified a frame-based ATM over SONET/SDH Transport (FAST) specification [AF FAST] in 2000 that addressed many of the efficiency issues of cell-based transport of variable-length packets while preserving ATM OAM, signaling, addressing, and routing features. In contrast to DXI and FUNI, which were defined only at the UNI, FAST not only supported the UNI but also defined an NNI or trunk application. Furthermore, in principle FAST could operate at any standard SONET/SDH rate.

In terms of function, FAST is essentially a superset of FUNI; however, the format on the wire is different, as shown in Figure 17-19. The basic frame in the center of the figure is a simplified version of the PPP over HDLC framing as defined in RFC 1662, also known as Packet over SONET (POS), always starting and ending with an HDLC flag with error detection performed by a 32-bit FCS. FAST defines two modes of operation for the variable-length information payload field, as shown at the top and bottom of the figure. Mode 0 is the simplest, containing the first four octets of the ATM cell header (see Chapter 11), followed by a variable-length user data or ATM OAM cell payload, and ending with a one-octet user-user field. Mode 0 is capable of encapsulating either a single cell or a variable-length AAL5 SDU frame. The user-user octet is defined as the AAL5 CPCS_UU field (see Chapter 12) when operating in frame encapsulation mode.

On the other hand, as shown at the bottom of Figure 17-19, Mode 1 is more complex in support of several additional functions. As in mode 0, the first four octets of the ATM cell header begin the information field. The next two octets contain a fragmentation header and sequence number in support of fragmentation and reassembly of frames larger than that supported by the HDLC link layer. The cell position indicator (CPI) field is used only for the OAM cell payload type to place OAM cells in their proper relative position, which is necessary for ATM security and performance measurement functions that employ ATM OAM cells. Following the variable-length user data or OAM cell payload field is a trailer field that is used to transport the AAL5 PDU trailer, as described in ITU-T Recommendation I.363.5 (see Chapter 12) for applications that require use of the entire AAL5 PDU.



In summary, FAST mode 0 provides a means to transfer variable-length packets as efficiently as MPLS over POS in a simple manner, losing support for some ATM OAM and AAL5 functions. The more complex FAST mode 1 adds full support for these functions missing in mode 0, as well as a link layer means to support a very large MTU size. Although defined at both the UNI and NNI and supported by several vendors, FAST is most commonly used in a trunking application as a means to make the trunking efficiency for carriage of frame-based protocols like FR or IP more efficient without having to change out the entire network infrastructure. Transmission efficiency is an important consideration to service providers, particularly in places where they must lease circuits or where there is a shortage of capacity. Since the FAST protocol presents the appearance of a standard ATM layer functions to the higher-layer protocols, any of the ATM-based protocols can run over FAST. The ATM Forum also standardized a chip-level physical interface in support of frame-based ATM protocols so that FAST could be implemented cost-effectively [AF FBATM].

Frame-Based ATM Transport over Ethernet (FATE)

Frame-Based ATM Transport over Ethernet (FATE) was developed in response to a different driver than FAST. Whereas FAST strove to achieve high efficiency and full ATM functionality, the objective of FATE was to provide ATM capabilities across the cheapest LAN technology, namely Ethernet, even if it was somewhat inefficient in a local area net-

work [AF FATE]. Figure 17-20 illustrates the conceived application of FATE as a means of allowing end user PCs access to native ATM services when sharing an ATM over ADSL connection. The specification defined support for the entire AAL5 PDU, ILMI, and partial OAM support in either PVC or SVC mode. This protocol is not widely supported by vendors or service providers for several reasons. First, there is not a great deal of support on PCs for native ATM services, and that which exists is not easy to configure. It is possible that a serious shortcoming of FATE is that the native ATM services supported on the PC excluded AAL2 and AAL1, which could have been used to support voice and video over ATM quite efficiently over the bandwidth-limited ADSL access line, as described in Chapter 16.

MPLS-BASED SUPPORT FOR LINK LAYER PROTOCOLS

Chapters 12 and 13 introduced the work being done by the IETF Pseudo Wire Emulation Edge to Edge (PWE3) working group in support of a broad range of services over an MPLS or IP tunnel. Figure 17-21 illustrates a layered model specifically related to support of connection-oriented link layer services over an MPLS or IP tunnel using either a packet mode or cell mode encapsulation. The packet mode includes encapsulation and service support for FR, HDLC, and either the AAL5 SDU or the entire AAL5 PDU. As shown in the right-hand side of the figure, the PWE3 cell mode encapsulation is targeted to provide transport of ATM cells, or else a logical interconnection of ATM ports, where unassigned and idle cells are not transferred (see Chapter 11). Chapter 12 describes the ATM Forum-specified AAL5 PDU or native ATM cell over MPLS encapsulation, which is the basis of several proposals being considered in PWE3, along with some other alternatives.

In this section, we first describe some generic tunnel, pseudo-wire, and service emulation considerations unique to the connection-oriented services shown in Figure 17-21. What follows is a specific example of the payload encapsulation to illustrate some of these considerations. Finally, we give a summary of the transport of Frame Relay over MPLS as defined by the Frame Relay Forum and the MPLS Forum.

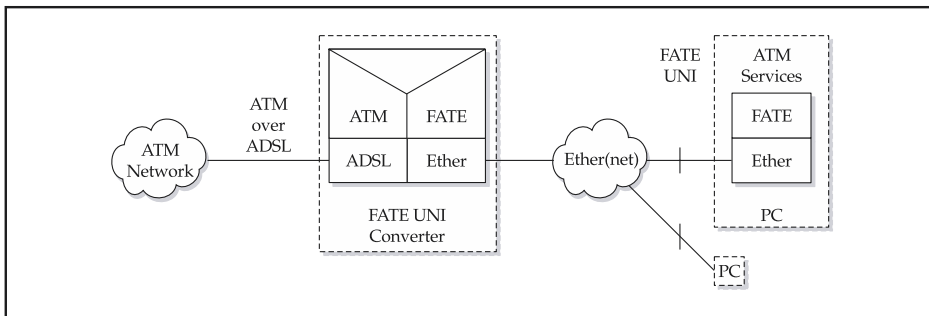


Figure 17-20. Frame-based ATM transport over Ethernet (FATE) target environment

Frame Relay	HDLC	AAAL5 SDU	AAAL5 PDU	ATM Cell	ATM Port
Packet Mode Pseudo Wire				Cell Mode Pseudo Wire	
MPLS or IP Tunnel					
Physical Medium					

Figure 17-21. MPLS support for link layer protocols

Pseudo-Wire and Service Emulation Considerations

The IETF PWE3 working group is chartered to document a number of general requirements and considerations in support of connection-oriented data services. As described in Chapter 12, an MPLS or IP tunnel must first be established between a pair of devices before any pseudo-wire-specific functions can occur. After a tunnel is established, pseudo-wire provider edge (PE) devices must implement some means to learn each other's capabilities and agree on a means for perform pseudo-wire functions, for example, encapsulation and sequencing, as well as status monitoring, checking, and reporting. These mechanisms may be explicit manual configuration, or a to-be-agreed-upon discovery and signaling protocol. ATM OAM and FR both have some specific requirements for support of status-related functions. Specifically, as detailed in Part 7 ATM has alarm and performance management requirements, while FR has a very specific set of status signaling requirements, as described in Chapter 7. These status-related functions should be capable of being done on an individual VC basis, or in the interest of efficiency for a trunking application on a collective basis. Pseudo-wire connections must have the same protocol type on each end. Furthermore, it is highly desirable for the PWE3 protocols to automatically detect pseudo-wire connections made to an incorrect destination or with inconsistent or incompatible parameters. Of course, PWE3 will also define management of the emulated services, for example configuration, provisioning, performance monitoring, fault management, and traceroute.

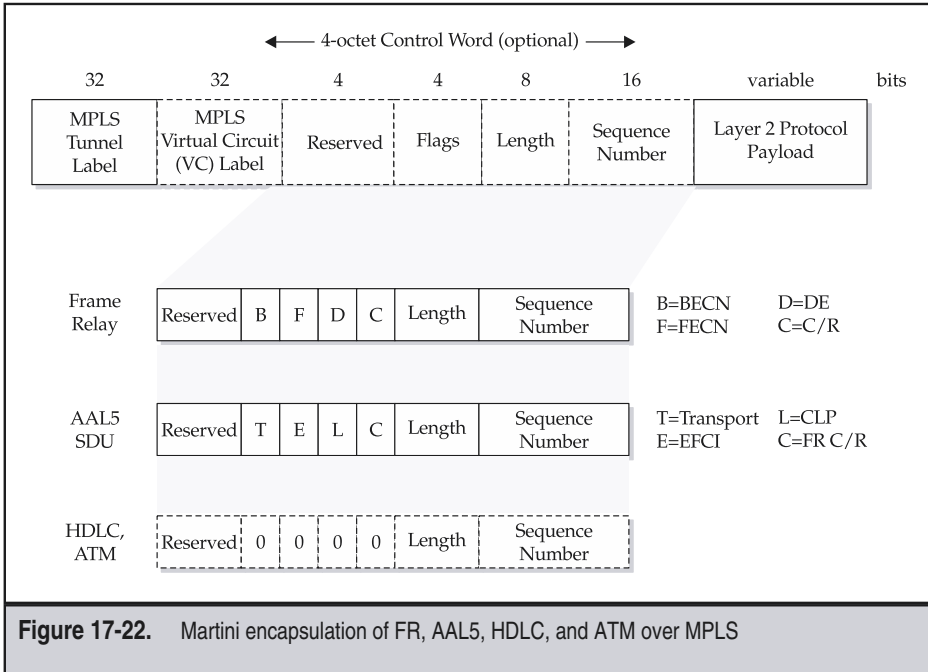
What is also interesting in the PWE3 working group charter and requirements scope are aspects that are not requirements. For example, service interworking is out of scope and should instead be performed by a native service processing function, as described in Chapter 12. Since PWE3 specifies the only encapsulation and signaling involved with establishment of the pseudo-wire, there can be no requirement that the quality of the emulated service be as high as that of the native service. Other considerations outside of the scope of PWE3 are security and traffic management.

Martini Encapsulation and Transport of FR, AAL5, ATM, and HDLC

Figure 17-22 illustrates the Martini encapsulation of FR, AAL5, HDLC, and ATM [Martini 02]. The top of the figure shows the generic format of the Martini encapsulation described in Chapter 12, along with an optional control word with reserved, flags, length, and sequence number fields. The MPLS tunnel label identifies the endpoint of the pseudo-wire, while the MPLS virtual circuit/connection (VC) label, sometimes also called a demultiplexing label, allows a single tunnel to support many VCs between tunnel endpoints, as described in Chapter 12. Tunnel endpoints use the control word if small packets need to be padded out to the minimum length of a link layer network (e.g., 64 octets in Ethernet), if the L2 protocol requires in-sequence delivery, or if other flags of the encapsulated L2 protocol need to be carried over the pseudo-wire. A value of zero in the sequence number field indicates that sequence numbering is not used. The control word is mandatory for encapsulation of FR and AAL5 Service Data Units (SDUs), with a specific definition applied to each of the four flag bits, as shown in the middle of the figure. For Frame Relay, the Backward and Forward Explicit Congestion Notification (BECN and FECN), Discard Eligible (DE), and Command/Response (C/R) bits are copied from the FR frame header (see Chapter 7). For AAL5 SDU encapsulation, the T-bit indicates whether the payload contains an AAL5 SDU or an ATM OAM cell. The E-bit carries the Explicit Forward Congestion Indication (EFCI) bit, while the L-bit carries the Cell Loss Priority (CLP) bit from the last ATM cell header in an AAL5 PDU, or the value from a single cell. The C-bit carries the FR C/R bit when FR/ATM service interworking [FRF.8.1] is implemented.

This encapsulation approach also defines support for carrying HDLC frames and one or more ATM cells in the variable-length layer 2 protocol payload field. If present, the flags in the control word are all zeros, as shown at the bottom of Figure 17-22. When used for HDLC or ATM, the control word length field supports padding to a minimum frame size of the underlying link layer network or supports sequencing. The entire HDLC PDU except flags is carried in the L2 payload field (see Chapter 7). When supporting ATM, the first four octets of each ATM cell header along with the 48-octet ATM cell payload are copied from each cell into the L2 payload field. The link layer MTU size and the amount of time the transmitter will wait to accumulate cells destined for the same pseudo-wire limit the number of cells that can be placed in a single packet.

The Martini draft specifies an extension to the Label Distribution Protocol (LDP) to communicate the binding between a VC label, overall VC parameters, and interface parameters specific to each VC to the other tunnel endpoint. The downstream unsolicited mode of LDP (see Chapter 14) is used to distribute these bindings for each of the two unidirectional tunnels that make up the pseudo-wire in support of a bidirectional connection-oriented protocol. It is possible for different VC labels to support different VC types within the same MPLS tunnel. The VC parameters include a 32-bit VC identifier, the VC type (e.g., DLCI, AAL5, ATM, HDLC), an indication of whether the VC label and/or control word are present, and a group identifier that allows many VCs to be associated with the same group. The interface parameters include MTU size, maximum number of concatenated ATM cells, and an optional interface description string for administrative convenience.



LDP label mapping release of a VC label is used to restart sequence numbering, or else indicate failure of one or more VCs to the transmitting tunnel endpoint. This basic mechanism is capable of supporting FR active bit status signaling and HDLC interface messaging (see Chapter 7), as well as providing basic ATM OAM alarm indication signal (AIS) functionality (see Chapter 28). The VC group identifiers allows the wildcard withdrawal of multiple VC labels in support of a major interface or card failure and therefore more efficiently accomplishes this status reporting in bulk. The Martini encapsulation also does not fully support ATM OAM functions in the AAL5 mode, which is a reason that the PWE3 working group is looking at other approaches for support of connection-oriented protocols over MPLS and IP tunnels.

This encapsulation is relatively efficient, but it does require mapping of the native protocol identifier (e.g., DLCI, VPI/VCI) to an MPLS label and transformation at the bit level of header fields into the flags field. These can be processing-intensive real-time operations if done in software, but they are readily amenable to hardware implementation. At the time of this writing, the IETF was considering the Martini drafts as a starting point for PWE3 standards. In addition, a number of vendors had implemented the Martini encapsulation and LDP signaling transport extensions, as evidenced by several interoperability testing events. In fact, as described in the next section, another emerging standard for transporting FR over MPLS is compatible with the Martini encapsulation.

FR over MPLS Network Interworking

Figure 17-23 shows the context and a summary of the protocol operation for the transport of FR service over an MPLS LSP, as defined by the Frame Relay Forum and the MPLS Forum [FRMPLS] in a draft specification. The function of the FR-MPLS provider edge (PE) device is to carry one or more bidirectional FR network-network and/or user-network interface (UNI and NNI) virtual connections over a pair of unidirectional MPLS LSPs, as shown in the middle of the figure. The UNI and NNI Frame Relay service interfaces should be unaware that the FR traffic has been carried by the MPLS network. That is, MPLS transports FR, or in the terminology introduced earlier for FR and ATM interworking, this scenario supports FR-MPLS network interworking. At the time of writing, this work primarily addressed the encapsulation and user plane interworking functions. Support for PVC status management, traffic management, QoS, and details of the fragmentation procedure were identified as future work.

The protocol stack diagrams that follow the elements at the bottom of Figure 17-23 illustrate their function, in a manner similar to that used to describe FR/ATM interworking earlier in this chapter. Of course, the FR (switching) node or DTE implement the FR NNI or UNI protocol, respectively, as shown at the left and right. These FR devices interface to the FR-MPLS PE, which terminates the FR UNI or NNI protocol stack and implements an interworking function (IWF) that translates FR into the protocol stack for MPLS

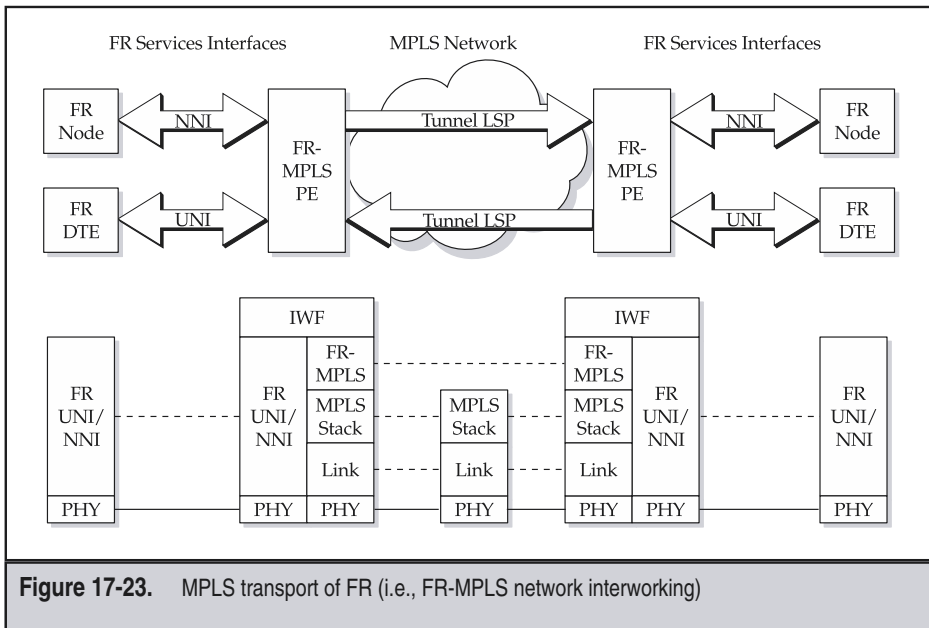


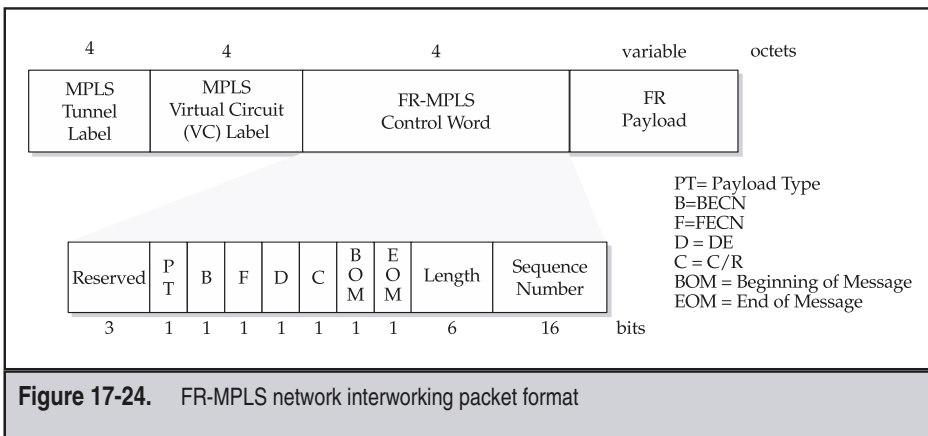
Figure 17-23. MPLS transport of FR (i.e., FR-MPLS network interworking)

transport of FR. This contains the FR-MPLS control, along with an MPLS label stack containing a topmost label corresponding to a unidirectional tunnel and a bottommost label corresponding to a VC associated with the FR DLCI. The FR-MPLS PE then places this packet into the link layer PDU for transfer by the MPLS network to the destination PE, where the reverse process occurs.

Figure 17-24 illustrates the FR-MPLS network interworking packet format. Comparing this figure to the Martini encapsulation shown in Figure 17-22, we see that the formats are almost identical. The only differences are that the VC label is required, and that the FR-MPLS network interworking defines a bit in the reserved field and uses the two high-order bits of the length field for the purpose of link layer fragmentation and reassembly. When the Payload Type (PT) bit is zero, the payload is user data; otherwise, the payload is network data. The BOM and EOM bits are proposed for use in link layer fragmentation and reassembly and reduce the length field of the Martini encapsulation. However, a six-bit length field is still sufficient to handle any length packet transferred over Ethernet, which has a 64-octet length minimum frame size.

REVIEW

This chapter covered the ATM and MPLS protocol support for connection-oriented wide area data networking, with a particular focus on Frame Relay. The coverage began by defining the basic concepts of physical and logical access, interworking, and trunking. The text then covered specific examples of Frame Relay trunking over ATM using the network interworking protocol, true service interworking between Frame Relay and ATM, and access to SMDS via ATM to illustrate these concepts. Next, the chapter covered the ATM Forum's specifications for low-speed frame-based interfaces defined by the ATM Data Exchange Interface (DXI) and Frame-Based UNI (FUNI) protocols, which make



ATM available to existing CPE cost effectively via external converters and a software-defined protocol, respectively. We then summarized the high-speed frame-based ATM over SONET/SDH transport (FAST) and transport over Ethernet (FATE), which address the cell tax issue of ATM when carrying variable-length Internet packets. The chapter continued with an overview of the IETF pseudo-wire emulation efforts, beginning with a discussion of requirements and scope. We then described the Martini encapsulation specific to FR, AAL5, ATM, and HDLC. Finally, the text summarized a specific application of FR trunking over MPLS.

CHAPTER 18

ATM and MPLS Support for LAN Protocols

This chapter begins with a summary of the first standard methods defined to support one or more local area network (LAN) protocols over an ATM network. The multiprotocol encapsulation over ATM standard was driven by the economic need to find a means to avoid the cost of a VCC per routed or bridged protocol (such as IP or Ethernet), especially when an enterprise often had multiple layer 3 protocols in use at many sites. These protocols are very simple and are the most widely deployed. The chapter then moves on to the subject of more feature-rich support for LANs. As described in Chapter 9, Ethernet has won the battle in the late 1990s for the next-generation LAN protocol. However, for a period of time and still in some deployments ATM's LAN Emulation (LANE) protocol provides a means to seamlessly interconnect legacy LANs with high-performance local area ATM networks. Because Ethernet over MPLS attempts to solve a similar problem of supporting a connectionless LAN protocol, in particular Ethernet, using the connection-oriented MPLS protocol, the coverage focuses on lessons learned from the history of ATM support for Ethernet. The chapter concludes with a summary of current directions and approaches being considered by the IETF, as well as a number of vendors and service providers for supporting Ethernet over MPLS or IP tunneled networks in support of a virtual private LAN service (VPLS). The description includes the encapsulation and signaling protocols, the overall architecture, and several examples of how such a service could be used.

MULTIPROTOCOL ENCAPSULATION OVER AAL5

First defined in 1993 in RFC 1483 and superseded by IETF RFC 2684 in 1999, this standard defines the specific formats for routing or bridging a number of commonly used protocols over ATM Adaptation Layer 5 (AAL5) using either protocol encapsulation or VC multiplexing. *Protocol encapsulation* provides the capability to multiplex many different protocols over a single ATM Virtual Channel (VC, which is the commonly used shorthand name for Virtual Channel Connection [VCC]). The *VC multiplexing* method assumes that each protocol is carried over a separate ATM VC. Both of these encapsulation methods utilize AAL5. This section covers first the protocol encapsulation method and then the VC multiplexing method.

Protocol Encapsulation

Protocol encapsulation operates by prefixing the Protocol Data Unit (PDU) with an IEEE 802.2 Logical Link Control (LLC), as described in Chapter 9. Hence, RFC 2684 calls this *LLC encapsulation*. In some cases, the IEEE 802.2 Subnetwork Attachment Point (SNAP) header must also be present. The LLC/SNAP header identifies the PDU type and allows two nodes to multiplex different protocols over a shared LAN MAC sublayer. This method was initially designed for public network or wide area network environments where a customer premises device would send all protocols over a single VCC, because some carrier pricing structures favored a small number of PVCs. In other words, this capability saves money by sending multiple protocols over one VCC, avoiding the

administrative complexity and expense of ordering a separate VCC for each protocol. Note that all packets get the same ATM QoS, because they share a single VCC.

Figure 18-1a illustrates LLC encapsulation by routers, showing a network of three routers, each multiplexing separate Ethernet and Token Ring LANs over a single VCC interconnecting the locations. The routers multiplex the Ethernet and Token Ring PDUs onto the same VCC using the encapsulation described in the next section. The drawing in Figure 18-1b illustrates how bridges multiplex PDUs from Ethernet and Token Ring in interfaces to yield a bridged LAN. The Ethernet bridges use a spanning tree composed of a subset of the ATM VCCs at any one point in time, which the example illustrates by a dashed line to indicate the unused VC link in the spanning tree where the center bridge assumes the role of the spanning tree root. Note that token ring bridges may make more efficient use of the ATM VCCs through source routing as described in Chapter 9.

LLC Encapsulation for Routed Protocols

We now describe LLC encapsulation for routed protocols. Figure 18-2a depicts the LLC/SNAP encapsulated routed ISO protocol payload structure. The three octets of the LLC header contain three one-octet fields: a Destination Service Access Point (DSAP), a Source Service Access Point (SSAP), and control. The routed ISO protocol is identified by a one-octet NLPID field that is included in the protocol data. NLPID fields include, but are not limited to, SNAP, ISO CLNP, ISO ES-IS, ISO IS-IS, and Internet IP. The values of these octets are shown in hexadecimal notation as X'zz', where z is a hexadecimal digit representing four bits. For example X'1' corresponds to a binary '0001', and X'A' corresponds to a binary '1010'. Figure 18-2b depicts the payload for a routed non-ISO protocol, specifically showing the example for an IPv4 PDU. The three-octet SNAP header follows the LLC and contains a three-octet Organizationally Unique Identifier (OUI) and the two-octet Protocol Identifier (PID). RFC2684 defines support for other routed (that is, layer 3) protocols, such as ISO CLNP and IS-IS, through the use of different network layer PIDs (that is, NLPID).

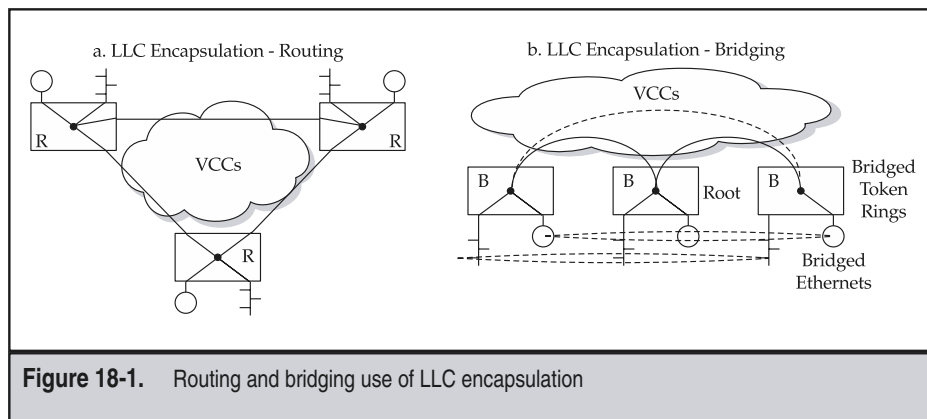
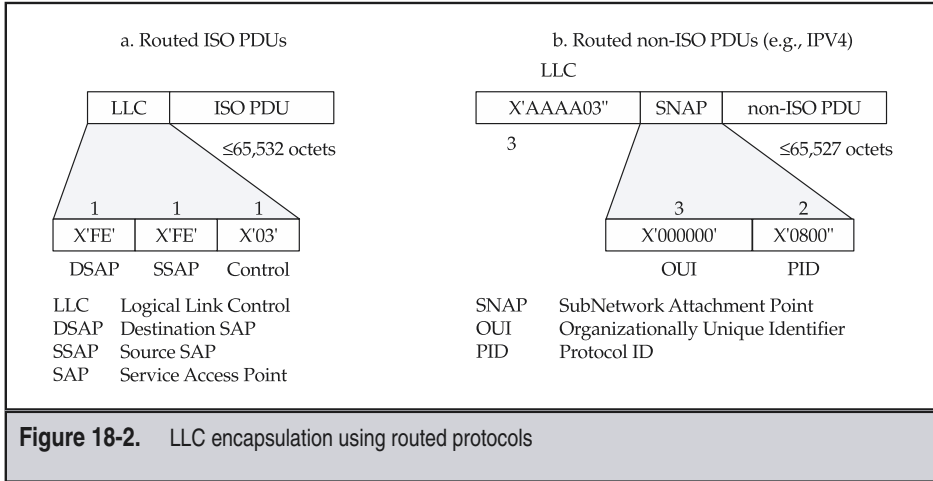


Figure 18-1. Routing and bridging use of LLC encapsulation



LLC Encapsulation for Bridged Protocols

The LLC encapsulation method also supports bridging for LAN and MAN protocols. The SNAP header identifies the type of bridged medium and whether the original LAN Frame Check Sequence (FCS) is included with the PDU, as illustrated in Figure 18-3. The LLC header identifies a non-ISO PDU as before. The OUI field identifies an 802.1 identification code. The PID field then identifies the actual protocol. The remaining fields are either padding or the actual LAN PDU. Bridgeable protocols include 802.3 Ethernet, 802.4 Token Bus, 802.5 Token Ring, Fiber Distributed Data Interface (FDDI), and 802.6 DQDB, as shown in the figure.

VC-Based Multiplexing

The second method defined in RFC 2684 for carrying multiple protocols over ATM is through VC-based multiplexing, which supports a single protocol per virtual connection. In other words, the VCs are multiplexed, rather than the protocols themselves as done in the protocol encapsulation method. With this method, different protocols can have different bandwidth allocations and QoS, unlike with the multiprotocol encapsulation method, at the potential expense of additional administrative complexity and per-VC charges.

Figure 18-4a illustrates the VC multiplexing concept for routed protocols, showing a separate VCC connecting the routing point for the attached Ethernet and Token Ring LANs. Figure 18-4b illustrates the same situation for bridged protocols, again requiring twice as many VCCs as for protocol encapsulation. In the example, the source-routed bridged Token Ring load balances across all VCCs, while the spanning tree Ethernet does not use the VCC indicated by a dashed line. Comparing this to Figure 18-1, observe that the only difference is the use of one VCC for each protocol (i.e., Ethernet and Token Ring) that is being routed or bridged versus one VCC between each pair of routers or bridges.

	LLC	OUI	PID	PAD			LAN FCS
a. 802.3	X'AAAA03'	X'0080C2'	X'0001' X'0007'	S'0000'	MAC Destination Address	MAC Frame *	If PID X'0001'
	LLC	OUI	PID	PAD			LAN FCS
b. 802.4	X'AAAA03'	X'0080C2'	X'0002' X'0008'	X'000000'	Frame Control	MAC Destination Address	MAC Frame * If PID X'0002'
	LLC	OUI	PID	PAD			LAN FCS
c. 802.5	X'AAAA03'	X'0080C2'	X'0003' X'0009'	X'0000xx'	Frame Control	MAC Destination Address	MAC Frame * If PID X'0003'
	LLC	OUI	PID	PAD			LAN FCS
d. FDDI	X'AAAA03'	X'0080C2'	X'0004' X'000A'	X'000000'	Frame Control	MAC Destination Address	MAC Frame * If PID X'0004'
	LLC	OUI	PID	PAD			LAN FCS
e. 802.6	X'AAAA03'	X'0080C2'	X'000B'	Common PDU Header	MAC Destination Address	MAC Frame *	Common PDU Trailer

* Remainder of
MAC frame

Figure 18-3. LLC encapsulation using bridged protocols

The multiplexed PDU payload is devoid of the LLC and SNAP protocol identifiers used in protocol encapsulation, resulting in less overhead, less processing, and higher overall throughput. This method is designed for environments where the user can dynamically create and delete large numbers of ATM VCCs in an economical fashion, which occurs in private ATM networks or ATM SVC networks. Routed protocols can make use of the entire 65,535-octet AAL5 CPCS PDU. Bridged protocols have the same

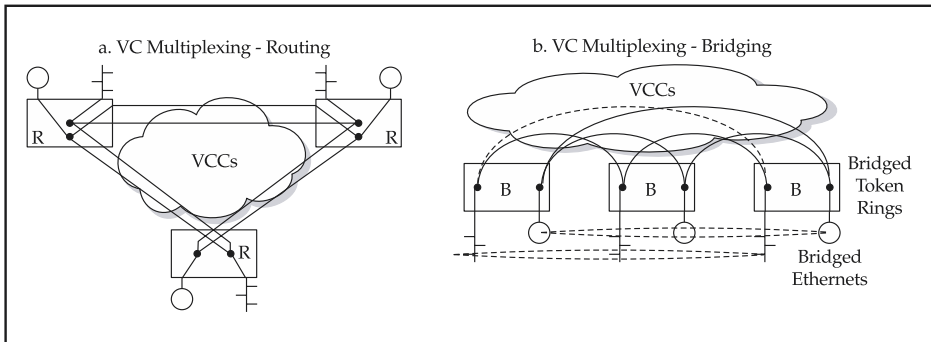


Figure 18-4. Routing and bridging usage of VC multiplexing

format as shown in Figure 18-3 but do not have the LLC, OUI, or PID fields. The use of LAN FCS in the bridged protocols is implicitly defined by association with the VCC by configuration.

Considerations in the Selection of Multiplexing Method

Either of the two types of multiplexing methods, LLC encapsulation or VC multiplexing, can be used with PVCs and SVCs. The method is selected as a configuration option for PVCs. SVCs use information elements in the signaling protocol for the two routers to communicate whether to employ LLC/SNAP protocol encapsulation or VC multiplexing. Signaling also indicates, when using VC multiplexing, whether the original LAN FCS is carried in the PDU.

If your application is IP-based, then the choice of the RFC 2684 encapsulation method can have a profound impact on efficiency. Recall from Chapter 8 that the minimum length TCP/IP version 4 packet for a stand-alone acknowledgment is exactly 40 octets in length. Although TCP attempts to “piggyback” acknowledgments on packets headed in the reverse direction, the experience in real-world networks is that approximately one third of the packets in TCP/IP networks are exactly 40 bytes long. Recall from Chapter 12 that AAL5 adds 8 octets of overhead, along with an optional padding field to extend the AAL5 PDU to exactly fit within an integer number of cells. If the network designer selects VC multiplexing, then a 40-octet TCP/IP packet and the AAL5 overhead exactly fit into one cell, as illustrated in Figure 18-5a. On the other hand, if the network designer chooses LLC encapsulation for nonrouted ISO PDUs (i.e., IP), the router adds an additional eight octets of LLC/SNAP overhead. The consequence is shown in Figure 18-5b, where these additional eight octets require AAL5 to use two cells instead of one to carry the commonly encountered minimum size 40-octet TCP/IP packet. This results in an ATM payload utilization of approximately 58 percent (that is $(40 + 8 + 8)/96$) when using protocol encapsulation, versus 100 percent utilization achieved when employing VC multiplexing. Of course, the utilization improves for TCP/IP packets of greater length, a subject that we cover in greater depth in Part 8. Therefore, if your network protocol is IP, then unless the economics of multiple VCCs in your ATM network overcome the reduction in efficiency due to protocol encapsulation, seriously consider using VC multiplexing over multiple parallel VCCs.

RFC 2684 also added support for virtual private networks (VPNs) to the protocol support defined in RFC 1483. The encapsulation applied to a VPN implemented over an

a. VC Multiplexing			b. Protocol Encapsulation						
Cell Header	TCP/IP ACK	AAL5 Trailer	Cell Header	LLC/SNAP	AAL5 Trailer	Cell Header	TCP/IP ACK	AAL5 PAD	AAL5 Trailer
5	40	8	5	8	32	5	8	32	8

Figure 18-5. TCP/IP efficiency using VC multiplexing and protocol encapsulation

ATM subnet using the 7-octet VPN identifier defined in RFC 2685. The vision was that this identifier contains an OUI corresponding to the VPN service provider, who then assigns a VPN indices to customers. The VPN identifier precedes the LLC encapsulation header in encapsulation mode. In VC multiplexing mode, the VPN identifier may precede the LAN frame; however, it can be configured administratively for PVCs or dynamically signaled with SVCs. Although this VPN capability has been standardized, it is not widely deployed.

ATM FORUM LAN EMULATION (LANE)

The ATM Forum defined LANE in 1995 [AF LANE 1.0] in order to provide an interoperable transition from legacy Ethernet and Token Ring networks to ATM. Prior to LANE, legacy LAN and WAN protocols (Ethernet, Token Ring, FDDI, and so on) required proprietary conversion devices to benefit from ATM. In the mid-1990s, LANE empowered LAN designers with a means to reap the benefits of high-capacity, scalable, bandwidth-controlled, guaranteed quality ATM networking while preserving the best elements of their legacy LAN infrastructure. A second version of the LANE specification in 1997 [AF LANE 2.0] added support for QoS, LLC multiplexing, support for multicast, and support for the ATM Forum's Multiprotocol over ATM (MPOA) specification. Although LANE had a period of rapid growth in the latter half of the 1990s, Ethernet won the LAN marketplace, as discussed in Chapter 10. This occurred because of the addition of QoS and virtual LAN (VLAN) features to basic Ethernet, along with the mass market economics of Ethernet. Therefore, we only summarize the context and important protocol aspects of LANE in this section as background to a discussion on Ethernet over MPLS, because a number of challenges are similar to those facing Ethernet over MPLS. Foremost among these challenges are supporting a connectionless service over a connection-oriented protocol, minimizing configuration complexity, emulation of a broadcast medium using point-to-point connections, and the complexity of supporting bridging protocols and virtual LANs in a shared service provider infrastructure.

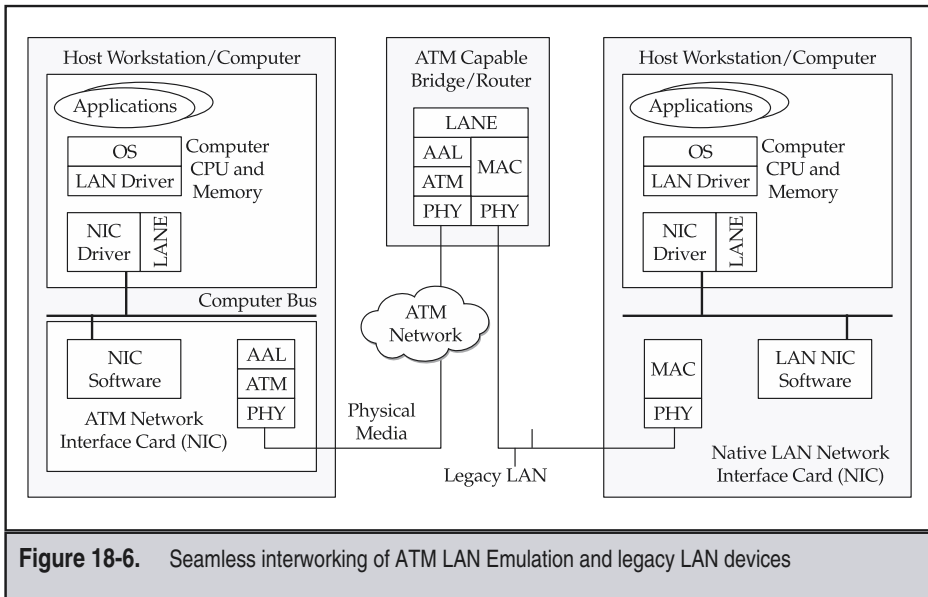
Hardware and Software in an Emulated LAN

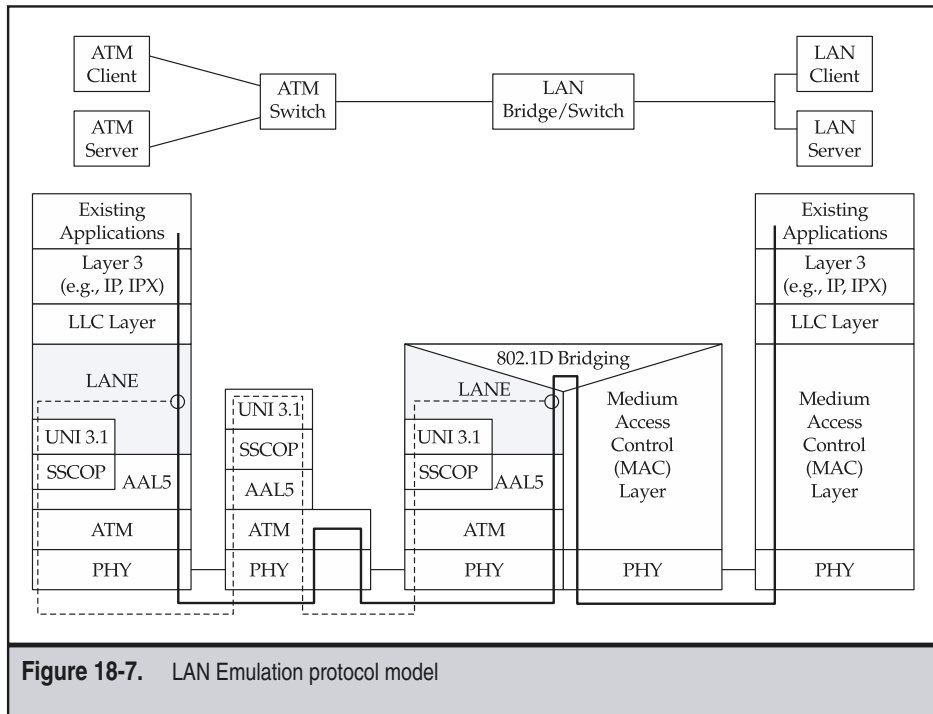
LANE improves upon the overall throughput of a shared-medium LAN by combining a broadcast server and fast connection switching. The LANE specification also defined how a workstation with an ATM NIC card and LANE software automatically joins an existing LAN to achieve plug and play operation by supporting the standard Ethernet spanning tree or Token Ring source route bridging. LANE also defined a form of virtual LANs, which allowed a network administrator to flexibly assign users to different virtual workgroups, but this was rendered commercially obsolete by the widespread adoption of the incompatible IEEE 802.1Q VLAN standard.

LANE enabled computers running an application on a legacy LAN like Ethernet to directly communicate with the same application on an ATM-enabled computer or server via a high-performance ATM network, as shown in Figure 18-6. These applications ran essentially unchanged using the existing software device driver interfaces on a workstation with an ATM network interface card (NIC), as shown in the left-hand side of the figure.

The computer runs a set of applications on a specific operating system (OS), which has particular LAN driver software, for example, the Microsoft Network Driver Interface Specification (NDIS). The NIC card (or the OS) provides LAN Emulation software that interfaces to the ATM NIC card via NIC driver software in the host. The cells from the ATM NIC traverse an ATM network to an ATM capable bridge/router, which maps to a legacy LAN media access control (MAC) and physical layer. The bridge/router interfaces to the host on the right-hand side of the figure, which has a native LAN NIC connected to a legacy LAN, for example, a 100 Mbps Ethernet. The LAN NIC in this workstation interfaces to LAN NIC driver software in the host, which provides an identical interface to the operating system's LAN driver software. Hence, applications running on the ATM-NIC and native LAN-NIC networked computers see no difference in terms of software functions.

Figure 18-7 shows the two types of LANE protocol data flows: a signaling connection shown by the dashed line and a data path connection shown by the solid line. The signaling connection sets up an SVC for the data-direct path between the ATM client or server and the LAN bridge or switch for each extended data flow between computers. Starting on the left-hand side, an ATM host or server runs an existing application and networking protocol (such as IP or IPX) that interfaces using the LLC protocol implemented in the host's LAN driver software. Note how the LANE software provides the same LLC interface to the network layer that Ethernet-attached hosts and servers on the right-hand side of the figure do. Moving to the right, an ATM network interconnects ATM clients and servers. In the user plane, the ATM switch operates only on the ATM cells. As indicated by the dashed line, the ATM switch operates in the control plane to dynamically establish and release connections. In the upper-middle portion of the figure is a LAN router,





bridge, and/or switch with an ATM interface and a LAN interface. This device terminates both the user and control plane components of the ATM-based LANE protocol and converts the frames to the MAC sublayer for transmission over a legacy LAN like Ethernet. On the right-hand side of the figure, the applications running on the native LAN workstation see the same LLC and layer 3 networking protocol that the ATM-empowered workstation on the left does.

The LANE 2.0 specification defines a software interface for network layer protocols identical to that of existing LANs that encapsulate user data in either an Ethernet or Token Ring MAC frame. LANE does not emulate the actual media access control protocol of a particular LAN concerned (i.e., CSMA/CD for Ethernet or token passing for 802.5). Instead, LANE defines three servers that clients access over a number of ATM connections designated for specific control and data transfer purposes. LANE does not directly define support for FDDI; however, devices readily map FDDI packets into either Ethernet or Token Ring using existing translation bridging techniques. Because all Ethernet standards use the same MAC packet formats, they map directly into LANE Ethernet or Token Ring formats and procedures. As described in the preceding text, LANE literally bridges ATM and LANs by interworking at the media access control (MAC) layer, which provides device-driver interfaces such as Open Data-Link Interface (ODI) and Network Driver Interface Specification (NDIS) to higher-level applications.

LANE Components and Connection Types

Figure 18-8 illustrates how virtual channel connections interconnect the following four logical components in the LANE specification:

- ▼ LAN Emulation Client (LEC)
- LAN Emulation Configuration Server (LECS)
- LAN Emulation Server (LES)
- ▲ Broadcast and Unknown Server (BUS)

Figure 18-8 also illustrates the control and data virtual channel ATM connections between LANE components via directed arrows. The figure shows the following control connections as dashed lines:

- ▼ A bidirectional, point-to-point configuration-direct VCC set up by the LEC to the LECS.
- A bidirectional, point-to-point control-direct VCC set up by the LEC to the LES.
- ▲ A unidirectional control-distribute VCC set up from the LES back to the LEC. Typically, this is a point-to-multipoint connection, but it may be implemented as a set of unidirectional point-to-point VCCs.

Figure 18-8 shows the following LAN emulation data connections as solid lines:

- ▼ A bidirectional, point-to-point data-direct VCC set up between two LECs to exchange data.
- A bidirectional, point-to-point Multicast Send VCC set up by the LEC to the BUS.
- ▲ A unidirectional VCC Multicast Forward VCC set up from the BUS to the LEC. Typically, this is a point-to-multipoint connection with each LEC as a leaf, but it may also be a set of unidirectional point-to-point connections from the BUS to each served LEC.

The next section provides a narrative of how these components use these connections to perform initialization, join an emulated LAN, resolve MAC and ATM addresses, transfer MAC frames, and emulate broadcast functions.

Summary of LANE Operation

The LEC runs in every ATM end and intermediate system (e.g., a host, server, bridge, or router) that provides a standard LAN service interface to higher-layer interfaces. An LEC performs data forwarding, address resolution, and other control functions in this role. ATM Network Interface Cards (NICs) in hosts and servers, as well as ports on switches,

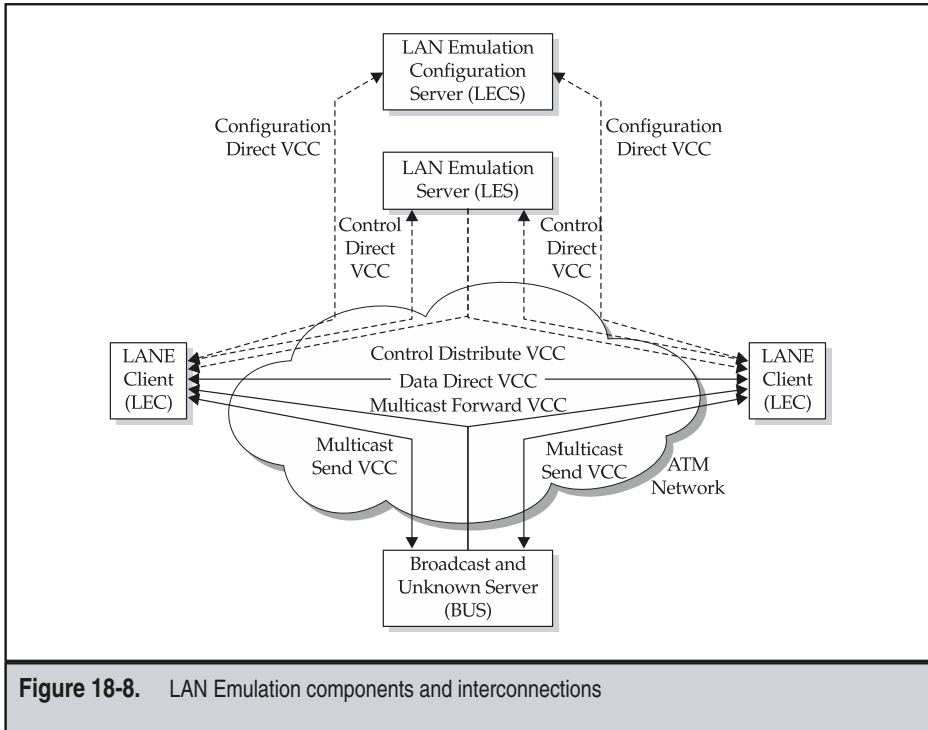


Figure 18-8. LAN Emulation components and interconnections

bridges, and routers, are examples of LEC implementations. A unique ATM address identifies each LEC, which the LANE protocol associates with one or more MAC addresses reachable through its ATM UNI. The LEC address is either preconfigured into the LEC or dynamically discovered via the Integrated Local Management Interface (ILMI) protocol. A LAN switch or bridge implementing an LEC dynamically associates all the MAC addresses reachable through its LAN ports to a single ATM address.

An LEC joins an emulated LAN by first connecting to the LECS. Logically, one LECS serves all clients within an administrative domain. The LEC determines the LECS address by either using the ILMI procedure, defaulting to a well-known LECS SVC address, or else using a well-known permanent connection to the LECS (e.g., VPI=0, VCI=17). Next, the LEC sets up a configuration-direct connection control VCC to the LECS. The LECS informs the LEC of information required for entry into its target ELAN through use of a configuration protocol. This information includes the controlling LES ATM address, the emulated LAN type, maximum frame size, and an ELAN name. The LECS is the place where virtual LANs are implemented.

A single LES implements the address registry and address resolution server for a particular emulated LAN. After the LEC obtains the LES address from the LECS, it may clear the configuration-direct VCC to the LECS, because it requires no further configuration information. The LEC next sets up a control-direct VCC to the LES using standard signaling procedures. The LES assigns the LEC a unique LEC Identifier (LECID). The LEC registers its MAC address and ATM address with the LES. It may optionally also register other MAC addresses for which it acts as a proxy, for example, other reachable MAC addresses learned by a spanning tree bridge.

The LES then adds the LEC to the point-to-multipoint control-distribute VCC. The LEC employs the control-direct and -distribute VCCs for the LAN Emulation Address Resolution Protocol (LE_ARP). The LE_ARP response message returns the ATM address corresponding to a particular MAC address. The LES responds to an LE-ARP directly to the LEC if it recognizes this mapping; otherwise, it forwards the request on the point-to-multipoint control-distribute VCC to solicit a response from a LEC that recognizes the requested MAC address. LANE uses this procedure because an LES may be unaware of a particular MAC address, because the address is “behind” a MAC bridge that did not register the address.

An LEC may respond to an LE_ARP because it is acting as a proxy for that address on the control-direct VCC to the LES. The LES then forwards this response back either to the requesting LEC or, optionally, on the point-to-multipoint control-distribute VCC to all LECs. When the LES sends the LE_ARP response on the control-distribute VCC, then all LECs learn and cache the particular address mapping. This aspect of the protocol significantly reduces the transaction load on the LES.

An LEC uses this LE_ARP mechanism to determine the ATM address of the BUS by sending an LE_ARP for the all-ones MAC broadcast address to the LES, which responds with the BUS's ATM address. The LEC then uses this address to set up the point-to-point multicast send VCC to the BUS. The BUS, in turn, adds the LEC to the point-to-multipoint multicast forward VCC. Having completed initialization, configuration, and registration, the LEC is now ready for the data transfer phase.

The BUS is a multicast server that floods unknown destination address traffic and forwards multicast and broadcast traffic to clients within an Emulated LAN (ELAN). An emulated LAN may have multiple BUSs for throughput reasons, but each LEC transmits to only one BUS. Typically, the association of the broadcast MAC address (i.e., “all ones”) with the BUS is preconfigured into the LES. An LEC can send frames to the BUS for delivery to the destination without setting up a data-direct VCC to the destination LEC. However, the LANE 2.0 specification limits the rate at which an LEC may send frames to the BUS to prevent broadcast storms.

The data transfer phase begins in an end system when a higher-layer protocol generates a packet in a NIC, or else when an ATM LANE port in a bridging device receives a frame from another LAN (or LANE) port. For the first frame received with a particular destination MAC address, the LEC doesn't know the ATM address needed to reach the destination. In order to resolve this, the LEC formulates an LE_ARP packet and sends it to the LES. While waiting for a response to the LE_ARP, the LEC may forward packets to the

BUS, which floods the packet to all LECs over the multicast distribute connection. Alternatively, the LEC could buffer the packets and wait for connection establishment, but this increases response time to perform the other functions that will be described. Flooding ensures that no packets are lost, and also reaches MAC addresses “behind” bridging devices that don’t register all of their MAC addresses with the LES. Many network protocols are very sensitive to loss, and hence the flooding step ensures good performance. When the LEC receives an LE_ARP response indicating the target ATM address, then it sets up a data-direct VCC. The LEC then sends a ready indication frame over the data-direct VCC and not over the multicast-distribute VCC. The use of direct VCCs makes efficient use of the underlying ATM network capabilities. Of course, ATM switches in the LANE network must be capable of processing a large number of SVC requests per second in order for LANE to realize this efficiency.

Before an LEC uses the direct path, it may choose to utilize the optional LANE flush procedure to ensure in-sequence packet delivery. The flush procedure ensures that packets previously sent to the BUS arrive at the destination prior to sending any packets over the data-direct VCC. In order to accomplish this, the LEC stops sending packets over the BUS and sends a flush packet over the BUS multicast-send VCC. The LEC buffers any packets while awaiting a flush acknowledgment from the destination LEC. Once the destination LEC receives the flush packet, it generates a flush acknowledgment on the control-direct VCC to the LES. The LES distributes the flush ACK to at least the source LEC, and now the LECs can exchange frames over the newly established data-direct VCC.

If a data-direct VCC already exists to the destination LEC, the source LEC may choose to use this same data-direct connection, thereby utilizing connection resources efficiently and reducing latency. Furthermore, a pair of LECs may set up parallel connections to support application with different QoS requirements. If the LEC receives no response to the LE_ARP, then it continues sending packets to the BUS. The LEC periodically re-sends LE_ARPs packets in an attempt to solicit a response. Typically, once the BUS floods a packet through the emulated LAN, another LEC will learn the destination’s location and respond to a subsequent LE_ARP.

Often, an LEC locally caches MAC address-to-ATM address mappings learned from listening to LE_ARP responses. The LEC first consults a local cache table for the destination MAC address, and it uses the cached mapping if a match is found instead of resolving the MAC address to the destination ATM address using the LE_ARP procedure described previously. The LEC ages cached entries over an adjustable interval, usually on the order of minutes, and removes the cached entry if no packets were sent to that MAC address. This aging procedure ensures that invalid mappings eventually leave the cache. The LEC clears data-direct VCCs if no activity occurs for the predetermined adjustable interval on the order of minutes. This time-out procedure ensures that ATM network connection resources are used efficiently.

LECs also utilize the BUS for broadcast and multicast packets. Because the BUS sends these packets to all LECs over the multicast forward VCC, the source LEC receives its own broadcast/multicast packet. Because some LAN protocols prohibit this, LANE specifies

that the source LEC prefix each packet with its LECID so that it can filter out packets received from itself via the BUS over the multicast forward VCC.

Even at this relatively high level of detail, the reader should rightfully conclude that LANE is a rather complex protocol. In fact, the LANE 2.0 specification contains over 120 pages of descriptions and drawings detailing the operations and protocol elements just summarized, and also covering some optional LANE capabilities not described. Comparing LANE with the use of multiprotocol encapsulation to connect bridges or routers as described earlier in this chapter, the level of complexity is even more striking. As discussed in the next section, a network-based service that attempts to interact with LAN bridging protocols introduces even more complexity.

LANE and Spanning Tree

The LANE protocol supports the IEEE spanning tree protocol described in Chapter 9. A number of complex situations result when external networks connect to emulated LANs via LAN switches and bridges that employ the spanning tree protocol. Furthermore, these external switches and bridges may be connected over shared-media LANs, creating the possibility of multiple paths between source and destination, and hence the possibility of fatal bridging loops.

LECs within LAN switches exchange spanning tree bridge packets over the BUS. If a bridging device detects a loop via the spanning tree protocol, then it disables one of the external ports involved in the loop. Because the spanning tree protocol employs bandwidth-weighted metrics, it first turns off lower-speed LAN ports prior to disabling any high-speed ATM ports.

Within a complex bridged network using the spanning tree protocol, the reachability of external MAC addresses through a particular LEC changes whenever network conditions change. This dynamic nature of the spanning tree protocol can interact unfavorably with the LANE protocol. For example, the ARP cache could map one or more external MAC addresses to an LEC's ATM address; which is no longer capable of reaching the MAC address as determined by the spanning tree protocol in response to a change in the legacy LAN topology.

The LANE protocol supports LE-Topology-Request messages to minimize the duration of these transient lapses in connectivity. Any LEC implementing the spanning tree protocol that detects a bridged topology change that triggers a BPDU configuration update message should also distribute an LE-Topology-Request via the LES. Upon receipt of the LE Topology Request message, all LECs must reduce the aging period on their cached ARP information. This action flushes out the cached information more rapidly and causes LECs to update mapping information through LE-ARPs. LECs do not disconnect existing data-direct connections; however, the updated cache information will cause inactivity on the data-direct VCC, causing it to eventually time out as well. Hence, the emulated LAN heals itself in conformance to the dynamically changed LAN within no more than a few minutes, as controlled by the aforementioned timers.

LANE Implementation Considerations

Recall from Chapter 9 the important characteristics of a LAN: high-speed, broadcast-capable, connectionless service, along with plug and play operation. A key challenge for LANE was resolving the fundamental difference between ATM's connection-oriented, point-to-point protocol and the inherently connectionless, shared-medium broadcast capability of LAN protocols, in particular Ethernet and Token Ring. Hence, a key function in LANE is the emulation of a broadcast medium. Note that the amount of broadcast traffic limits the overall capacity of an emulated LAN (ELAN) to that of the slowest interface. Generally, it is not a good idea to do a lot of LANE over the wide area, because of broadcast traffic. For example, the IP ARP procedure described in Chapter 8 generates a smaller amount of broadcast traffic than other network layer protocols do. For example, if five percent of the traffic is broadcast, than an OC3 worth of traffic offers almost 8 Mbps of traffic to a single 10 Mbps Ethernet station. For this reason, ELANs should have no more than a few thousand stations. In order to improve scalability, LANE 2.0 separates multicast traffic from the general broadcast path via a filtering protocol that determines which members of the emulated LAN receive particular multicast frames.

An LEC resides on every ATM-attached station in an emulated LAN. Although Figure 18-8 depicts all server functions separately, the LANE protocol does not specify the location of any of the server components; any device or devices with ATM connectivity suffice. For the purposes of reliability and performance, most vendors implement these server components on networking equipment, such as ATM switches or routers, rather than on a workstation or host.

Each LEC is part of an ATM end station, which represents a set of users identified by their MAC address(es). Communication among LECs and between LECs and the LES occurs over control and data ATM VCCs, which may be implemented as either SVCs, PVCs, or some combination thereof. There are no call setup and release procedures in a PVC-only LAN; instead, layer management sets up and clears connections. However, the large number of PVCs required makes all but the smallest emulated LAN networks using PVCs too complex to manage.

Because MAC addressing is flat (that is, no logical hierarchy exists), bridges must flood connectivity data throughout the emulated LAN. LANE makes the same trade-off that bridges do, achieving the ease of plug and play operation at the cost of decreased scalability. A further differentiating advantage of ATM LAN emulation over LAN switching is that it implements LANs at aggregate speeds on the order of gigabits per second. For example, a network of 16 workstations with 622 Mbps ATM NIC cards operates at an aggregate rate of 10 Gbps (full duplex). Note, however, that when a system is connected to low-speed legacy LANs, like 10 Mbps Ethernet, the slowest port on the LAN limits the total amount of broadcast traffic, and hence the maximum emulated LAN capacity. Also, emulated LANs make efficient use of WAN bandwidth, avoiding the problems in spanning tree bridging by using a point-to-point topology of ATM VCCs instead of the tree topology of a spanning tree.

On the other hand, a LAN switch learns about MAC addresses on adjacent LAN segments and doesn't propagate this information. Hence, a switch shields LAN segments

from each other by not propagating broadcast information. Several LANE functions specifically support the operation of LAN switching.

LAN designers must use routers to interconnect ELANs to achieve greater scalability. This works because LANE looks exactly like a LAN to a router. The administrator may also assign file servers, printer, Internet gateways, and other resources to the ELAN with the highest local traffic usage to minimize the amount of traffic flowing through the router.

The LANE UNI 2.0 protocol does not allow for the standard support of multiple LESs or BUSs within an ELAN. Hence these components represent both single points of failure and potential bottlenecks. The ATM Forum extended the LANE specification to address this issue by defining a LAN emulation NNI (LNNI), which operates between the server components within a single emulated LAN [AF LNNI 2.0].

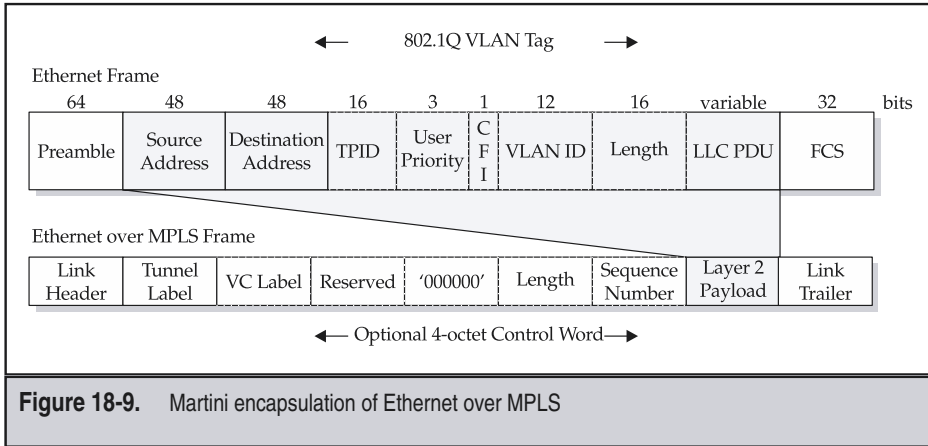
Multivendor interoperability of LANE NIC cards, bridges, and routers is good, and in this sense, the LANE specification was successful. However, the degree of complexity of configuration when compared with a native LAN card was an impediment to the adoption of ATM LANE in workstations. Furthermore, high-performance LAN NIC cards became so cheap when compared with ATM NIC cards that LANE did not achieve critical mass in the marketplace. LANE still has some use in interconnecting bridges and routers in enterprise networks, but faster, cheaper Ethernet solutions are more often used. As will be seen in the discussion regarding Ethernet over MPLS, there is a potential to make a solution overly complex if the lessons of LANE are not learned.

ETHERNET OVER MPLS

This section summarizes the emerging set of protocols and architectures envisioned to support Ethernet over MPLS. We begin with the easy part, namely encapsulation, before covering scenarios that are more complex, such as networks of devices and network participation in bridging protocols. The section concludes with a discussion regarding the possible use of MPLS to extend Ethernet across metropolitan and wide area networks. There are some important issues regarding efficiency, complexity, and scalability of such approaches, a topic further discussed in Chapter 29.

Martini Encapsulation of Ethernet over MPLS

Few things are simpler than the Martini encapsulation of Ethernet over MPLS from the IETF Pseudo Wire Emulation Edge to Edge (PWE3) working group, as illustrated in Figure 18-9 [Martini 01]. The transmitter places an entire valid Ethernet frame, without the preamble or FCS, in a single packet, along with an optional control word. The preamble and FCS are not needed, because the link layer header and trailer perform these functions. The tunnel label is set up using the procedures described in Chapter 14. The VC label and associated parameters are distributed using an extension to the Label Distribution Protocol (LDP) in a manner similar to that described for Frame Relay in Chapter 17. If the four-octet control word is used (e.g., for sequencing as described in Chapter 12),



then the flag bits must be set to zero and are ignored on receipt. The situation is similar for an Ethernet 802.1Q VLAN-tagged Ethernet frame (see Chapter 9), where the four-octet VLAN tag is also sent along with the rest of the standard Ethernet frame, sans the preamble and FCS fields in a single packet along with an optional control word. Note that the egress LSR may overwrite the four-octet VLAN tag, which allows Ethernet MPLS implementations to support a much larger number of VLANs than native Ethernet switches can.

Furthermore, the ingress LSR may consider the user priority field of the 802.1Q VLAN tag when determining what QoS treatment to apply to the packet, for example, the EXP field of the MPLS label stack as described in Chapter 20. Similarly, the egress LSR may consider the QoS of the encapsulating protocol when processing the packet prior to forwarding. These QoS capabilities in conjunction with MPLS traffic engineering described in Chapter 14 provides some important differentiators when compared with native Ethernet switching, as discussed later in this section.

This encapsulation is relatively efficient in that it drops 12 octets of overhead by not sending the preamble and FCS, which is offset by the 12 octets for the tunnel label and optional VC label and control word. The only additional overhead is then that required for the link layer header and trailer. However, when Ethernet frames are carried over an IPv4 tunnel, a 20-octet IPv4 header plus other information (e.g., L2TPv3) replaces the four-octet tunnel label, and therefore the encapsulation efficiency is further reduced.

Virtual Private LAN Service (VPLS)

At the time of writing, the IETF provider provisioned VPN (PPVPN) working group was working on the requirements [Augustyn 02], framework [Andersson 02], and protocol extensions necessary to support a Virtual Private LAN Service (VPLS). We summarize the current state of this work in the remainder of this section. The reader interested

in more up-to-date information should consult the PPVPN working group page at www.ietf.org. This effort was driven by several vendors as well as service providers, and therefore the summary of Ethernet support over MPLS and IP tunnels in the remainder of this chapter was based upon the draft standards work just cited, as well as white papers available on the Web sites of Cisco, Extreme Networks, and Riverstone Networks.

A VPLS supports the connection of multiple sites over a provider-managed IP or MPLS network where all sites appear to be on the same broadcast-capable Ethernet LAN segment. As such, a VPLS supports a set of devices at each of the sites identified by MAC address(es) and/or 802.1Q VLAN tag(s). As shown in Figure 18-10, a network of VPLS provider edge (PE) devices interconnected over MPLS and/or IP network(s) may support a VPLS for multiple customers by using an instance of a virtual forwarding and switching (VFS) function for each customer edge (CE) device at a VPLS site that is supported. Examples of a CE device are an Ethernet bridge, Ethernet switch, or a multi-protocol router attached via Ethernet to the PE. Other proposals describe use of an Ethernet pseudo-wire over MPLS as the access connection between the CE and the PE. A VFS instance can be thought of as a virtual LAN switch dedicated to a specific customer VPLS. For this reason, sometimes the term VFS instance is abbreviated as virtual switch instance (VSI). The access connection between a VPLS customer site and a PE may be a

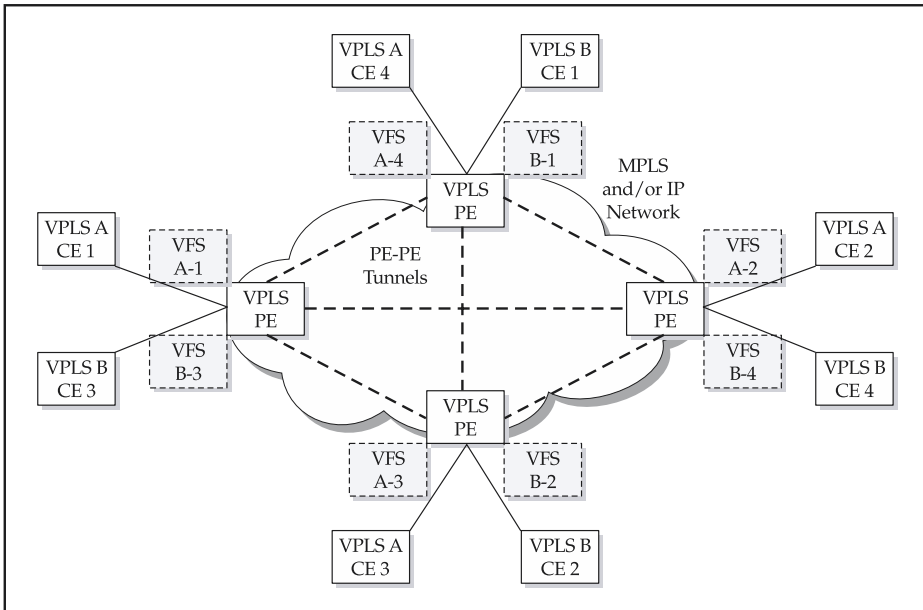


Figure 18-10. Virtual private LAN service (VPLS) networking terminology

physical or link layer logical circuit capable of transferring Ethernet frames. The VFS instances at each PE that are part of a particular VPLS are interconnected by a set of emulated pseudo-wire connections, for example, using the Martini encapsulation of Ethernet over MPLS described in the previous section. Of course, many pseudo-wires (e.g., as identified by the optional VC label) may use the same MPLS or IP tunnel established between a pair of PE devices. This aspect of the architecture helps VPLS scale better, because it limits the number of required tunnels. The VFS function performs MAC address learning and switching of Ethernet frames based upon learned MAC addresses and/or configured VLAN tag information among the CEs at the sites that are members of a VPLS or a VPLS VLAN. An important differentiator of the Ethernet over MPLS/IP tunnels versus native Ethernet switches is that different customers may use the same VLAN tags and even the same MAC addresses. Of course, an important requirement on the intercommunication between the VFS functions associated with a VPLS instance is that loop-free forwarding result. One implication of this requirement is that a VFS never sends frames back out on the access connection or pseudo-wire ports on which they are received.

As in ATM LANE, broadcast emulation is an important function in VPLS. The service requires supports of Ethernet broadcast within each customer instance of VPLS, and when VLAN tags are supported, broadcast must be limited to the set of sites that are part of that VLAN within a VPLS. Figure 18-11 illustrates an example of one approach under consideration that shows more detail of a feasible VPLS implementation and shows how it supports switching of Ethernet unicast and broadcast frames. This example is an exploded view of the network described previously for a particular VPLS. Three PE devices around the edge of an MPLS and/or IP network cloud are interconnected by a full mesh

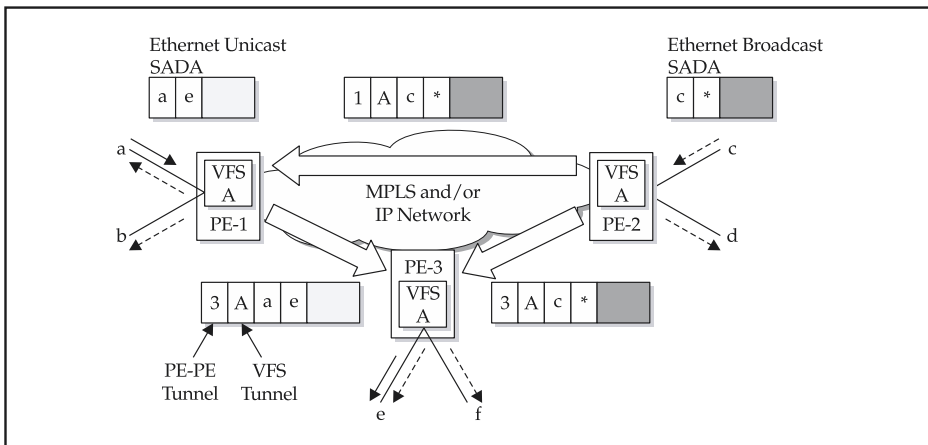


Figure 18-11. Ethernet unicast and broadcast over MPLS support in a VPLS

of tunnels. For MPLS, this would be a set of unidirectional tunnels, with a subset of these tunnels shown via the wide directed arrows through the cloud. The outermost (top) label is labeled with the number of the destination PE in the example, with the innermost (bottom) label indicating the target VFS. If the tunneling protocol is MPLS, then observe that this packet has the same generic structure as that shown for the Martini Ethernet over MPLS pseudo-wire encapsulation described earlier, and therefore the inter-VFS tunnels can be viewed as a pseudo-wires. As described earlier, a PE contains a VFS for every VPLS to which there is an access connection to a CE. The figure shows the source and destination address (SA and DA) Ethernet MAC addresses as lowercase letters attached to the VFS corresponding to VPLS A, denoted as VFS A at each of the PEs. We now give an example of the forwarding of the Ethernet unicast and broadcast frames.

The CE with MAC address a in the upper left-hand corner of Figure 18-11 generates an Ethernet unicast frame destined for MAC address e. The VFS A in PE 1 has determined through communication with the VFS A in the other PEs that this MAC address is (currently) connected to PE 3. Therefore, VFS A in PE 1 pushes on the VFS tunnel header A identifying that the packet should be processed by VFS A at the tunnel endpoint and then pushes on the PE-PE tunnel header 3 indicating that the destination is PE 3. Once PE 3 receives this packet, it pops the two tunnel headers and delivers the Ethernet frame over the access connection labeled e, as shown by the solid arrow in the figure. Other instances of the VFS function in the PEs operate completely independently of VFS A, but can use the shared PE-PE tunnels, with Ethernet frames multiplexed using the VFS tunnel header such that LAN traffic is kept completely separate. Furthermore, because the MAC addresses and VLAN tags are processed only by the VFS instances at the PE nodes supporting a particular VPLS, the MAC address and VLAN tags need only be unique within a particular VPLS. This mechanism allows a network of VPLS-capable PE nodes to implement many more than 4096 VLAN tags across such an Ethernet over MPLS and/or IP tunnel network.

Our example continues with the forwarding of an Ethernet broadcast frame from interface c in the upper right-hand corner of Figure 18-11, where the asterisk in the Ethernet DA field is used to indicate broadcast. The VFS A in PE 2 identifies the frame as broadcast from the Ethernet header as described in Chapter 9. It has determined via a MAC learning protocol that there are other CEs connected to VPLS A supported by a unique instance of the VFS in PE 1 and PE 3. VFS A in PE 2 then generates a copy of the Ethernet broadcast frame prepended by the VFS tunnel header and then, for each destination PE, pushes the PE-PE tunnel header onto the packet. The MPLS and/or IP network delivers the packet to each of the PEs, which then broadcast the packet on each of their interfaces that are part of VPLS A. That is, as shown by dashed arrows in the figure, PE 1 sends the broadcast packet on interfaces a and d, PE 2 sends the broadcast packet on interfaces d and e, while PE 3 also sends the broadcast packet out on interface f, but not interface e. Although this method for supporting broadcast is less efficient than a network-based weighted spanning tree, it can be effective for a relatively small number of PEs over a limited geographic area for Ethernet traffic that does not have a large proportion of broadcast traffic.

Comparing the VPLS service to that of multiprotocol encapsulation over ATM of Figure 18-1, note that a similar full mesh of tunnels or connections connects routing or

bridging devices. Also note that the emulation of Ethernet broadcast via a full mesh of virtual tunnels is similar to a particular implementation option of LANE. However, the use of stacked labels in VPLS significantly reduces the number of tunnels necessary if there are many VFS instances in each PE. This occurs because VPLS is a network-based virtual service, while ATM LANE is inherently a dedicated private network solution.

In a manner similar to LANE, manual configuration of very small networks like that of the preceding example may be possible; however, in order to implement a scalable solution automatic protocols are necessary. At the time of writing, specific protocols to support VPLS were under development within the IETF PPVPN working group. These protocols include a means to associate membership of CE devices to a particular VPLS and to discover new CE members and authenticate them, as well the details of signaling necessary for the instantiation of labels and other parameters for the multiplexing of communication between VFS instances over the shared inter-PE tunnels. Different architectural approaches are possible for some of these protocols. For example, the discovery and authentication functions could be performed by a protocol (e.g., DNS or LDAP) using some type of centralized server similar to LANE, or through use of extensions to a distributed IP routing protocol (e.g., BGP), and/or via exchange of tokens via the VFS tunnel establishment signaling protocol.

VPLS and Access to the Internet

The vision for use of VPLS-type services can include a mix of point-to-point and broadcast (sometimes also called point-to-multipoint by some vendors and Internet drafts) services. Furthermore, the use of the VPLS technology can also provide access to the Internet over the same Ethernet interface to a CE, as illustrated by the following example. Figure 18-12 illustrates use of VPLS point-to-point and broadcast services providing virtual Ethernet segments between CE sites along with access to an Internet service provider (ISP) [Extreme 01]. CE devices are for a particular customer, who has three instances of VPLS, denoted as A, B, and C. The first instance is VPLS A, which provides access from CE 1 to an Ethernet port on an IP router in an Internet service provider (ISP) network, which provides access to the Internet and/or network-based IP VPN services, as described in Chapter 19. The second instance is VPLS B, which provides a broadcast-capable LAN segment between CEs 1 through 4, as shown by the solid lines converging on a dot in the center of the Ethernet over MPLS network cloud in the figure. The third instance is VPLS C, which is a point-to-point Ethernet segment between the CEs at sites 2 and 3, as shown at the bottom of the figure.

When a CE or an IP router uses the same physical Ethernet interface for more than one VPLS, either VLAN tags or multiple MAC addresses are necessary to distinguish the VPLS service for which Ethernet frames are destined. The use of different VPLS instances in the preceding example could be motivated by the following factors. VPLS A provides access to the Internet via CE 1, because this site has firewall and network address translation devices and is the security enforcement point for access to the Internet for this set of customer sites. VPLS C provides a guaranteed capacity point-to-point Ethernet segment with a guaranteed level of QoS between CE 2 and CE 3, which have mission-critical servers

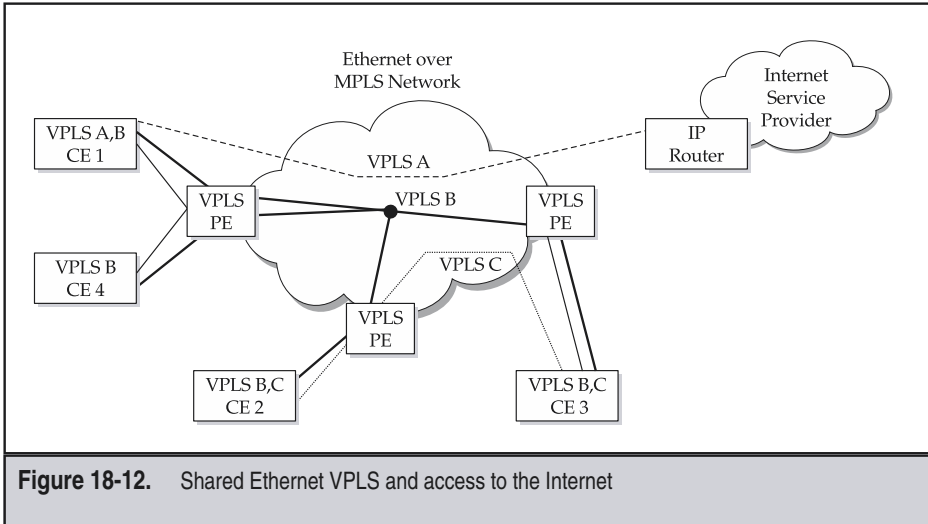


Figure 18-12. Shared Ethernet VPLS and access to the Internet

that require such a capability. Finally, VPLS B provides a broadcast-capable virtual LAN segment that provides for communication between all CE devices for this customer. CEs 2, 3, and 4 could use VPLS B to communicate with the security devices at the CE 1 site for access to the Internet. Of course, many other combinations are also possible, with the preceding example provided as illustration of one possible scenario.

Interworking Network Layer Protocols over MPLS

The solution summarized in the previous section would be great, if there were a ubiquitous Ethernet over MPLS network available everywhere. However, it is unlikely that any layer 2 networking technology will be available everywhere at the range of speeds and price points that make economic sense at all customer sites. In response to this need, some networking engineers are exploring the possibility of whether a protocol solution is feasible to allow sites running an arbitrary network protocol over Ethernet, ATM, FR, and other link layer technologies [Andersson 02]. The motivation here is similar to that of multiprotocol encapsulation of any network layer protocol over ATM and FR discussed earlier in this chapter. As described earlier in this chapter, supporting an arbitrary network layer protocol over a hybrid FR and ATM network involves a straightforward mapping of the multiprotocol encapsulation header and the format of ARP and inverse ARP messages. This mapping is straightforward because FR and ATM have similar semantics and are both nonbroadcast multiple access networks (NBMA) in the networking taxonomy described in Chapter 9, and because multiprotocol over FR and ATM implementations commonly use only the inverse ARP protocol described in Chapter 17. However, when we add Ethernet to the mix, the situation becomes more complex because Ethernet is

inherently a broadcast network and implementations use the ARP mechanism described in Chapter 9.

Two possible types of solutions have been proposed to address the potential need to interwork network layer protocols over a network with a mix of NBMA and broadcast access connections. The first approach also involves some form of function in an L2 VPN network, which maps between the inverse ARP paradigm used by FR and ATM and the ARP paradigm used in Ethernet. A draft specification of this type of function has been specified [Shah 02], which we summarize here. Figure 18-13 illustrates the problem and the form of the proposed solution in the context of a pseudo-wire connecting the VFS corresponding to the two customer sites, CE 1 connected via Ethernet and CE 2 connected via a FR or ATM VC. As shown in the upper left-hand corner of the figure, the Ethernet CE 1 router uses an IP ARP that contains the source IP address (IP-1) and source MAC (MAC-1) addresses, which PE 1 learns, and that also contains the query for the MAC address corresponding to a destination address IP-2. The incompatibility arises because the CE 2 device with IP address IP-2 is attached via an NBMA FR or ATM VC-2, as shown on the right-hand side of the figure. In the upper right-hand corner of the figure, CE 2 uses an inverse ARP (inARP) message to communicate its IP address (IP-2) and implicitly also communicates the VC identifier by virtue of transmitting the inARP message, which also contains the query as to what IP address is at the other end of the VC. When an Ethernet- or FR/ATM-attached IP device is attached to another device, the response is straightforward. In the mixed link layer scenario here, the PEs must exchange some additional information and perform additional functions to resolve the incompatibility as shown in the middle of the figure.

Observe that PE 1 learns from the ARP message that IP-1 is associated with CE 1 and the pseudo-wire (PW) for PE 2, and that PE 2 learns from the inARP message that IP-2 is associated with the other end of the PW. PE 1 and PE 2 can then exchange this information over the L2 VPN network, as shown in the center of Figure 18-15. Now, PE 2 has the information necessary to generate the response to the inverse ARP, and PE 1 assigns MAC address MAC-2 to act as the proxy ARP response for the FR/ATM end of the virtual LAN segment. Note that this approach introduces some complexity into the network and

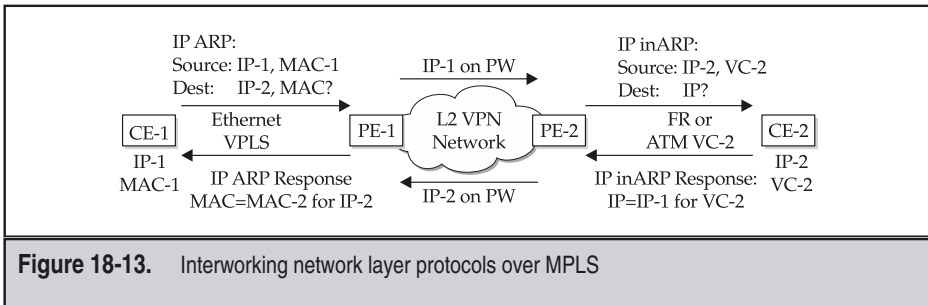


Figure 18-13. Interworking network layer protocols over MPLS

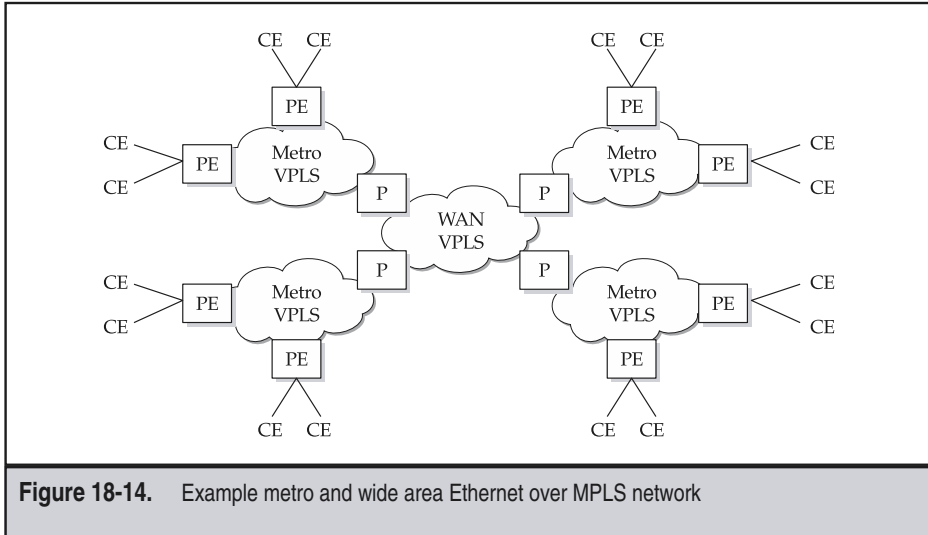


Figure 18-14. Example metro and wide area Ethernet over MPLS network

works only for IP. Unfortunately, there are other functions in other network protocols, such as the designated router, that operate differently on broadcast and NBMA networks, adding further complexity. Other issues include the incompatibility of the media type in IP routing protocols (e.g., CE 1 advertises a link type of broadcast for Ethernet, while CE 2 advertises a link type of NBMA), as well as the mapping of traffic management parameters and status signaling.

The second approach would be for the CE to further encapsulate each network layer packet over a common (or at least, mappable) multiprotocol encapsulation and link layer protocol and configure every such link as a point-to-point, NBMA access connection [Andersson 02]. For example, L2TP or PPP could be used for this purpose. The address learning protocol would then be common across all such access connections, avoiding the incompatibility of inverse ARP used in FR and ATM and that of broadcast ARP used in Ethernet. These point-to-point connections would then need to terminate on some network-based function, which could be either a single device or a function logically distributed across the network, that would perform the link layer encapsulation between FR, ATM, and Ethernet, along with any mapping of multiprotocol encapsulation. Although this alternative is conceptually simpler, it could require a change in configuration to a number of CE devices.

Metropolitan and Wide Area Ethernet over MPLS Networking

Although Ethernet has its origins in the local area network, expansion into a metropolitan area is relatively straightforward and has been announced by a number of vendors and

service providers [Extreme 02]. When combined with MPLS or core IP tunneling technology, Ethernet-style services could also be extended to the wide area, as illustrated in Figure 18-14. Around the periphery of the figure, four metro area VPLS networks, for example, implemented using the approach described earlier (see Figure 18-10) are connected by one wide area network (WAN) VPLS network. The functions involved in interconnecting the metro VPLS networks across the WAN may be different from those at the PE edge, which we show as a provider (P) device function without any customer CE facing interfaces interposed between the metro and WAN VPLS networks. In theory, essentially the same services of point-to-point LAN segments, broadcast LAN segments, access to the Internet, and access to other services could be supported by this architecture.

Often-cited MPLS enablers for these Ethernet metro and wide area services are traffic engineering and Quality of Service. Using MPLS, the tunnels between PE devices can be traffic-engineered to provide a particular level of capacity for the set of customers using VPLS, as well as provide some infrastructure for support of QOS and SLAs. A conservative design would be to establish a fully meshed set of point-to-point tunnels, or “pipes,” between each of the PEs to handle the worst-case traffic originated by every CE, because the traffic from any CE site may be destined for any other CE site on a particular VPLS. However, this would be a rather expensive and inefficient solution. An alternative service model that is being considered is one where the amount of capacity that a particular CE can send is limited, which effectively defines the size of the “hose” over which a CE site can spray traffic to any of the other sites. Service providers could then determine PE-PE tunnel capacity sizing that is much less than the worst-case “pipe” model through use of traffic statistics that will likely be predictable when relatively large numbers of customers use a shared VPLS infrastructure.

Although this is an attractive high-level concept, MPLS does not fundamentally change a number of the issues arising from implementing a broadcast-capable protocol like Ethernet across a large geographic area or a large number of devices. If the CE devices are Ethernet bridges, then the same issues described in Chapter 9 relating to the inefficient use of WAN capacity by the spanning tree protocol remain. In summary, these are use of only a subset of point-to-point links and the relatively inefficient routing of Ethernet frames by the spanning tree protocol up toward the root of the tree and then back down the tree toward the destination, is a behavior that that can be quite unexpected in a WAN. As discussed in Chapter 9, use of routers as CE devices alleviates many of these issues, at the expense of increased complexity. On the other hand, on broadcast networks routed protocols and applications generate some broadcast traffic, which limits the size of the broadcast domain in a manner similar to that encountered in ATM LANE, as described earlier. Possibly an effective combination of services could be broadcast domains in metro VPLS areas interconnecting bridges and/or routers, but use of only point-to-point Ethernet LAN segments for connections between routers in different metro areas. However, this type of architecture has competition when the native service required by a customer is IP. The next chapter describes a network-based L3 VPN as a potential alternative and then compares and contrasts it with this Ethernet-based approach.

REVIEW

This chapter covered the important topic of ATM and MPLS support of local area network (LAN) protocols, with a focus on Ethernet. We first covered the Multiprotocol over ATM encapsulation standard defined in IETF RFC 2684. The text described the protocol encapsulation and VC multiplexing methods, providing guidelines for selection of the most economical and efficient method for a particular application. The text then summarized how the ATM Forum's LAN Emulation (LANE) protocol supports seamless interconnection of ATM hosts and servers with their legacy LAN counterparts. The treatment covered hardware and software aspects, defined the components and connections involved in the LANE protocol, and summarized the steps involved with initialization, registration, address resolution, and data transfer. During this exposition, we highlighted reasons why LANE was attractive in the late 1990s, but how added functions to Ethernet as well as practical and economic considerations resulted in adoption of other approaches. Our discussion began these LAN over ATM approaches, because in some aspects, the emerging Ethernet over MPLS approaches build upon the lessons learned from this experience. The last part of the chapter summarized the emerging Ethernet over MPLS efforts in the industry and related standards efforts. We reviewed the Martini encapsulation of Ethernet frames over MPLS and described how this is used between provider edge devices that are shared among many customers, providing to each a virtual private LAN service (VPLS). The text described the functional components of a VPLS and summarized the state of standards work needed to result in interoperable implementations. The coverage then moved on to the support for networking protocols over Ethernet and other link layer networks. Finally, the chapter concluded with some design considerations involved in VPLS, focusing on aspects that are important in metropolitan and wide area networks.

CHAPTER 19



ATM and MPLS Support of Enterprise-Level IP Networks

In the late 1990s, the Internet Protocol (IP) emerged as the de facto standard for internetworking not only within large enterprises but on the desktops of most information workers as well as home office and residential users. We covered the general support for IP over ATM in ISP networks and the motivation for migration to IP over MPLS in Chapter 10, with specific details in support of IP over MPLS covered in Chapter 14. This chapter details the ATM and MPLS protocol support for IP in enterprise-level virtual private networks (VPNs), in roughly historical order. Figure 19-1 summarizes the protocols covered in this chapter. The ATM-based designs overcome the fundamental difference between ATM's nonbroadcast, connection-oriented protocol and the Internet's broadcast, connectionless orientation via a combination of servers, address resolution, and emulation of broadcast capabilities. We focus primarily on describing support for classical IP subnetworks overlaid over ATM, since this was the most widely adopted ATM-based approach. We also summarize the standards developed for the Multiprotocol over ATM (MPOA) approach for extending IP address resolution beyond IP subnet boundaries and emulation of the IP multicast group service, focusing on why these approaches were not successful and what important lessons were learned. We continue with an overview of an emerging use of MPLS and IP tunnels in support of network-based virtual private networks (VPNs). We summarize the basic terminology and taxonomy of architectural approaches being considered in the IETF and being deployed by service providers. The text describes as well as compares and contrasts the two principal architectural approaches of aggregated routing across multiple VPNs and a separate virtual router (VR) instance per VPN. The chapter concludes with a discussion of path maximum transfer unit (MTU) discovery in ATM and MPLS networks.

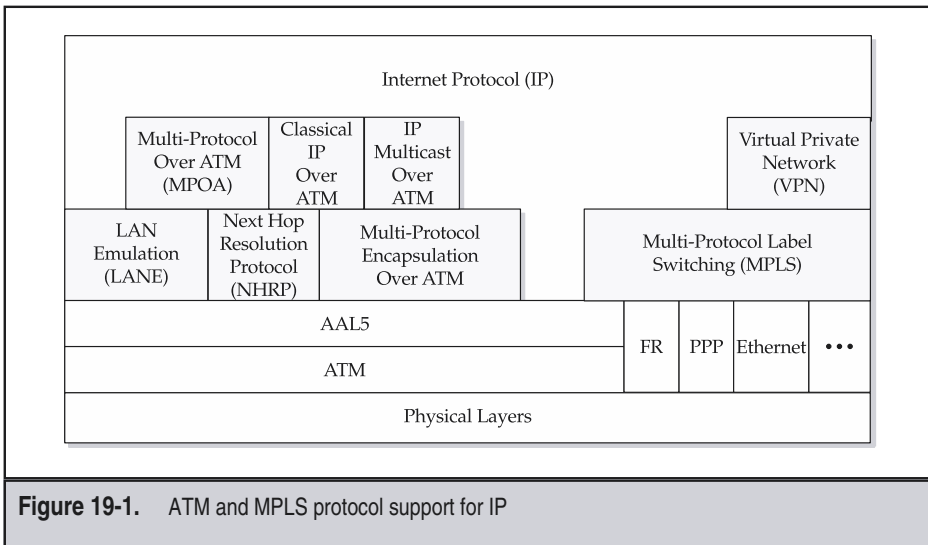


Figure 19-1. ATM and MPLS protocol support for IP

IP OVER ATM VIRTUAL PRIVATE NETWORKS

In support of ATM-based VPNs, this section provides a description of classical IP over ATM subnetworks, along with a summary of MPOA and IP multicast over ATM. As we will see, a basic concept involved in all methods supporting IP over ATM is resolution of an IP address to a corresponding ATM address, and then using that address to identify an already established VCC or dynamically setting up an SVC to that ATM address.

Classical IP over ATM

IETF RFC 2225 specifies Classical IP over ATM for the use of ATM as a direct replacement for the “wires” interconnecting IP hosts, LAN segments, and routers in a logical IP subnetwork (LIS). RFC 2225 combined RFC 1577 and RFC 1626 into one document and clarified several points gained from operational experience in operating real-world Classical IP over ATM networks. RFC 2225 specifies that implementations must support IEEE 802.2 Logical Link Control/Subnetwork Attachment Point (LLC/SNAP) encapsulation as described in RFC 2684, covered in Chapter 18. LLC/SNAP encapsulation is the default packet format for IP datagrams.

An LIS consists of a group of hosts or routers connected to an ATM network belonging to the same IP subnet; that is, they all have the same IP subnet number and mask as described in Chapter 9. Typically, this would be an enterprise network, such as for a corporation or government agency. These hosts and routers (generically called stations) must have both IP and ATM addresses in Classical IP over ATM subnetworks. The Classical IP over ATM procedures apply to both Permanent and Switched Virtual Connections (PVCs and SVCs). We cover the PVC case first.

Inverse Address Resolution Protocol (InARP)

RFC 2390 defines an *Inverse Address Resolution Protocol (InARP)* as a means for routers to automatically learn the IP address of the router on the far end of an ATM VCC PVC. Basically, it involves a station sending an InARP message containing the sender’s IP address over the ATM VCC PVC. This situation occurs in PVC networks upon initialization, or when a router reloads its software because the VCC is known, but the IP address reachable via the VCC is unknown. The router on the other end of the ATM VCC PVC then responds with its IP address, establishing an association between the IP addresses of the pair and the ATM VCC’s VPI/VCI on each ATM interface.

Figure 19-2 illustrates the principle of Inverse ARP (InARP) over an ATM network with three VCC PVCs. The router port on the left-hand side with IP address *A* sends an inverse ARP over VCC 52 [InARP(*A*)] in the first step. This appears on VCC 51 to the router port with IP address *C* on the right hand side of the figure. The router port with IP address *C* responds with its identity [InARP Response(*C*)] on VCC 51 in the second step. Now the router ports with IP addresses *A* and *C* know that they can reach other by transmitting on ATM VCCs 52 and 51, respectively.

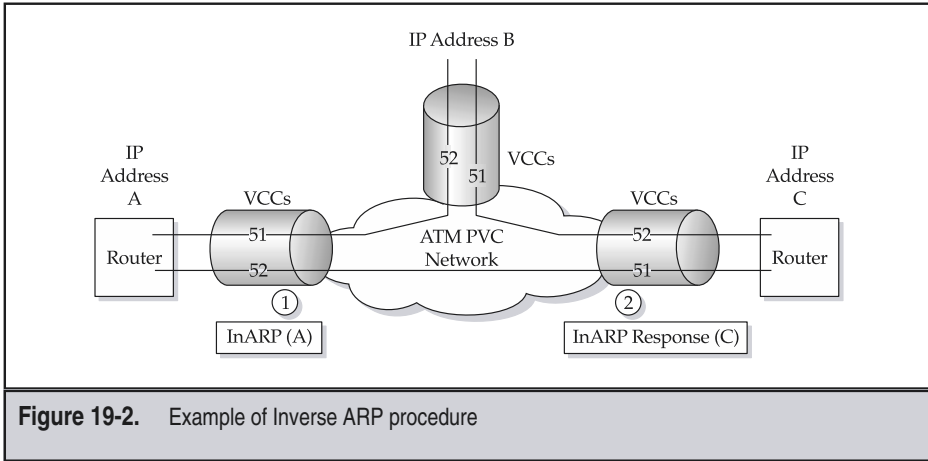


Figure 19-2. Example of Inverse ARP procedure

ATM Address Resolution Protocol (ATMARP)

Having to manually configure large numbers of PVCs does not scale in larger networks. Therefore, RFC 2225 also specifies an automatic configuration method using SVCs along with one or more ATM Address Resolution Protocol (ATMARP) servers, which allow IP/ATM stations to resolve the association of an IP address with an ATM address. Armed with the destination's ATM address, the originating station dynamically sets up an ATM SVC to the destination station. After a period of inactivity, the stations take down the SVC to efficiently utilize bandwidth. The ATM network may also release the SVC in response to failure or overload conditions.

IP/ATM stations register with the ATMARP server by establishing an SVC to the ATMARP server. The ATMARP server may either transmit an Inverse ARP request to the newly attached client to determine the station's IP and ATM addresses, or determine this information from ARP requests. In either case, the ATM ARP server stores the association of a station's IP and ATM addresses in its ATMARP table. The ATMARP server may periodically confirm presence of an IP/ATM host using an InARP message. The server also ages old entries and eventually removes unresponsive IP/ATM stations. Hosts must also age their ARP table entries to remove old data.

Figure 19-3 illustrates a simple example of the IETF's Classical IP over ATM concept. Two interfaces are shown: one with IP address A and ATM address X and the other with IP address B and ATM address Y. The devices with IP addresses A and B have already established an SVC to the ATMARP server and registered so that the ATMARP server knows the association of their IP and ATM addresses. IP/ATM interfaces have a VCC over which ATM UNI signaling messages are sent, as illustrated by the dashed line. When the station with IP address A wishes to send data to the station with IP address B, the first step is to send an Address Resolution Protocol (ARP) message to the ATMARP

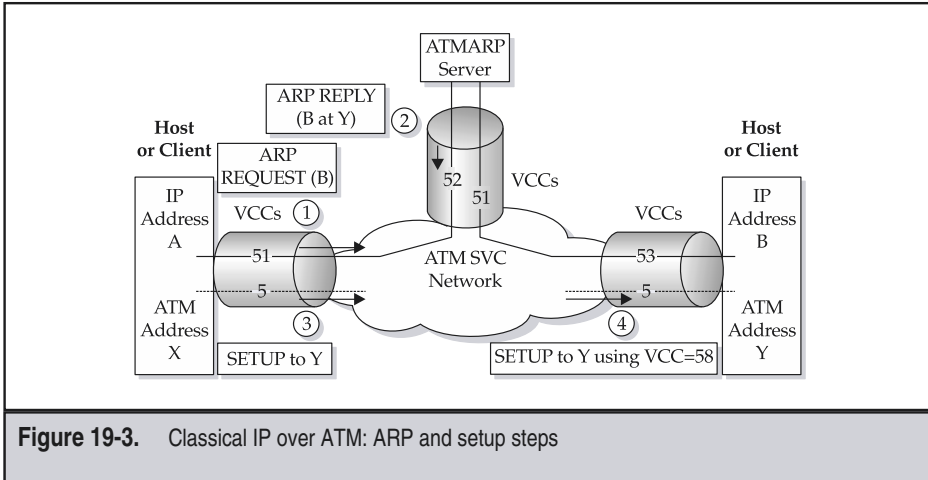


Figure 19-3. Classical IP over ATM: ARP and setup steps

server. In the second step, the server returns the ATM address *Y* of the device with IP address *B*. In the third step, the device with ATM address *X* originates a SETUP message to ATM address *Y*. The ATM network switches the SETUP message through to the destination ATM address *Y* on VCC 5. The switched ATM network makes a connection and requests a SETUP to ATM address *Y* using the signaling VCC 5. The SETUP message specifies that data traffic should be sent on VCC 58 as indicated in the figure.

Figure 19-4 illustrates the final steps in the Classical IP over ATM scenario. In the fifth step, the device with ATM address *Y* responds with a CONNECT message, which the switched ATM network uses to establish the VCC back to the originator, ATM address *X*. In the sixth step, the ATM network sends a CONNECT message to the device with ATM address *X* indicating that VCC 54 makes the connection with the device with ATM address *Y* that is using VCC 58. In the seventh and final step, communication between IP addresses *A* and *B* occurs over VCC 54 for IP address *A* and VCC 58 for IP address *B* as indicated in the figure. Either ATM address *X* or *Y* could release the ATM SVC by issuing a RELEASE message. Classical IP over ATM emulates the connectionless paradigm of datagram forwarding by automatically releasing the SVC call if no packets flow over the VCC for a preconfigured time interval. Typically, these times are on the order of minutes.

Classical IP over ATM Signaling Considerations

RFC 1755 specifies the details that hosts and routers require to achieve interoperability when using the SVC capabilities referred to in RFC 2225. In particular, it specifies the precise utilization of UNI 3.0/3.1 information elements in the SVC implementation of RFC 2225's Classical IP over ATM [McDysan 98]. These include the AAL Parameters, Broadband Low Layer Information (B-LLI), Logical Link Control (LLC), and Called/Calling Party Address. The address and subaddress fields may be either of NSAP format or E.164

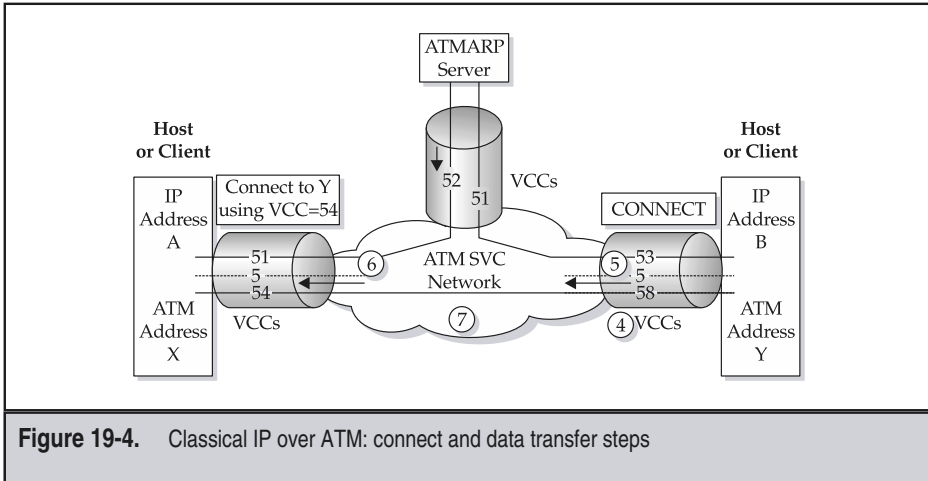


Figure 19-4. Classical IP over ATM: connect and data transfer steps

as described in Chapter 13. It also describes how the QoS Parameter, ATM Traffic Descriptor, and Broadband Bearer Capability information elements are used to request a specific ATM service category, as detailed in Chapter 20. The standard defines support for RSVP-type systems employing a token-bucket style characterization of the source, such as that defined in the RSVP standards (see RFC 1363 and RFC 1633). The standard also supports virtual ATM “pipes” between two routers, as well as a best-effort service targeted for end system usage. RFC 2331 updates these guidelines for the UNI 4.0 signaling specification. This specification defines support for Available Bit Rate (ABR) signaling for point-to-point calls, traffic parameter negotiation, and frame discard. The document also defines a procedure for holding up the ATM SVC as long as the receiver periodically refreshes its reservation using the RSVP protocol. See RFCs 1755 and 2331 for detailed coding of the UNI signaling messages.

Classical IP over ATM relies heavily on ATM SVCs. Acceptable call setup times and the support for reasonable call attempt rates are critical factors in the success of the approach of supporting connectionless services by dynamic connection switching. In a typical Classical IP over ATM subnetwork environment, the number of ATM SVC calls per end station is relatively small. Usually, a single user never has more than a connection open to a few file servers at most, an Internet firewall router, and a print server simultaneously. Since the duration of these connections are all on the order of minutes to hours, many users will have fewer connections. Hence, the largest call setup rates occur when a server or gateway fails and many stations attempt to reconnect.

Interconnecting Logical IP Subnetworks

Figure 19-5 illustrates the operation of multiple interconnected LISs. Each of the three LISs operates and communicates independently of all other LISs, even if they are all on

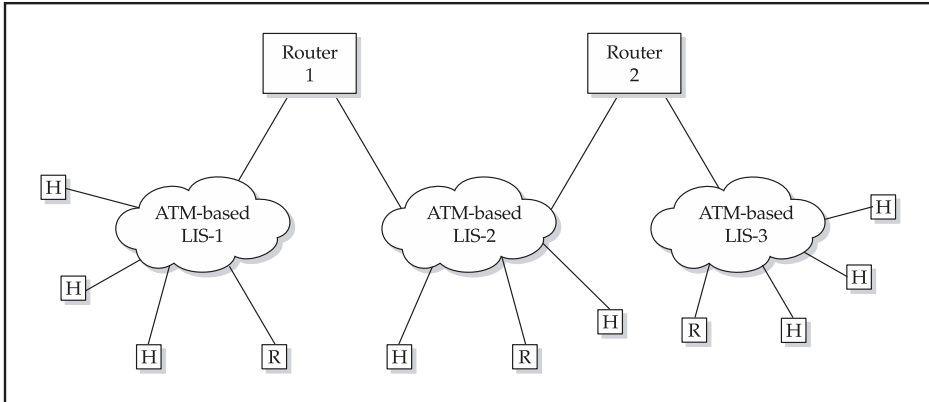


Figure 19-5. Router interconnection of logically independent subnets (LIS)

the same ATM network. The classical model of RFC 2225 requires that any packet destined outside the source host's or router's logical IP subnet (LIS) must be sent to an IP router. For example, a host on LIS-1 must send packets to the router 1 (i.e., the default router for packets not on its LIS) for delivery to hosts or routers on LIS-2 or LIS-3. Routers 1 and 2 are configured as endpoints of the LISs to which they are connected. RFC 2225 notes that this configuration may result in a number of disjoint LISs operating over the same ATM network, but the standard states that hosts on different LISs must communicate via an intermediate IP router, even though said hosts could open a direct VC between themselves. If the stations did not follow this rule, then routing loops could result. Making use of direct ATM connectivity between IP/ATM stations was a significant driver in the development of MPOA, as covered in the next section.

Multiprotocol over ATM (MPOA)

The ATM Forum initiated the Multiprotocol over ATM (MPOA) work in response to a perceived demand to extend ATM to protocols other than IP (e.g., IPX, AppleTalk, and DECNET). This section presents a brief history regarding the origins of MPOA, an overview of important aspects of the protocol, and an analysis of the lessons learned from this effort, along with a discussion on the reasons it was not commercially successful. For more details on MPOA, see the cited references or [McDysan 98].

MPOA History and the Roads Not Traveled

The early stages of the MPOA group considered three markedly different models for multiprotocol operation over ATM [Alles 95]. Several companies proposed a peer model that employed an algorithmic mapping of all network layer addresses into NSAP-based

addresses. This approach had the advantage of allowing PNNI to directly route signaling requests and precluded the need for a separate address resolution protocol, like that defined for Classical IP over ATM in the previous section. On the other hand, the peer model required different routing protocols in mixed ATM and router networks, resulting in suboptimal end-to-end routing and concerns about multivendor interoperability. Furthermore, it required that every ATM switch have address tables large enough for the ATM NSAP formatted addresses and the addresses resulting from the algorithmic mapping of the other address spaces.

The Integrated PNNI model (I-PNNI) proposed that ATM switches and routers universally employ the PNNI protocol. A motivation for this approach is the fact that PNNI is the most powerful and scalable routing protocol defined to date. I-PNNI could support the separate ATM address space, also called an overlay model, or the peer model with mapped addresses. I-PNNI's principal disadvantage was that all routers would need to implement it, and hence, its introduction would take years and the migration process would likely be extremely complex.

In the end, instead of developing an entirely new protocol, the ATM Forum decided to leverage two protocols developed earlier. The first was use of virtual LANs as the basis for MPOA utilizing extensions to the LANE protocol described in Chapter 18 applied to layer 3 switching with distributed routing. In this context, layer 3 switching refers to devices that make forwarding decisions based upon the network layer in the packet header. Advantages of this approach were elimination of multiple hops through software-based IP routers, reduction of the impact of broadcast storms seen in bridged networks, and provision of a means to map ATM QoS benefits to other protocols. MPOA standardized centralized route processing, which disseminated routing information to simpler packet forwarding engines, called edge devices. The ATM Forum also chose to reuse the layer 3 next hop and address resolution components of the IETF's Next Hop Resolution Protocol (NHRP). The design specified a query/response protocol that an MPOA client could use to request the next hop and address resolution information corresponding to a layer 3 address from an MPOA route server.

Overview of MPOA

Conceptually, MPOA distributed the principal functions of a router—data forwarding, switching and routing control, and management—into three networking layers [Riley 97], as illustrated in Figure 19-6. The traditional router implements all of these functions on a single machine, as shown in Figure 19-6a. Often, the central router processor card is the bottleneck in such designs. MPOA defines the concept of a *virtual router*, which distributes the traditional router functions to different network elements. Figure 19-6b illustrates how MPOA maps the router I/O card forwarding function to MPOA edge devices, how it distributes the router backplane to a network of ATM switches, and how it consolidates the routing computation at every network node to a smaller number of MPOA route servers.

The ATM Forum worked closely with the IETF in developing the MPOA version 1.0 specification. For example, the ATM Forum published drafts of the MPOA specification

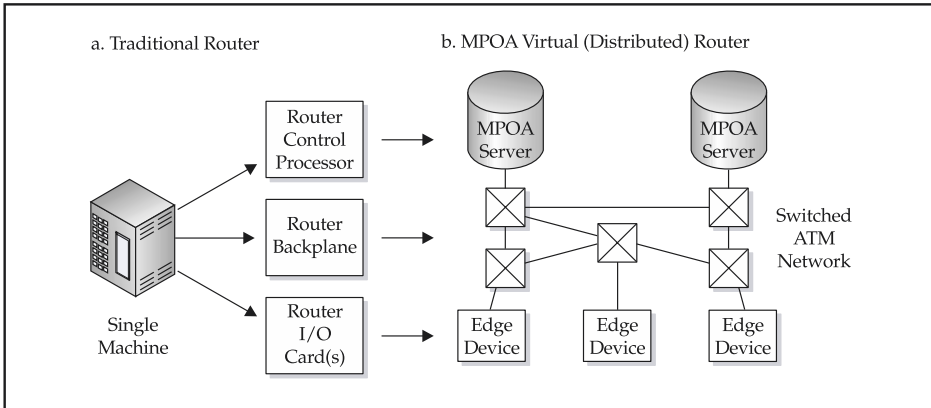


Figure 19-6. MPOA architectural separation of functions: the virtual router

and contributions, normally accessible to members only, to the public in support of this effort. This summary is based upon the final version of the MPOA specification [AF MPOA 1.0] and other descriptions [Swallow 96].

Figure 19-7 illustrates the two components in an MPOA network: edge devices, and MPOA-capable routers. An emulated LAN (ELAN) connects a network of edge devices and MPOA routers. LAN Emulation Clients (LECs) interconnect MPOA edge devices (also called hosts) and MPOA-capable routers. An MPOA edge device also contains an

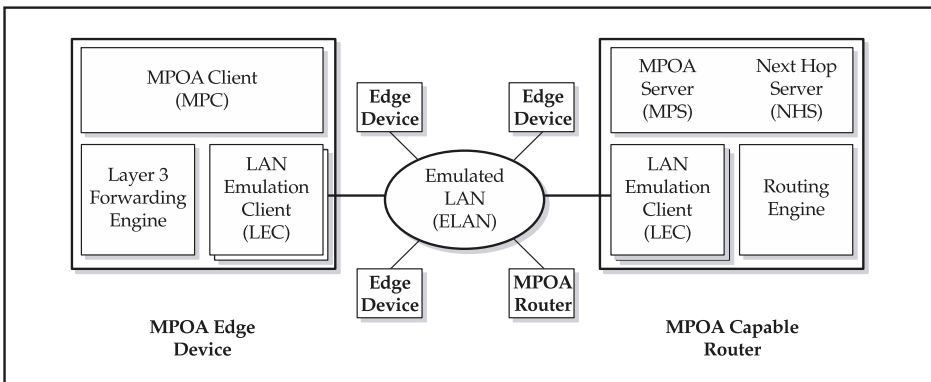


Figure 19-7. MPOA network components

MPOA Client (MPC) and a layer 3 forwarding function. Edge devices reside on the periphery of ATM networks, usually supporting traditional LAN interfaces, such as Ethernet and Token Ring. Edge devices are capable of bridging with other edge devices and hence can be part of virtual LANs. They also have a limited amount of layer 3 processing capability based upon information fed from MPOA route servers. MPOA edge devices do not participate in routing protocols.

On the right-hand side of Figure 19-7, an MPOA-capable router contains an MPOA Server (MPS), which includes the NHRP Next Hop Server (NHS) function and a routing function. MPOA synthesizes bridging and routing with ATM. LAN Emulation (LANE) performs the bridging function. MPOA separates switching (or forwarding) from routing in a technique called *virtual routing*, in which the MPOA routers compute the values for the next hop forwarding tables and download them to the edge devices. MPOA servers participate in standard routing protocols, like OSPF, IS-IS, and RIP with each other as well as with traditional routers on the periphery of the MPOA network. The routing function in MPOA servers exchange topology information and calculate routes, while the edge devices perform the majority of the layer 3 forwarding. Since the edge devices perform layer 3 forwarding, latency is reduced and higher throughput is achieved more cost effectively than with traditional routers using processor-based forwarding capabilities. The NHRP Next Hop Server (NHS) distributes layer 3 forwarding information to MPOA clients via NHRP with the addition of a cache management protocol. The NHRP function supports the potential to resolve the ATM address associated with a destination station on a different subnet, overcoming the limitation of Classical IP over ATM mentioned earlier. Unfortunately, some additional configuration complexity was necessary to minimize the possibility of routing loops [McDysan 98].

MPOA specifies that a single route server may serve multiple edge devices, and that multiple route servers may support a single edge device. This many-to-many mapping provides redundancy for the route server function and eases administration, since MPOA route servers are simply configured to join the virtual LAN for the edge devices they serve. MPCs and MPSs use LANE bridging for communication within their own emulated virtual LANs as illustrated in Figure 19-8. Normally, packets flow over the default-routed path until the MPOA client recognizes a long-lived flow that justifies the multiple NHRP and ATM SVC messages required to set up the shortcut path. Once the MPOA client resolves the ATM address of the shortcut next hop using the NHRP protocol, it establishes an SVC for the shortcut path as shown at the bottom of the figure.

The stimulus for establishing the shortcut could be data driven when the MPOA client recognizes a long-lived packet flow, for example, transfer of a large file from a server. The driver for establishing the shortcut could also be control driven, for example as determined by layer 3 topology information learned by an MPOA server. The price performance trade-off of data-driven forwarding is strongly dependent upon the distribution of flow duration and the computing necessary for the NHRP and SVC message processing. On the other hand, the price performance trade-off of a control-driven approach is dependent upon the amount of layer 3 topology information and the rate at which it changes. At a high level, the notion of MPLS forwarding equivalence classes (FEC) and

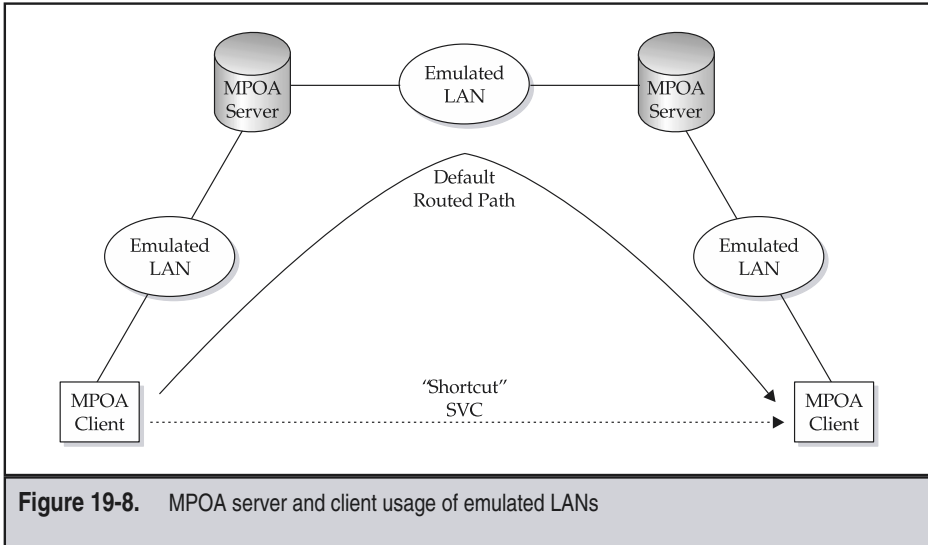


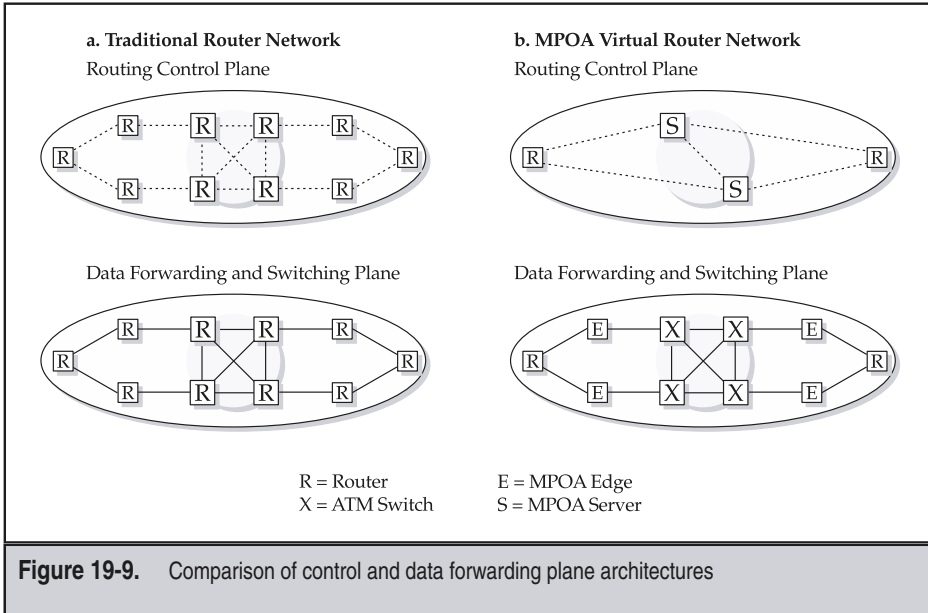
Figure 19-8. MPOA server and client usage of emulated LANs

the establishment of label switched paths (LSPs) based upon IP routing information is an example of a control- or topology-driven approach.

A good way to visualize the potential benefit of the virtual router concept of the MPOA architecture is to compare the distribution of function in separate logical planes for routing control and data forwarding and switching [Riley 97]. Figure 19-9a shows how the physical structure of a typical routed network; such networks often employ edge and backbone routers arranged as a hierarchy to support even moderate-sized networks. Every machine processes the network routing protocol in the control plane, as well as forwards and switches packets as shown in the data plane. Figure 19-9b illustrates the same physical network topology implemented using ATM switches, MPOA servers, and MPOA edge devices. Note how traditional routers on the edge of the network peer with the MPOA servers in the routing control plane.

Lessons Learned from MPOA

Users with large ATM-based emulated LANs supporting thousands of clients were the principal potential customers of MPOA. Some vendors announced MPOA products, often with proprietary extensions, but there was never a significant deployment of this technology. As discussed earlier, an important factor that stunted widespread adoption of ATM-based LANs was the availability of more cost-effective, high-speed, easier-to-operate Ethernet interfaces and higher performance routers. Furthermore, MPOA was a rather complex protocol, and the prospects of vendor interoperability were never very promising.



However, several concepts from MPOA have been adopted or are being considered in other efforts. As has been discussed, the use of MPOA in the control-driven (sometimes also called topology-driven) establishment of ATM SVC shortcuts is similar to the concept of establishment of MPLS LSPs in support of IP FECs, as described in Chapters 10 and 14. At the time of writing, the IETF Forwarding and Control Element Separation (FORCES) working group was working on a similar concept of defining an interface between routing control and forwarding to facilitate a similar separation.

IP Multicast over ATM

The capability of allowing one address to be able to broadcast to all other addresses in the group, effectively emulating a LAN, is very useful in the exchange of LAN topology updates, ARP messages, and broadcast information for use by routing protocols.

Overview of IP Multicast

RFC 1112, later updated by RFC 2236, defines IP multicasting as a service where a datagram sent by one host is sent to all other hosts in the multicast group. Hosts dynamically join and leave specific multicast groups by sending Internet Group Management Protocol (IGMP) report messages to the targeted multicast group. IGMP is a required part of the IP protocol and is supported on the specific range of Class D IP addresses from

224.0.0.0 to 239.255.255.255. One of these addresses uniquely identifies a particular multicast group.

Often a router maps the IP multicast to either an Ethernet or Token Ring–hardware-based multicast capability. Multicast routers also form multicast groups based upon IGMP messages received from their own subnets and selectively direct multicast traffic using particular routing techniques, such as Protocol Independent Multicast (PIM). In general, routers listen on all multicast group addresses for IGMP messages.

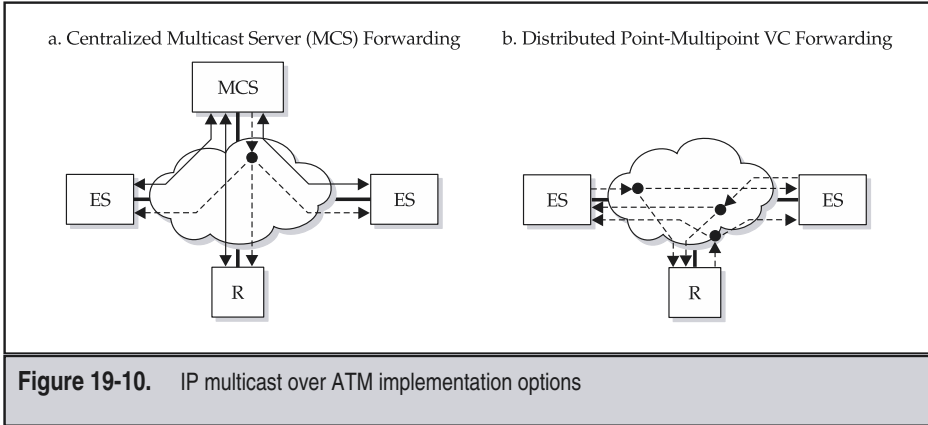
All multicast end systems and routers on a subnet are members of a particular group, called the all hosts group, on IP address 224.0.0.1. All IGMP messages are in IP datagrams with the all hosts address, so that all multicast devices on a subnet receive every IGMP message. Routers send IGMP query messages to poll the status of previously active multicast groups. Any host that is a member of that multicast group waits a random amount of time, and then responds with an IGMP response message, unless some other node responds first. Hence, IGMP avoids congesting the underlying network by efficiently sending join messages once, and polling to detect inactive multicast groups once per minute using only two messages.

IP Multicast over ATM Components and Operation

IETF RFC 2022 specifies the means to implement IP multicast over ATM. The standard utilizes only the Unspecified Bit Rate (UBR), or best-effort, ATM service category. IP over AAL5 has important consequences on the design of a multicast service. Recall from Chapter 12 that in AAL5 *all* cells from a packet must be transmitted sequentially on a single point-to-point or point-to-multipoint VCC. Note that in a point-to-multipoint ATM connection, transmission is strictly from the root to each of the leaves. If the leaves transmitted to the root, then cells could be interleaved from multiple packets when arriving at the root, resulting in AAL5 SAR failures and loss of packet data. Some solutions to this problem were later proposed under the category of VC merge as part of the development of MPLS, as described in Chapter 10.

RFC 2022 defines two methods for implementing the forwarding aspect of IP multicast over ATM, namely a multicast server (MCS) or a full mesh of point-to-multipoint VCs. In the approach illustrated in Figure 19-10a, all nodes join a particular multicast group by setting up a point-to-point connection with the MCS. As shown by a dashed line in the figure, the MCS may have a point-to-multipoint connection to each node in the multicast group. Alternatively, the MCS may emulate the broadcast connection via transmitting packets in the reverse direction on every point-to-point connection. The MCS receives packets from each of the nodes on the point-to-point connections and then retransmits them on the point-to-multipoint connection. This ensures that the serialization requirement of AAL5 is met, that is, all cells of an entire packet are transmitted prior to cells from any other packet being sent.

The full mesh point-to-multipoint connection approach, illustrated in Figure 19-10b, involves establishment of a point-to-multipoint connection between every node in the multicast group. Hence, as seen from inspection of the figure, every node is able to transmit and receive from every other node, avoiding any of the problems with AAL5. Note



that the number of connections required is even more for this relatively simple network. Considering the complexity involved in configuring PVCs for such a network, a point-to-multipoint SVC capability is essential in the practical deployment of an IP multicast over ATM network.

RFC 2022 supports the control aspects of IP multicast using a Multicast Address Resolution Server (MARS) connected via another point-to-point VCC to each end system node. The separate VCC carries registration, address resolution, join, and leave messages between IP/ATM multicast nodes and the MARS, but it never carries multicast application packets. The MARS keeps a mapping of IP multicast addresses to a list of ATM addresses that are currently members of the particular multicast group. The MARS distributes group membership update information over a cluster control VC to all nodes in a multicast group. The cluster control VC is a separate point-to-multipoint VC from the MARS to every node, or when an MCS is present, its control VC to every node can be used instead. IP/ATM multicast nodes use information received from the MARS to establish and release ATM SVCs as appropriate to the MCS or point-to-multipoint forwarding method described previously. Nodes use timers to clear inactive connections caused by lost messages.

The standard also requires that IP/ATM multicast routers join all multicast groups so that it can meet the requirement of listening on all multicast group addresses. Furthermore, all nodes are members of the all hosts multicast group, which is the way that the IGMP query/report protocol for polling multicast group status is implemented. The net result of this protocol and these SVCs is that systems that operate on Ethernet interoperate with ATM end systems implementing the RFC 2022 IP/ATM Multicast protocol.

Lessons Learned from IP over ATM Multicast

The approaches for implementing IP multicast over ATM have advantages and disadvantages. The point-to-multipoint mechanism requires each node to set up and maintain

a connection to every other node in the group, while the multicast server mechanism requires at most two connections per node. Therefore, the point-to-multipoint method places a connection burden on each of the nodes, as well as the ATM network, while the multicast server approach requires that only the server support a large number of connections. This means that the multicast server mechanism is more scalable in terms of being able to dynamically change multicast group membership, but presents a potential bottleneck and a single point of failure.

IP multicast over ATM was implemented by several vendors but was never widely deployed. A significant reason for this lack of adoption is that IP multicast itself is a very complex protocol, which is also not widely deployed. Sometimes cited as a killer application for video or audio conferencing or distribution of broadcast information, IP multicast has yet to find a significant market. Another reason for limited adoption of IP/ATM multicast is that it had no QoS differentiation or traffic reservation over native IP multicast. Since many target applications are essentially broadcast, the IETF has been working on a source-specific multicast protocol in an attempt to simplify the multicast protocol.

IP VIRTUAL PRIVATE NETWORKS (VPN) OVER MPLS OR IP TUNNELS

Virtual networking becomes ever more critical, since it is not only what you know, but also who knows it, that determines the final value of any information. Early in the twenty-first century, there was a significant amount of industry interest and energy in the emerging technology of network-based IP VPNs. It is a topic that has in fact filled entire books, and therefore our coverage in this section is only an overview and summary. The IETF Provider-Provisioned VPN (PPVPN) working group and the ITU-T have developed terminology, requirements, and an architectural framework for VPNs, which form the basis for the material in this section. However, many of the standards are not yet complete. Therefore, the reader interested in up-to-date information and/or further details should consult the cited references, check for updates at the IETF PPVPN page at www.ietf.org, or else consult the following references for more background, details, and examples [PepeInjak 01, Davie 00].

General Virtual Private Network (VPN) Terminology and Concepts

This section provides definitions of terminology and introduces several of the key concepts involved with VPNs based upon IP and MPLS tunneling [PPVPN Requirements, PPVPN Framework, ITU Y.1311]. We begin with a summary of VPN terminology used in this section with reference to Figure 19-11. A *user* is someone or something authorized to access a VPN service, for example, a person at a workstation, a router, or a server, shown as small black dots in the figure. A *site* is a set of users that have local connectivity without use of a provider network, for example, the users that are part of the same enterprise in a

building or campus, shown as ellipses at the left and right of the figure. An *enterprise* is a single organization (e.g., a corporation, or government agency) that administratively controls and sets policy for communication among the set of sites under its control. A *virtual private network (VPN)* is a set of sites that have been configured to allow communication. A set of users at a site may be a member of one or many VPNs. If all sites are part of the same enterprise, then the VPN is often called an *intranet*, while if sites are from different enterprises, then the VPN is often called an *extranet*. The figure shows the case of two VPNs differentiated by the number after the acronym CE at each of the sites.

Continuing the introduction of terminology with reference to Figure 19-11, a *customer edge (CE)* device provides access for users at a site to VPN(s). The CE interfaces to a PE device via an *access connection*, for example, a physical circuit, an FR or ATM VC, switched Ethernet, or the Internet. A layer 2 access connection provides means to further divide a physical interface into subinterfaces, which can then be used to logically partition a physical site into multiple virtual sites. The CE allows users at a site to communicate over the access connection with other sites that are members of the same VPN(s), that is an intranet or one or more extranets. As seen from the figure, a *provider edge (PE)* device faces the service provider core network on one side and interfaces via an access connection to one or more CE devices on the other. A *provider (P)* device is in the service provider backbone. In general, it has little to no knowledge about VPNs. The figure also illustrates some other important real-world network environments. In order to achieve improved reliability for important sites, the CE is often dual-homed to different PEs, as shown in the lower left-hand corner of the figure. The far right-hand side of the figure shows a direct “back-door” link between sites in the same VPN, which is sometimes used in enterprise

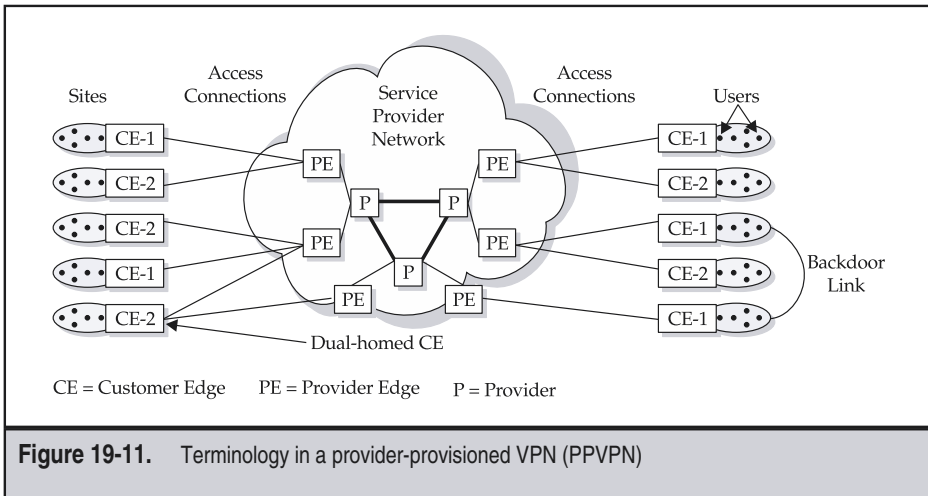


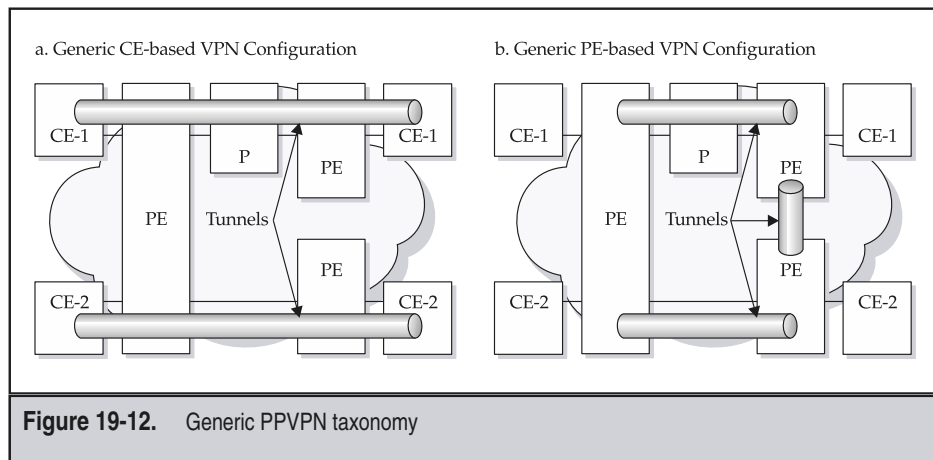
Figure 19-11. Terminology in a provider-provisioned VPN (PPVPN)

networks that use services from multiple carriers to achieve a level of diversity that results in network designs that are highly resilient to many failure modes.

As introduced in Chapter 10, a *tunnel* is formed by encapsulating packets with a header that the provider network uses to forward the original packet to the tunnel endpoint. Encapsulating one tunneled packet within another tunnel header forms a *hierarchical tunnel*, which can significantly reduce the number of tunnels in backbone networks. Protocols under consideration by the PPVPN working group for tunnels are: MPLS [RFC 3031], L2TP [RFC 2661], GRE [RFC 2784], IPsec [RFC 2401], and IP-in-IP [RFC 2003].

The IETF PPVPN taxonomy of VPN types is primarily determined by whether the MPLS or IP tunnels that provide the VPN service terminate on CE or on PE devices [PPVPN Framework, PPVPN Requirements]. ITU-T Recommendation Y.1311 describes the architecture and service requirements for network-based VPNs, which includes optical VPNs. Figure 19-12a illustrates the case of a CE-based VPN where the tunnels terminate on the CE. The CE devices implement the VPN service, since the routers in the service provider network operate only on the tunnel headers, which in this case are usually IP. An example of a CE-based layer 3 VPN is an IPsec-based CPE router-implemented VPN operating over the Internet. This approach for constructing VPNs usually relies on cryptographic techniques for encryption and/or authentication in order to provide privacy across what is often viewed as an open public Internet. This approach saw significant deployment beginning in the late 1990s, and some degree of vendor operability has been achieved. For more information on CE-based IPsec VPNs, see references [Brown 99, Kosiur 98, Fowler 99, McDysan 00b].

We do not cover the CE-based case here since standards in this area currently do not have any plans to employ MPLS, but instead focus on the PE-based VPN case illustrated in Figure 19-12b. A PE-based VPN requires that devices in the service provider network have knowledge of customer VPNs, such that packets are forwarded only between



customer sites that are part of the same VPN using the customer's address space. A PE-based layer 2 VPN switches link-layer packets between customer sites using the customer's link-layer identifiers, for example, Ethernet MAC addresses, ATM VPI/VCI, or FR DLCI. We covered the topic of PE-based L2 VPNs in Chapter 18, with a particular focus on Ethernet virtual private LAN service (VPLS). A PE-based L3 VPN provides an network layer (e.g., IP) service that forwards packets and exchanges routing information only between customer sites within the same VPN using the customer network's address space. The remainder of this section focuses on the PE-based (also called network-based) IP VPNs.

Network-Based IP VPN Concepts

A network-based IP VPN is a combination of several components within a PE device designed to work together in a specific way to constrain the forwarding of traffic and exchange of routing information to only the set of sites that are part of a VPN. These elements are: a separate forwarding table for each VPN, an extended IP address that supports potentially nonunique address spaces, and a means to constrain distribution of routing information to only those sites of a VPN; all done over the tunnels connecting PE routers described in the previous section.

In a network-based IP VPN, tunnels interconnect a layer 3 Virtual (or VPN) Forwarding Instance (VFI) for each VPN instance in a PE switching router. Figure 19-13 illustrates the principal alternatives considered for use of tunnels to interconnect VFIs in PE-based L3 VPNs. The greatest degree of traffic isolation and guaranteed capacity is achieved when a tunnel is dedicated to each VPN (sometimes called the "pipe" model) by interconnecting VFIs, as shown in Figure 19-13a. The other alternative involves a hierarchical tunnel used between PEs with the outermost tunnel providing PE-PE connectivity and tunnels within these used to segregate traffic between customer VPNs, as shown in

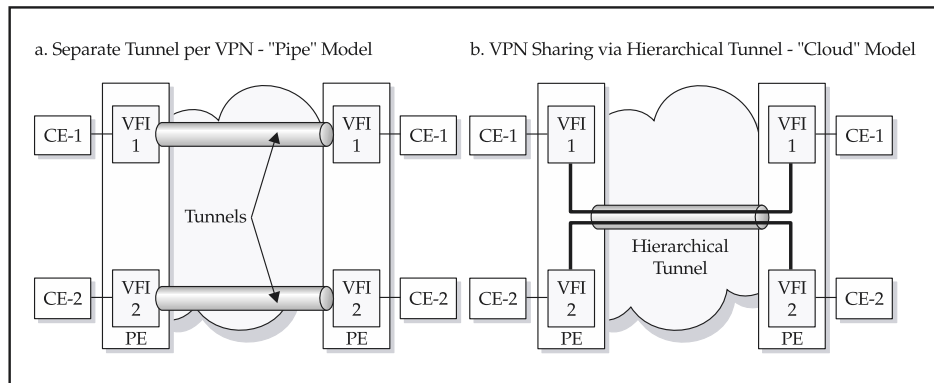


Figure 19-13. Usage of tunnels between virtual forwarding instances (VFIs) in PE devices

Figure 19-13b. The shared hierarchical tunnels between PEs do not dedicate capacity to any one VPN, and hence give more of the appearance of a “cloud” shared by all VPNs. Note that this use of hierarchy can significantly reduce the number of MPLS tunnels in the core of a service provider network.

Figure 19-14 shows a more detailed example of forwarding using a shared hierarchical tunnel between CE devices in two different VPNs using the terminology described in Chapter 11. On the left-hand side of the figure, the CE devices generate IP packets destined for a user associated with the CE devices on the right-hand side of the figure as determined by operation of the routing protocols associated with specific IP address prefix forwarding equivalence class (FEC). On the PE-1 LER, there is a separate forwarding information base (FIB) for each VPN, which in this example is associated with the physical or logical interface on which the CE device for that VPN is attached. In this example, the packet generated for VPN-1 with destination FEC a.b.c/24 and the packet generated for VPN-2 with destination FEC e.f.g/24 are both destined for the same PE, namely PE-2. This means that the VPN FIB entries for the topmost label are identical, since the same tunnel indicated by the thick gray arrow is used by all VPN traffic between PE-1 and

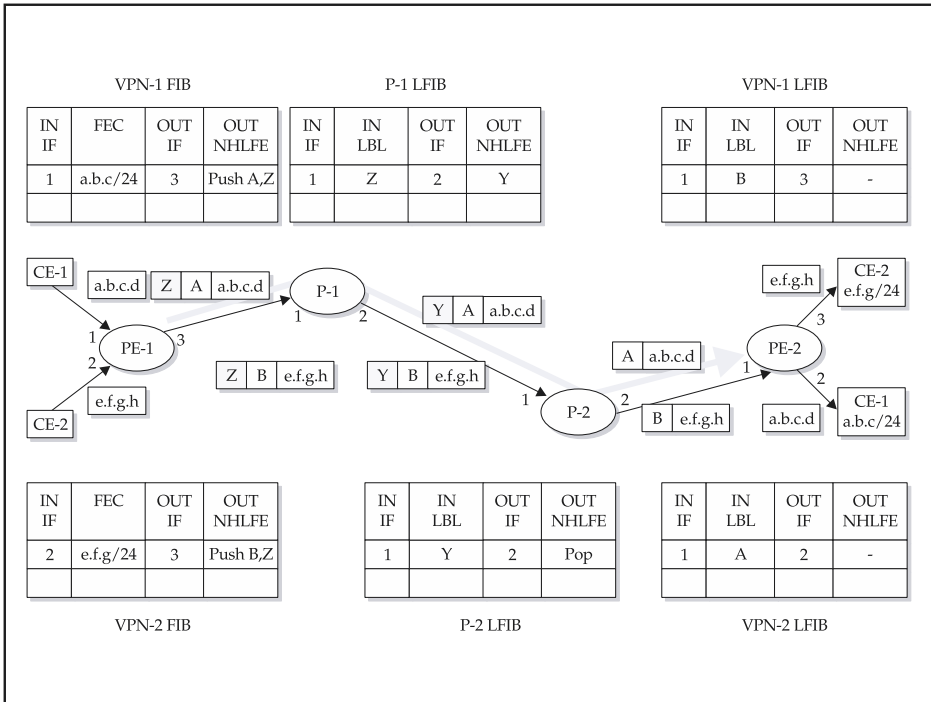


Figure 19-14. Network-based VPN forwarding example using shared hierarchical tunnel

PE-2. What differentiates the VPN traffic is the use of another label at the bottom of the stack, which is put there by pushing it onto the stack first. This is shown in the next hop label forwarding entry (NHLFE) for the VPN-1 FIB by pushing on label A, while the VPN-2 FIB pushes on label B. As these packets traverse the LSP, note that the P-1 label FIB (LFIB) operates only on the topmost label, which it switches from incoming label Z to outgoing label Y. LSR P-2 is the penultimate hop for the LSP between PE-1 and PE-2, and in a manner similar to that used for IP packet forwarding, the operation that it performs is to pop the topmost label from the stack before forwarding the packet on to PE-2. Note that the bottom label remains and is used by the VPN LFIBs in PE-2 to direct the packet to the correct destination CE device. Note that the per-VPN label must be unique to the set of VPNs sharing the hierarchical tunnel.

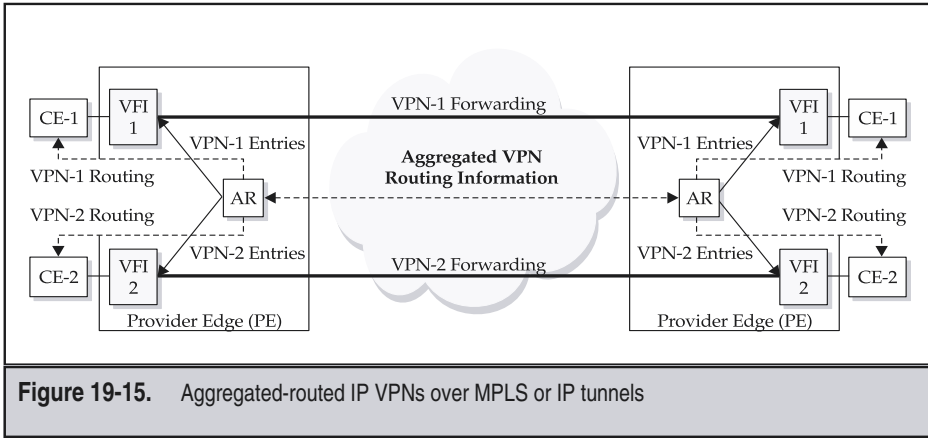
In other words, the bottom label is pushed on by the originating PE to indicate the VPN to which the packet belongs. Since this label is carried through the network to the destination PE, the packet is delivered to the correct VPN, even if the outer tunnel is not MPLS. In this type of network, it is impossible to transmit a packet from one VPN to another unless there is a configuration error of the routing and signaling protocols or there is an error in the forwarding implementation. Obviously, an important aspect of such network-based VPNs is an automatic means for routing and signaling protocols to configure the various forwarding tables in a large network-based VPN, since manual configuration would be prone to error.

There are two generic methods that PE devices can use to exchange routing information to populate the VFIs. The first is where a single instance of a routing protocol in a PE multiplexes routing information in its interchange with another PE in a technique called *aggregated routing*. The second is where there is an instance of a routing protocol for each VPN in an approach called *virtual routing* [PPVPN Framework]. We cover each of these cases in the following sections, since there are significant standards, vendor development, and provider deployment for network-based IP VPNs using each of these approaches.

Aggregated Routing Network-Based VPNs Using Tunnels

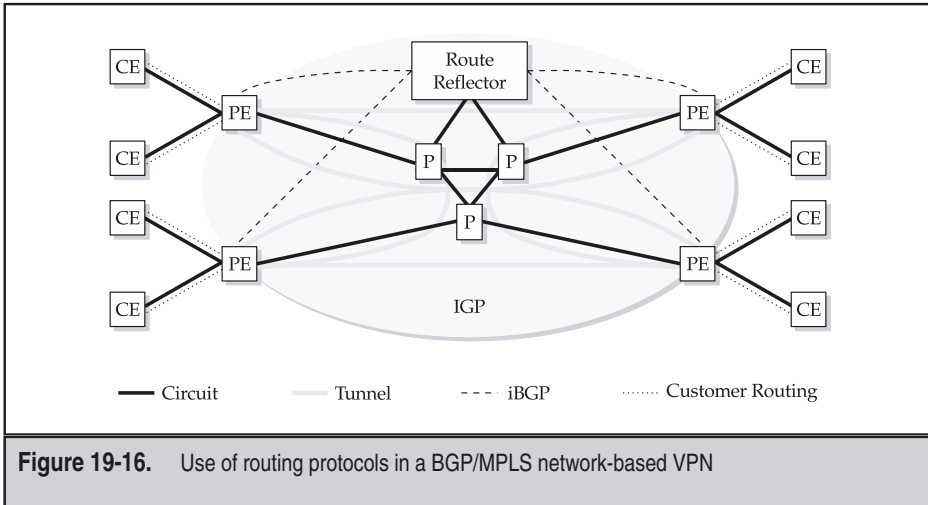
In 1999 Cisco unveiled a high-level description of a standards-based network-based IP VPN in informational RFC 2547. The principal standards involved were a means to aggregate VPN routing information using BGP between a set of PE routers connected via MPLS tunnels, and this gave rise to the title of the RFC, namely BGP/MPLS VPN. At the time of writing work in the IETF PPVPN working group [RFC 2547bis] was detailing the operation of the protocols, defining support over tunnels other than MPLS, and addressing important cases of customer network access to the network and configuration details.

Figure 19-15 shows the functions and their relationships involved in the aggregated-routed IP VPN over MPLS or IP tunnel approach. The figure shows CE devices from two VPNs, numbered 1 and 2 on the far left and right, each connected to a PE device. The virtual forwarding instance (VFI) for each VPN within each of the PEs has a forwarding association with the corresponding VFI in the other PE, as shown by the solid line. The forwarding association may be a tunnel per VPN or a shared, hierarchical tunnel as



described in the previous section. The figure shows interchange of routing information via dashed lines. Facing the CE devices, the aggregated routing function exchanges routing information independently with the CE for each VPN over the access connection using standard IP routing protocols, such as OSPF, BGP, or RIP. Note that from the CE point of view, there is only one routing adjacency with the AR per access connection. Alternatively, the routing information for a particular VPN site could be statically configured in the PE. The information for all customer VPN sites is tagged with a unique identifier and then aggregated for distribution to the other PEs that are part of the same VPN. The exchange of aggregated VPN routing information could occur over the same shared hierarchical tunnel that connects PEs, as shown in Figure 19-13b, or else use IP reachability that exists between the PEs. Obviously, aggregated routing information cannot use per-VPN tunnels. When the AR function receives updated routing information and a routing computation indicates the need to update a forwarding instance, it generates the appropriate entries for the VFI corresponding to that VPN, as shown by the solid arrow between the AR and the VFIs in the figure.

BGP/MPLS VPNs use several protocols in combination to automatically fill in the forwarding tables in the PE and P switch/routers, as shown in Figure 19-16. First, as shown by the shaded oval in the figure, all PE and P routers are in the same IGP, for example, OSPF or IS-IS. Second, the PE knows the address prefixes for each site either via static configuration or dynamically via routing exchange with the CE router, as shown by the dashed lines between the CE and the PE. Finally, the PE routers are also connected by a full mesh of iBGP sessions, or equivalently to one or more route reflectors [RFC 2796] that have an iBGP session with every other PE that has sites that are part of that VPN, the case shown in the figure. One means of scaling this design is that each PE need only support a subset of the VPNs offered by a provider. Route reflection and confederations are other proven techniques that can scale the iBGP portion of the VPN architecture. The end result



of the operation using extended iBGP [RFC 2858] is distribution of attributes and routes to addresses to the VRF in every PE supporting sites within a particular VPN. The aggregated routing function within the PE applies policy using the exchanged attributes to determine what routes are to be downloaded to the VFI table. This approach inherits all of the standard BGP features, such as local policy decisions regarding acceptance an advertised route, or selection from more than one advertisement for a VPN address prefix (e.g., using the route distinguisher). The interchange of extended iBGP routing information creates an association with the IP address prefix(es) of the VPN site as the BGP network layer reachability information (NLRI) and the IP destination address of the PE router as the BGP next hop address. The final step is mapping of the inner and outer MPLS labels to the VFI and destination PE, respectively.

PE routers use either a label distribution protocol (e.g., RSVP-TE, LDP, CR-LDP) or provisioning to create the necessary associations between an MPLS or IP tunnel and the PE IP addresses advertised via the IGP. This results in a full mesh of tunnels over the core P switch/routers that interconnects the PE routers supporting a particular set of VPNs, as shown by the thick shaded lines in the center of the figure. Typically, the VPN provider bases the provisioning of the capacity and path of these tunnels on historical traffic patterns and growth projections. Furthermore, each PE router uses iBGP to distribute a unique MPLS innermost label for each unique VPN-IP prefix of which it is aware. This results in the set of innermost tunnels that interconnect the VFIs for each site of a VPN. Now, the PE router has all the information necessary to fill in the VFI table. Note that in this design, the backbone P switch/routers are completely unaware of the existence of any VPN. However, in order to ensure isolation of VPN traffic when using MPLS, backbone routers must only accept labeled packets corresponding to interfaces on which the

label was distributed for the PE-to-PE tunnels. The hierarchical tunnel approach of BGP/MPLS has superior scalability in the core when compared with approaches that require a tunnel between each VPN site, for example, CE routers overlaid on a connection-oriented network like Classical IP over ATM or MPOA.

The BGP/MPLS VPN technique [RFC 2547, RFC 2547bis] implements routing and support for intranet and extranet VPNs in a flexible manner. A network administrator must assign at least one 8-byte route distinguisher (RD) to each site. The combination of the RD and the IPv4 address prefix makes up the 12-byte *VPN IPv4 address prefix*, which is unique (at least within a provider network). The RD should contain a 4-byte administrator field (e.g., a unique autonomous system (AS) number), and a 4-byte value field. This ensures that the VPN IPv4 address prefixes are globally unique so that VPNs can be implemented across multiple provider networks. The type field determines the format of the value field, which may contain an AS number or an IPv4 address, along with a subfield assigned by the administrator. The PE routers can use extended iBGP along with the standard BGP attributes described in Chapter 14 to advertise these VPN IP address prefixes. The RD does not imply any VPN membership or routing policy; it simply guarantees unique address prefixes, as well as provides a means to associate multiple policies with a particular site. For example, the route distinguisher attribute can be used to create different routes to the same IPv4 address prefix, depending upon whether the source is in the same intranet or comes from a site that is part of a larger extranet. This feature can be used, for example, to allow direct access to a server from intranet sites but to force routing of traffic from extranet sites through another site that has a firewall [RFC 2547bis].

Two additional attributes in BGP that update messages for a particular VPN IPv4 address prefix exchanged between PE routers implement the intranet and extranet capability. These attributes are called *route targets*, of which there are an import target and an export target. The *export target* attribute defines the set of sites that may be part of an intranet or extranet to which a PE router must distribute iBGP Update messages. Although this attribute is similar to the 2-byte BGP community attribute [RFC 1997] defined in the early days of the Internet to implement acceptable use policies, the plan is to base route target attributes on BGP extended communities. These route targets are similar to the RD in the sense that each includes a unique administrator field and an administrator-assigned subfield, which ensures that they are globally unique. A simple intranet requires only the export target attribute, since this completely defines the set of sites that can communicate.

PE routers use the *import target* attribute to provide additional information that further delimits an intranet or extranet using the mechanism whereby the PE supporting a particular site ignores routes other than those configured with the same import target attribute(s) for a VFI corresponding to a particular set of sites. As long as the provider's configuration of the target and origin VPN attributes is aligned with the intent of the customers, this type of solution delivers the isolation aspect of data communications security comparable to a connection-oriented VPN protocol, such as Frame Relay or ATM. Note that a route to a particular (virtual) site can have only one RD but can have multiple route targets. This assignment of multiple route target attributes to a combined RD, and

VPN IPv4 address prefix minimizes the number of routing messages needed to convey VPN membership information. We cover an example of how these route target attributes can be used to construct an extranet later in this section in a comparison of aggregated routed VPNs with VR-based solutions and MPOA.

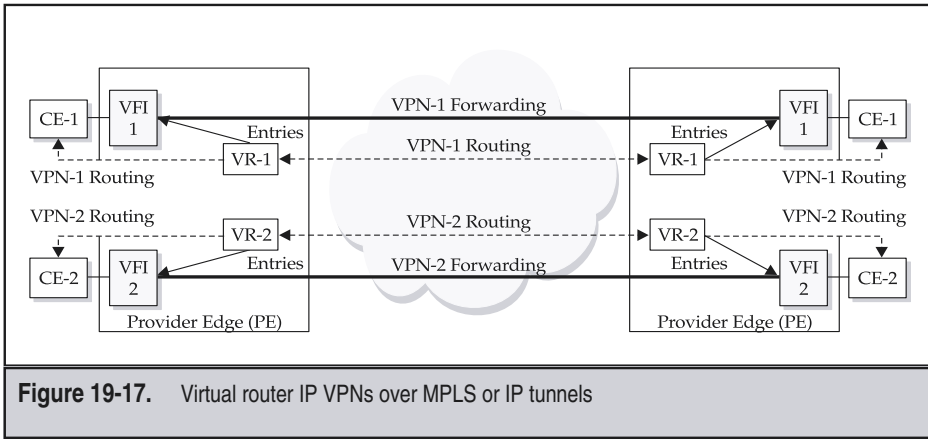
The preceding methods are all that is needed when the routing protocol between the CE and PE is BGP, RIP, or a static route. When the routing protocol is OSPF, several additional things need to happen. First, the PE must implement a separate instance of OSPF for each VPN, rendering it similar to the virtual router approach described in the next section. Second, from the CE perspective, the PE appears to be in the backbone area (i.e., OSPF area 0). Finally, the PE must use an extended BGP attribute to distribute the OSPF link state advertisement values (e.g., administrative weight) to all other OSPF instances in other PEs corresponding to that VPN. This configuration is useful in a hybrid network where CE sites have back-door links, which only the OSPF CE-to-PE routing interface can support.

At publication time, several other efforts and extensions were under way related to application of MPLS- and IP-related technologies to L3 VPNs. An interesting case is where one carrier could resell VPN service to another in what is called a carrier's carrier service [RFC 2547bis, PPVPN Requirements]. Other work in progress includes details in support of a site being able to reach other VPN sites, as well as the Internet over the same access connection, a means to discover and/or authenticate VPN membership, and development of network management infrastructure. See the IETF PPVPN Web site for up-to-date information on these developments.

Virtual Router Network-Based VPNs using Tunnels

Lucent (formerly Ascend) published informational RFC 2917 in 2000 documenting another method of implementing network-based IP VPN, which we summarize in this section using the terminology of the general taxonomy defined by the IETF PPVPN working group. Figure 19-17 shows the functions and their relationships involved in the virtual router (VR) IP VPN over MPLS or IP tunnel approach. CE devices from two VPNs, numbered 1 and 2, are on the far left and right, each with an access connection to a PE device. The virtual forwarding instance (VFI) for each VPN within each of the PEs has a forwarding association with the corresponding VFI in the other PE shown by the solid line, which may be a tunnel per VPN as shown in this example, or a shared hierarchical tunnel as shown in Figure 19-13. The figure shows interchange of routing information via dashed lines. Each logically separate virtual router (VR) instance in a PE peers with one or more CE devices that are in the same VPN using an IP routing protocol (e.g., OSPF, BGP, or RIP) or has a static routing configuration.

Since the VRs are logically separate, nonunique IP addresses may be used independently by each VPN. What is different with the VR approach versus that of aggregated routing is that the routing information interchange for each VPN is kept completely separate, as shown in the center of the figure. Furthermore, no changes are required to any of the protocols operating between VRs. The exchange of per-VPN routing information between the VR instances could occur over per-VPN tunnels or a shared hierarchical



tunnel that connects PEs. When a VR receives updated routing information and a routing computation indicates the need to update a forwarding instance, it generates the appropriate entries for the VFI corresponding to that VPN, as shown by the solid arrow between the VR and the VFI for each VPN in the figure.

In RFC 2917, the virtual routers that make up a VPN are a logically independent routing domain. This means that an enterprise can view each VR instance as another router completely in its VPN, enabling customer control of features like intranet and extranet membership control; traffic engineering; and use of standard IP network management tools like ping, traceroute, and SNMP, as described in Part 7. Like aggregated routed VPNs, this design allows different sites in the VPN to run different routing protocols. The service provider then manages only the layer 1 or 2 interfaces to the VRs on behalf of the customer. Each VR in a PE is allocated certain resources—interfaces to CE routers, route table space, and the like.

The service provider network must run some form of multicast routing in order to implement VR neighbor discovery through use of ARP over an emulated LAN. Routing updates can also be sent over this emulated LAN, reducing the number of adjacencies seen by a VR. Automatic neighbor discovery is important to service providers and customers because it reduces the amount of expensive, time-consuming, and error-prone manual configuration. Similar to RFC 2547, each VPN is assigned a unique identifier (VPN ID), for example, as specified in RFC 2685, to keep this information logically separate in the shared multicast domain.

At publication time, a related network-based IP VPN approach based upon virtual routers was being defined in the IETF PPVPN working group [Knight 02]. In many respects, this approach is similar to the RFC 2917 method just summarized. Specifically, it has a separate VR instance in each PE for each VPN, employs existing IP routing protocols and management tools, uses the RFC 2685 unique VPN identifier, and can use either

per-VPN or shared hierarchical tunnels. The principal differences are that this approach uses different methods to perform automatic discovery and distribution of VPN membership information, explicitly supports tunnels other than MPLS, and is designed to support VR-based VPNs across more than one service provider network. The options being considered for automatic discovery and VPN information distribution were a directory server, use of extensions to BGP, or explicit configuration.

Considerations and Trade-offs with Network-Based IP VPNs

A network-based IP VPN has a number of advantages over other approaches. In general, the number of routing adjacencies seen by each CE is equal to the number of access connections, for example, one for a single-homed site or two for a dual-homed site. This is significantly less than the potentially large number of routing adjacencies seen by a CE router overlaid on top of an FR or ATM PVC mesh or a set of IPsec tunnels over the Internet in a CE-based VPN. In these overlay networks designs, on the order of N^2 PVCs or tunnels are necessary for a network of N CE routers. As a consequence, many of these overlay designs must adopt a hub and spoke architecture. Also, network-based VPNs avoid much of the provisioning required in overlay networks. For example, adding a new site to a fully meshed overlay network with N sites requires adding $N - 1$ new connections or tunnels; but adding a new site to a network-based VPN requires configuration of only the access connection for the new site.

In the BGP/MPLS design, PE routers share a full mesh of tunnels between PEs across the all VPNs, achieving economy of scale and traffic engineering predictability because this mesh is shared across potentially hundreds of enterprise VPNs. Adding a new site requires configuring the new site with only the import and export target BGP attributes, with the routing automatic protocols performing the remainder of the configuration automatically.

In the VR-based designs, each customer has complete control over the configuration and management of his or her own routing. This makes it very difficult for the desired separation of forwarding and VPN routing to be misconfigured. When a separate set of tunnels interconnects the VR instances of a VPN, the customer also has control of the traffic engineering for his or her logically separate network.

On the other hand, a network-based IP VPN also has some disadvantages when compared with other approaches. First, the logical separation ends at the customer edge router, with additional authentication and encryption using IPsec protocols required to provide security within an enterprise LAN. Furthermore, the use of secured tunneling protocols is essential for dial-in access. At the time of writing, availability of network-based VPN technologies was largely limited to a single provider, although a few providers were cooperatively offering such services.

In the BGP/MPLS design, one of the greatest risks to ensuring that forwarding and routing remain separate is that of service provider misconfiguration. A similar situation exists for FR and ATM, but in the BPG/MPLS VPN, once a site is configured to be part of a VPN, the protocol mechanics automatically allow exchange of routing and forwarding information with all of the other sites in the VPN. Another potential issue is the scalability

of this design once the number of VPNs exceeds the capacity of a single PE. Segregating VPNs into disjoint sets may require deployment of more PE routers than would otherwise be required, with commensurate increase in cost and additional administrative effort required to manage the disjoint VPN set assignment to PE routers.

In the VR-based designs, the scalability and stability of software supporting multiple instances of a routing protocol may be a challenging aspect of this technology. This type of design can create a large number of MPLS tunnels in the core of a network when each VPN requires a full mesh. A solution here could be use of hierarchical MPLS traffic engineering in the core, as described in Chapter 21.

Considerations Regarding Choice of Tunnel Type

In the chapters in this part, we have noted that pseudo-wires, layer 2 services, and IP can operate over MPLS- or IP-based tunnels. However, not all tunnels are created equal. Probably the greatest difference is that of efficiency. An MPLS-based tunnel can add as little as four octets of overhead, while an IPv4-based tunnel adds at least 20 octets of overhead. In effect, for a average packet size of 300 bytes, there is an “IP-tunnel tax” that is on the order of five percent. However, if the tunneled traffic is not a significant portion of the overall volume, this inefficiency is less significant.

In many respects, an MPLS tunnel is similar to an ATM or FR VC, which should be no surprise, since, as described in Chapter 11, an ATM or FR VC can implement the topmost label of an MPLS LSP [RFC 3034, RFC 3035]. In this sense, the forwarding isolation or separation security aspects of an MPLS LSP are similar to those of a VC. This is true if routers are configured to accept label distribution only from trusted interfaces. Of course, misconfiguration and widespread deployment of automated MPLS label distribution could reduce this degree of isolation. Also, the MPLS LSP can be traffic engineered, as described in Chapter 14. At the time of writing, the biggest disadvantage of MPLS tunnels was their lack of ubiquity. It is not clear that MPLS will ever be available everywhere that IP is.

On the other hand, IP-based tunnels are not a panacea either. The principal advantage is that IP is ubiquitous, while at publication time MPLS was not. The ubiquity of the Internet is a two-edged sword, however. The good thing about the Internet is that you can reach everyone, but the bad thing is that everyone can reach you. This creates the potential for someone to “spoof” the contents of a valid tunneled packet from anywhere on the Internet and creates the potential for a distributed denial of service (DoS) attack. There are also security concerns that the traffic carried by an IP tunnel could be intercepted, which would compromise confidentiality, and/or modified, which would compromise integrity. Several methods are available to address these issues. The first is use of IPsec tunnel mode encryption and/or authentication, which addresses the spoofing and security issues, but not the DoS attack. The second is use of filtering on IP address prefixes that would prevent packets from unauthorized interfaces from being sent to a tunnel endpoint, which addresses the spoofing and DoS attack issues, but not the security issues. Another technique to address the security issues would be to not advertise the IP addresses of the tunnel endpoints, which minimizes the possibility of hijacking a tunnel

endpoint or directing traffic through a point where it could be intercepted or modified. There is also an issue regarding support for QoS and capacity guarantees over an IP tunnel. QoS could be addressed by Diffserv as described in Chapter 20, with IP over MPLS traffic engineering providing the means to guarantee this QoS for a certain level of capacity.

VPN Representations and Configuration Complexity

Let's look at an example of two different methods for defining an intranet or extranet with reference to Figure 19-18. The figure contains sites for each of three enterprise networks, denoted as A, B, and C. The sites for each enterprise are indicated by a number. The network administrators of these enterprises have agreed to allow communication between their sites on a limited basis. There are several theoretical ways to represent the desired distribution of routes, and hence achievable connectivity, between these sites. The most explicit is that of set theoretic notation, which defines for each VPN the set of sites that can distribute routes to each other. This notation equates a VPN identified by a string of characters to the set of sites contained in braces (e.g., "VPN A"). For example, the notation can be read as follows: VPN A is composed of sites A1, A2, A3, and A4. For the example in the figure, the set theoretic notation showing the sets of sites for each VPN that make up an extranet is the following:

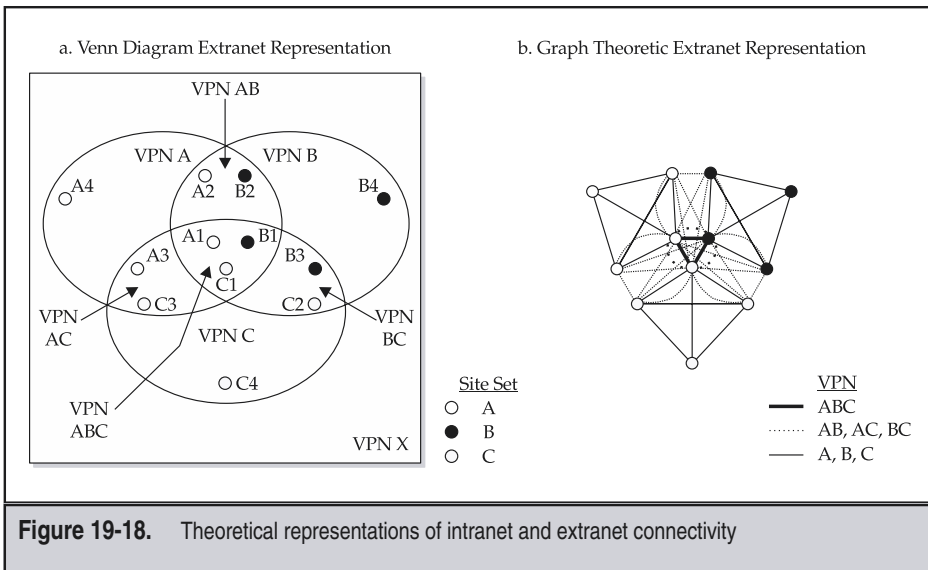
$$\text{VPN A} = \{A1, A2, A3, A4\}$$
$$\text{VPN B} = \{B1, B2, B3, B4\}$$
$$\text{VPN C} = \{C1, C2, C3, C4\}$$
$$\text{VPN AB} = \{A2, B2, A1, B1, C1\}$$
$$\text{VPN AC} = \{A3, C3, A1, B1, C1\}$$
$$\text{VPN BC} = \{B3, C2, A1, B1, C1\}$$
$$\text{VPN ABC} = \{A1, B1, C1\}$$

Although precise, this is a somewhat obtuse means to express the desired connectivity between sites. Other representations are possible, which have the benefit of providing a visual perspective on the VPN connectivity. Although visually appealing for small networks, these notations become cumbersome for large VPNs and the set theoretic notation becomes necessary. The BGP/MPLS aggregated routing approach uses the set theoretic approach through use of the extended BGP attributes described earlier, as illustrated in the Venn diagram of Figure 19-18a. Specifically, the export target BGP attribute is VPN X, which is the set of all sites. The import target VPN attribute that is assigned to the routes for each site corresponds to the VPN identifiers A, B, C, AB, AC, BC, and ABC. In other words, the BGP update message for the route(s) to each site has a list of import target attributes corresponding to an identifier for each VPN that it is a part of. As described earlier, the common export target attribute of VPN X defines the scope of all sites that are

part of the extranet. Since the AR function in each PE accepts routes only from other sites that have the same import target attribute, this achieves the desired connectivity.

On the other hand, the explicitly tunneled VR- and MPOA-based approaches more naturally tend toward a graph theoretic representation, as shown in Figure 19-18b. In this representation, allowed communication is explicitly shown as an arc between each set of sites for each VPN. Note that there is no VPN X in this case. The communication between the sites is indicated by a line style specific to that VPN. We need more dimensions than can be shown in a black-and-white publication, and the use of color would allow the connectivity of the seven VPNs in this example to accurately depict the graph. The line styles shown in the lower right-hand corner of the figure illustrate the sets of nodes in each of the classes of VPNs defined by the preceding set theoretic notation.

It is interesting to compare the amount of configuration information needed in these VPN representations and their associated implementations. For this example, the BGP/MPLS approach requires the assignment of 42 VPN identifiers (i.e., route target attributes) to the PE (sub-) interfaces representing each of the enterprise sites. On the other hand, the connection-oriented explicitly tunneled VR- or MPOA-based approach requires the establishment of 102 unidirectional connections to represent the same level of connectivity. Furthermore, the level of configuration for these approaches scales in markedly different ways. When the VPN definitions contain a number of sites N , the number of configurations for the BGP/MPLS approach scales linearly with N , while the connection-oriented approaches scale quadratically as N^2 . This fact is another reason that



MPOA as not successful, and it provides significant motivation for VR-based VPNs to share tunnels between PEs, unless separate tunnels are necessary for traffic management reasons.

IP PATH MAXIMUM TRANSFER UNIT (MTU) DISCOVERY

Experience shows that IP networks perform much better when the largest possible packet size is used, although different link layer protocols have different maximum transfer unit (MTU) sizes. For example, the minimum required MTU size is 576 bytes for X.25, 1500 bytes for Ethernet, 1600 bytes for FR, and 9180 octets for ATM AAL5. IP fragmentation is often a processor-intensive operation, and therefore, it is highly desirable that fragmentation be done as close to the source host as possible, preferably at the source itself. However, intermediate routers may also use these path MTU discovery procedures as well. RFCs 1191 and 1435 describe how, in order to perform this function, a process called path MTU discovery is employed to determine the smallest MTU along the entire path from source to destination. Once the source determines this smallest MTU size, it performs fragmentation so that other routers along the path do not need to fragment the packet (potentially more than once), and so that reassembly occurs only once at the ultimate destination.

A source begins the path MTU discovery process using the knowledge of the MTU size of the physical medium by which it attaches to an IP network. The source fragments packets to this MTU size and sets the Don't Fragment (DF) bit in the IP header (see Chapter 8). Some hosts use the practice cited in RFC 1191 of fragmenting the packet to the minimum of the interface MTU size and 576 bytes before implementing path MTU discovery, which can result in sending packets smaller than the path can support. Some hosts that don't implement path MTU discovery always send packets of a length less than or equal to 576 bytes; however, fragmentation could occur if an older link technology does not support a 576-byte MTU. Other hosts that do not implement path MTU discovery send packets of the MTU size of the native media if the destination IP address is in the same subnet or the same classful network IP address (see Chapter 8). These latter hosts create the greatest likelihood of fragmentation along the path. If the packet size is larger than the MTU of the next hop interface at a router along the path toward the destination and if the DF bit is set in the IP header, then that router must return an ICMP destination unreachable message to the source. This message contains a code indicating that fragmentation was required, but that since the DF bit was set, fragmentation could not be performed. The message also contains the MTU size of the link. Armed with this knowledge, the host (or router) can now send packets of a size that will not be fragmented anywhere along the path.

MTU Path Discovery over AAL5

Use of the path MTU discovery mechanism is especially important for hosts or routers that have an ATM interface, because the default MTU size for ATM is significantly larger than for older subnet technologies such as Ethernet and FDDI. RFC 2225 mandates that

devices supporting Classical IP over ATM must perform path discovery for this reason. This standard specifies the default MTU size over ATM AAL5 at 9180 bytes, aligning it with the default MTU size for IP over SMDS specified in RFC 1209. Adding the LLC/SNAP header of eight octets results in a default ATM AAL5 protocol data unit MTU size of 9188 octets. RFC 2225 also specifies procedures for use with ATM SVCs that allow dynamic negotiation of larger MTU sizes, up to the AAL5 limit of 65,535 bytes. Larger MTU sizes are more efficient because they minimize AAL5 overhead and IP forwarding processing.

MTU Path Discovery over MPLS

RFC 3032 dedicates a number of pages to the handling of MTU path discovery over MPLS networks. Recall from Chapter 11 that each MPLS label in the stack adds four bytes to the overall packet length. This may not seem like much, but the addition of even a single MPLS header can create the need for fragmentation. For example, consider a 1500-byte packet sent over Ethernet that is forwarded over a single-level stack MPLS LSP as a 1504-byte labeled packet. If the path taken by this LSP traverses a router with an Ethernet interface, then the MPLS label now makes the packet larger than the minimum required Ethernet MTU size of 1500 bytes. Either the router must fragment the packet, or if the DF bit is set in the IP header, it must discard it and generate an ICMP message as specified in the path MTU discovery procedures of RFC 1191. The situation becomes even more unpredictable if multiple levels of MPLS labels are employed.

RFC 3032 addresses these issues by defining several new parameters and procedures. The maximum initially labeled IP datagram size parameter defines the packet size at which an MPLS label edge router (LER) as described in Chapter 11 must fragment IP packets without the DF bit set in their header. For example, if this parameter is set to 1488 bytes, then a 1500-byte packet received on an Ethernet interface would be fragmented into pieces no larger than 1488 bytes. This would allow support for the addition of up to three MPLS labels. If this procedure is used and a labeled datagram is too large, then a label switch router (LSR) may silently discard the packet, since it should have been fragmented at the LER at ingress to the LSP. Optionally, the LSR may fragment the contents of the labeled packet, generate two or more labeled packets with these fragments, and forward these on toward the destination using MPLS.

The means by which an MPLS LSR determines whether labeled datagrams are too large is controlled by the two parameters. The conventional maximum frame payload size is equivalent to the MTU (e.g., 1500 bytes for Ethernet), while the true maximum frame payload size parameter is the actual size that the router can send on a particular interface (e.g., 1504 or more bytes for Ethernet). Using these maximum frame payload size parameters, an LSR may declare a labeled IP datagram too large if it exceeds the conventional size, but the LSR must consider it too large if it exceeds the true parameter. If the DF bit is not set and the datagram is too large, then the LSR must fragment the labeled packet. If the DF bit is set, then the LSR must generate an ICMP message directed back toward the source according to the path MTU discovery procedures of RFC 1191. If the LSR does not

have a route to the source IP address (e.g., if an MPLS tunnel traverses more than one service provider), then RFC 3032 describes additional procedures to handle these cases.

An important implication on an MPLS LSR for the procedures described to work correctly is that the ingress LER must determine the MTU of the entire MPLS tunnel, for example, using path MTU discovery procedures. Furthermore, the ingress LER must send an ICMP destination unreachable message to the source whenever an IP packet with the DF bit set is received that exceeds the MPLS tunnel MTU size. At publication time, the IETF MPLS working group was further detailing procedures related to discovery of MPLS tunnel MTU size when multiple levels of label stacking were employed or when MPLS is carried over different link layer technologies as part of an LSP.

REVIEW

This chapter moved up the protocol stack to the network layer by describing how ATM and MPLS support enterprise-level IP virtual private networks (VPNs). We began with the simplest case, the Classical IP over ATM protocol, which supports a single logical IP subnetwork (LIS) over ATM. The text then summarized the ATM Forum's Multiprotocol over ATM (MPOA) effort, which defined support for an IP over ATM-based infrastructure. We focused on the lessons learned from MPOA and the concepts from MPOA that are still being pursued in other standards or implementations. We also summarized a proposed standard means for how an inherently nonbroadcast ATM infrastructure can support IP multicast over ATM.

The chapter then described the terminology and techniques being defined in the IETF and the ITU to support layer 3 VPNs, specifically support for IP. We began by giving precise definitions for various terms and placed these into a taxonomy that describes whether the solution is customer equipment- or network-based, defines the layer at which the service is provided, and in the case of layer 3 networks, how routing is logically segregated. We remind the reader that these techniques consider MPLS as but one of several possible tunneling protocols. The coverage then discussed the two principal methods of performing per-VPN network-based IP routing, namely that where routing for multiple VPNs is aggregated together by a single routing protocol (e.g., BGP), and the other case where a separate virtual router (VR) instance is defined for each VPN.

The chapter concluded with a summary of how IP ensures efficient operation by avoiding IP packet fragmentation through proper determination and negotiation of the maximum transfer unit (MTU) size for the ATM and MPLS components of IP networks. Since ATM has a large MTU size, ATM hosts and routers must implement these path MTU discovery procedures. We reviewed the MPLS procedures that allow an LSR to determine whether a datagram is too large due to the addition of labels, as well as discover the MTU size for the entire LSP tunnel.

PART V



Quality of Service, Traffic Management, and Congestion Control

This part provides the reader with an application-oriented view of the Quality of Service (QoS) and the traffic contract, as well as controls that provide traffic management. Traffic controls include policing, shaping, measures of congestion, and a range of congestion control responses. First, Chapter 20 summarizes the basic proposition of a traffic contract: a network guarantees capacity with a specified QoS for that portion of the offered traffic conforming to a precisely specified set of traffic parameters. We define the basic QoS parameters in terms of

errors, loss, delay, and delay variation through precise definitions complemented by practical examples. We also look at recent enhancements to the UBR ATM service category and discuss the details of guaranteed frame rate (GFR), a new ATM service. Chapter 21 introduces the basic concepts of traffic and congestion control. We then detail ATM's usage parameter control (UPC), or traffic policing, function using some simple examples followed by the formal leaky bucket definition. This chapter also defines a similar traffic control technique used in IP networks called the token bucket. Next, the text describes how users can employ traffic shaping to ensure that their traffic conforms to the traffic contract. Chapter 22 addresses the topic of congestion control with phases of response ranging from management, to avoidance, and, as a last resort, recovery. This chapter also gives several examples of how popular flow control protocols react in the presence of congestion, including a description of the ATM available bit rate (ABR) closed loop flow control service category and protocol. The text highlights key standards when defining specific traffic management terminology. We explain each concept using analogies and several different viewpoints in an attempt to simplify a subject usually viewed as very complex. Furthermore, the text cites numerous references to more detailed discussions for the more technically oriented reader.

CHAPTER 20



The Traffic Contract and Quality of Service (QoS)

This chapter begins with the formal concept of a traffic contract from ITU-T Recommendation I.371 and the ATM Forum version 4.1 traffic management specification [AF TM 4.1]. It then generalizes this notion to that used in IP and MPLS networks. The philosophy and paradigms are defined by a specific set of terminology engineering to deliver QoS to a well-defined set of cells or packets (e.g., a class or flow) at a traffic level defined by a set of traffic parameters. Next, the text explains the concept of QoS and details the ATM and IP networking standards designed to support QoS. This chapter defines and illustrates the following specific terms and concepts: Quality of Service (QoS), traffic parameters, and conformance checking. Finally, we put all of these detailed terms and concepts back together using the ATM and IP terminology defined in standards. The text also gives some guidelines for choosing traffic parameters and tolerances.

THE TRAFFIC CONTRACT

Within the ATM paradigm, a network establishes a separate traffic contract with the user of each ATM virtual path connection (VPC) or virtual channel connection (VCC). This traffic contract is an agreement between a user and a network at a specific user-network interface (UNI) regarding the following interrelated aspects of any VPC or VCC ATM cell flow:

- ▼ A set of QoS parameters for each direction of the connection
- A set of traffic parameters that specify characteristics of the cell flow
- The conformance-checking rule used to interpret the traffic parameters
- ▲ The network's definition of a compliant connection

The definition of a compliant connection allows an ATM network some latitude in the realization of checking conformance of the user's cell flow. Unquestionably, a connection that conforms precisely to the traffic parameters is compliant. A network may treat a connection with some degree of nonconformance as compliant; however, this definition is up to the network. In this case, the QoS guarantee applies to only the conforming cells. Finally, the network need not guarantee QoS to noncompliant connections, even the conforming cells.

The counterpart for the traffic contract in IP-QoS is the traffic profile. RFC 2475 defines this as "A description of the temporal properties of a flow, such as a token bucket defined by a rate and a burst size."

The concept of a traffic contract also applies to an internal trunk connection that could be either ATM or MPLS. In this case, the parties of the contract are both in the network, but the "user" is a device that requires capacity between certain points while the "network" are the devices that provide the trunk connections. The generalized notion here is that the network delivers a certain level of quality for the trunk connection traffic that is less than or equal to a specified quantity.

The next sections of this chapter define the reference configuration for the traffic contract, Quality of Service (QoS) parameters, traffic parameters, and the conformance checking requirements using either a leaky or token bucket. Chapter 21 describes the details of the leaky and token bucket algorithms.

REFERENCE MODELS

This section provides several ATM and MPLS reference models used in the standards to define a framework within which further details can be specified.

Generic Allocation of Impairments Model

QoS on an end-to-end basis is the perspective most relevant to an end user. However, an end-to-end QoS reference model usually contains one or more intervening networks, each potentially with multiple nodes, as depicted in Figure 20-1. Each of these intervening networks may introduce delay, loss, or errors due to multiplexing and switching, thereby impacting QoS. Furthermore, statistical variations in the offered traffic may result in loss or excessive delay within congested network nodes. Of course, a network can also implement shaping between nodes, or between networks, to minimize the accumulation of delay variation and loss. In principle, the user should not need to know the details about the intervening networks and their characteristics, as long as the connection delivers the end-to-end QoS for traffic that conforms to the parameters of the traffic contract.

As described later in this chapter, ITU-T Recommendation I.356 takes the approach of defining a worst-case concatenation of networks and devices for specifying QoS. In an analogous manner, the IETF Diffserv model defines the aspects of node and network QoS. Therefore, as long as connections across networks stay within these bounds, users experience a consistent level of QoS performance.

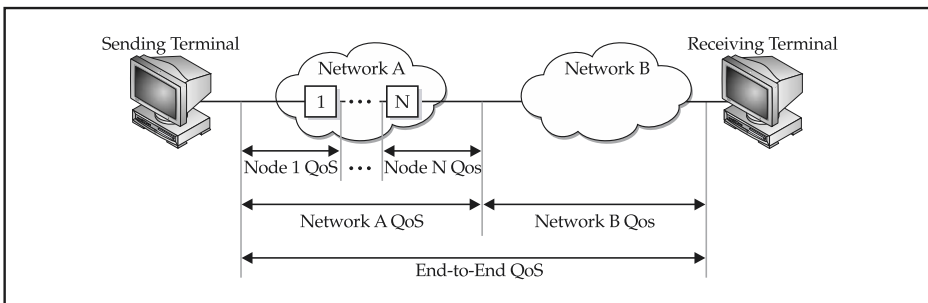


Figure 20-1. End-to-end QoS reference model

ATM Equivalent Terminal Model

The basis of the ATM traffic contract is a reference configuration, which the standards call an *equivalent terminal*. Figure 20-2 illustrates the components and terminology of this reference configuration.

An equivalent terminal need not be a real device and indeed may be a collection of devices, such as an ATM workgroup, or an ATM-capable router, bridge, or hub. Inside the equivalent terminal, a number of traffic sources at the ATM layer—for example, applications running on a workstation with an ATM NIC—generate cells as ATM protocol data unit (PDU) transmit requests to a specific VPC or VCC connection endpoint. The ATM layer inside the terminal multiplexes these sources together, for example using a switching backplane or fabric in a local ATM switch, router, or hub. Associated with the multiplexing function is a virtual traffic shaper that spaces out the cells from each connection to ensure that the cell stream emitted to the physical layer service access point (PHY SAP) conforms to the set of traffic parameters defined in the traffic contract with the network.

After the shaper function, other functions within the terminal equipment (TE) may create variations in the spacing of cells actually transmitted over a physical private ATM UNI interface such that it no longer conforms to the traffic parameters. Standards call the change in cell spacing from that originally transmitted *cell delay variation (CDV)*, a concept detailed in the next section. This ATM cell stream may then traverse other ATM devices in the private network, such as a collapsed ATM backbone, before arriving at the public ATM UNI, potentially accumulating even more CDV. Hence, the network provider defines

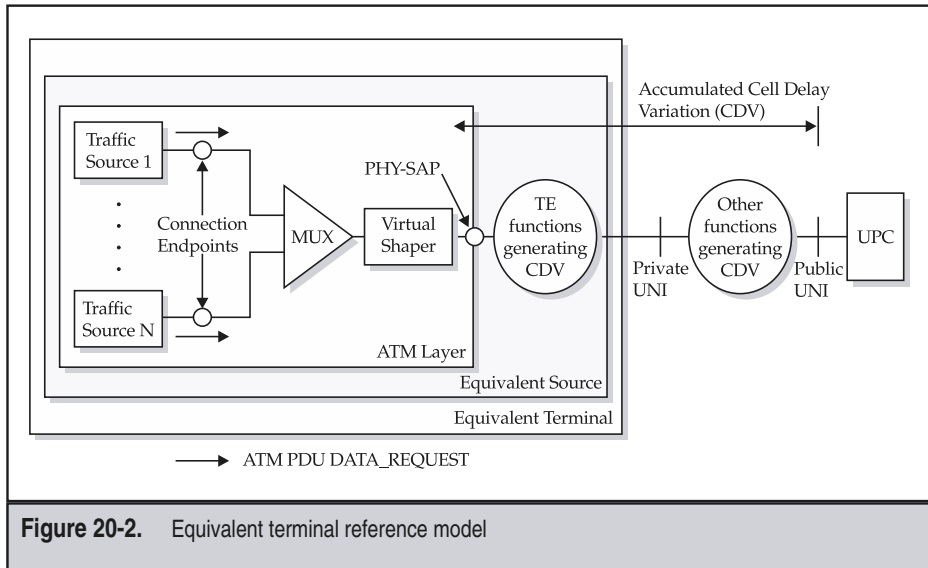


Figure 20-2. Equivalent terminal reference model

a tolerance for this accumulated CDV in the traffic parameters, called *cell delay variation tolerance (CDVT)*. Although these terms are similar, they mean different things: CDV specifies a variation in cell delay arrival times, while CDVT specifies a slack parameter for the peak intercell spacing when the network polices the user's cell stream. These terms are frequently confused, so remember that CDV is a QoS parameter and CDVT is a traffic parameter.

Diffserv Per-Hop and Per-Domain Behavior Models

The Differentiated Services (usually abbreviated as Diffserv) architecture defined in RFC 2475 defines some significant characteristics of packet transmission in one direction across a set of one or more nodes within a network. Therefore, Diffserv is inherently asymmetric. Characteristics can be statistically defined by throughput, delay, delay variation, as well as measures of loss, and of relative priority. The approach taken for Diffserv involves a component involved with forwarding data that is separate from that employed from control components, such as routing, policy administration, and configuration. As we saw in Part 2, many other protocols use this type of architecture that separates the data and control components, such as routing and signaling.

The Diffserv (also abbreviated DS) architecture defines a unique set of terminology, illustrated in Figure 20-3. A *DS-compliant node* utilizes the Diffserv code point (DSCP), the

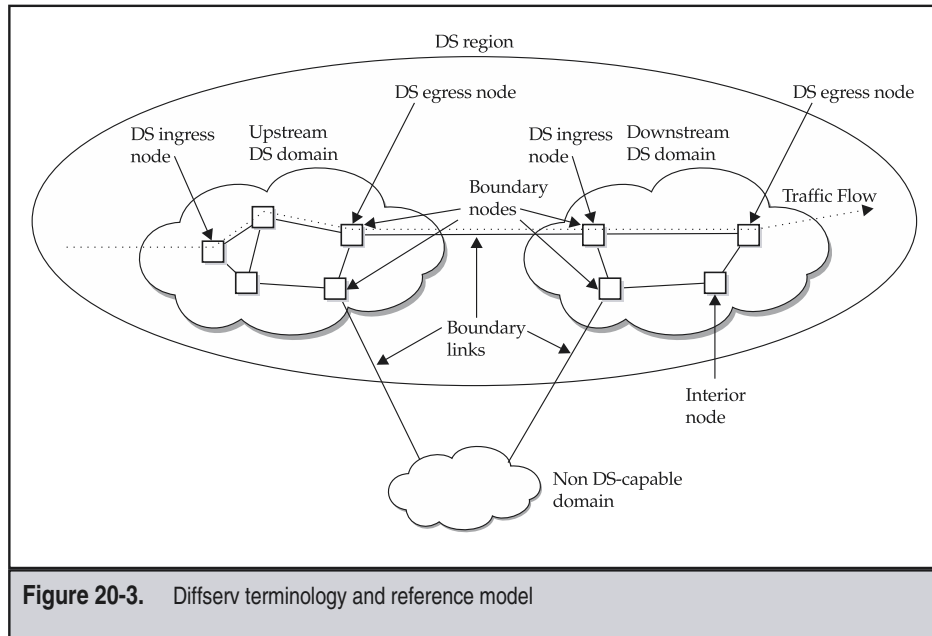


Figure 20-3. Diffserv terminology and reference model

first six bits of the Type of Service (TOS) byte in the IPv4 header or the traffic class byte in the IPv6 packet header (see Chapter 8 and RFC 2474), to determine which externally observable per-hop behaviors (PHBs) to apply to a packet. A *DS domain* is a set of contiguous nodes that implement a common set of PHBs, provisioned in a common manner to deliver a per-domain behavior (PDB) [RFC 3086]. A *DS region* is a set of contiguous DS domains that offer differentiated services. A *DS boundary node* connects via a *DS boundary link* to another DS domain or a non-DS-capable domain. With reference to a particular traffic flow, as shown in Figure 20-3, the DS domain that sends the flow is said to be *upstream*, while the DS domain that receives the flow is said to be *downstream*. The upstream DS domain boundary node that transmits traffic is called a *DS egress node*, while the downstream DS domain boundary node that receives traffic is called a *DS ingress node*.

Typically, ingress, interior, and egress DS nodes perform different functions. These functions include a small set of forwarding per-hop behaviors (PHBs), packet classification, and traffic conditioning functions including metering, marking, shaping, and policing, which the next chapter describes. In fact, a fundamental tenet of the Diffserv architecture is that scalability is achieved by implementing complex multifield classification and traffic conditioning functions at the edge, and then applying the appropriate PHBs within the core solely on the basis of the Diffserv field. We now summarize some other Diffserv-specific terminology from RFC 2475.

A *DS behavior aggregate (BA)* is a collection of packets with the same DSCP value crossing a link in a particular direction.

A *per-hop-behavior (PHB)* is the externally observable forwarding behavior applied at a DS-compliant network device to a DS behavior aggregate. At the time of this writing, the IETF had defined 22 PHBs: 1 for expedited forwarding [RFC 2598]; 12 for assured forwarding composed as four classes, each with three drop precedence values [RFC 2597]; 8 that operate on a class selector [RFC 2474]; and one default or best effort [RFC 2474]. We describe the characteristics and DSCP code points for these PHBs later in this chapter. A *PHB group* is a set of one or more PHBs that can only be meaningfully specified and implemented simultaneously, for example, the drop priorities of the AF PHB.

A *per-domain behavior (PDB)* is the expected treatment that an identifiable or target group of packets will receive from one edge to another of a DS domain [RFC 3086]. A particular PHB (or, possibly, a set of PHBs) and traffic conditioning requirements are associated with each PDB. No PDBs have yet been standardized, but several have been proposed. These include an assured rate PDB based upon the AF PHB, as well as a virtual wire PDB based upon the EF PHB that strives to replace dedicated circuits, and a bulk-handling PDB that is effectively a “less than best effort” class of service.

An important distinction between the ATM reference model and the IP Diffserv architecture is that the ATM model strives to assign numerical values to QoS parameters, while, at the time of this writing, the objective of Diffserv is only to provide differentiated performance. Nonetheless, an IP service provider could assign numerical IP performance parameters to a DS domain, and the performance of a concatenation of such domains may be meaningful. At the time of this writing, the ITU had begun an attempt to quantify IP QoS along these lines, with the results planned for Recommendation Y.1540.

QUALITY OF SERVICE

What is Quality of Service (QoS)? Many definitions exist in the industry, and the concept continues to be refined in the Internet standards. Since ATM precisely defines QoS, we focus primarily on this definition, introducing analogous concepts from IP standards along the way so that the reader can compare and contrast. ATM terminology defines QoS as the performance observed by an end user. For both ATM and IP, the principal QoS parameters are delay, delay variation, and loss. Since applications operate well only within certain performance limits, we begin with this generic view before detailing specific standards.

The ATM Forum's traffic management 4.1 specification and ITU-T Recommendation I.356 define specific Quality of Service (QoS) performance parameters for cells conforming to the parameters of the traffic contract. The IETF has defined a set of IP performance metrics (IPPM) described later in this chapter that define precise means for measuring delay, delay variation, and loss in either a one-way or round-trip context.

Application QoS Requirements

Before we take a detailed look at QoS performance parameters and metrics, it is important to recognize that selecting precise values for QoS parameters like loss, delay, and delay variation is not an easy task. Part of the difficulty arises from the subjective nature of perceived quality [McDysan 00]. The approach employed by both ATM and IP is to group together applications with similar QoS requirements into classes, or service categories, with the appropriate QoS parameters. Figure 20-4 [Woodruff 90, Dziong 97, Onvural 97] illustrates several examples of application-level QoS requirements for the loss and delay variation QoS parameters. These are the principal parameters of interest to the majority of applications. The other major distinction is that a real-time application requires delay bounded by the propagation speed of light over the transmission medium. The bubbles in this chart show a set of ranges that the indicated applications may be able to use. As described later in this chapter, ATM protocols allow applications to specify QoS in two ways: through the generic service categories (e.g., CBR and nrt-VBR) or through explicit enumeration of the QoS parameters in signaling messages. RSVP defines a means to request and confirm specific bounds on delay and delay variation for the guaranteed QoS integrated service, while the controlled load service implies a low level of loss, but no bounds on delay or delay variation. The current Internet standards assume that loss is always low for applications given preferred levels of quality. Indeed, RFC 1633 characterizes applications as real time (e.g., voice), predictive (e.g., broadcast video), or elastic (e.g., file transfer), depending upon the ability to handle variations in delay.

Characteristics of the human body's nervous system and sensory perceptions drive many QoS requirements for delay, loss, and delay variation for voice and video applications. The blink of an eye is approximately one-fiftieth of a second, or 20 ms. Video broadcast and recording systems utilize frame rates of between 25 and 30 video frames per second. When frames are played back at this rate, in conjunction with the image persistence provided by television displays, the human eye-brain perceives this as continuous motion. When lost or errored cells disrupt a few frames in succession, the human

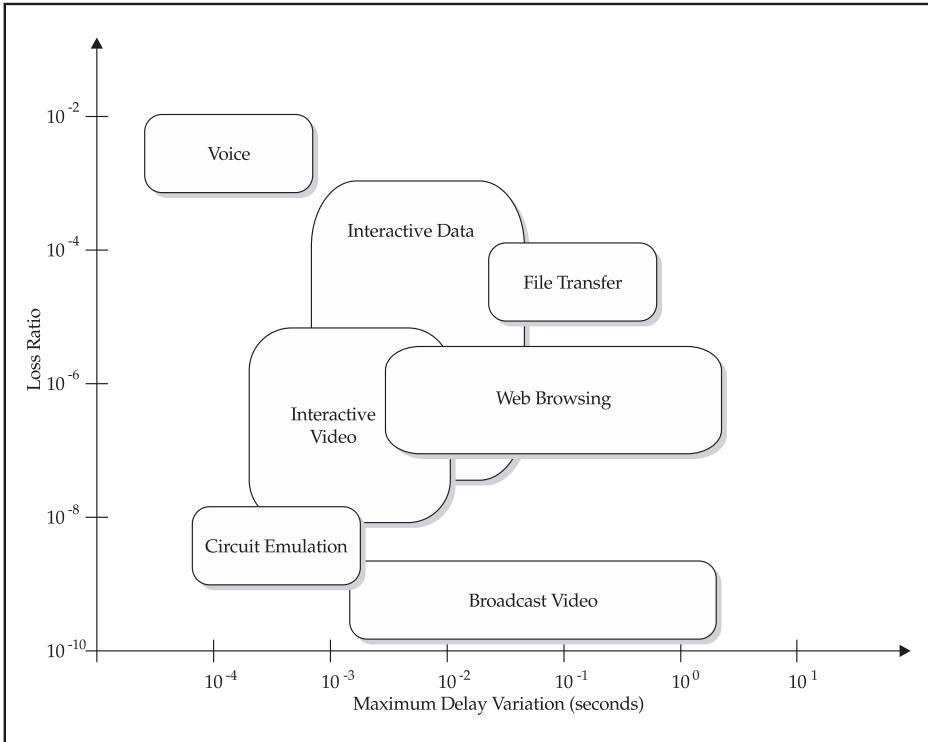


Figure 20-4. Example of application-specific QoS requirements

eye-brain detects the discontinuities in motion, which are subjectively objectionable. Video application requirements depend upon several factors, including the video coding algorithm, the degree of motion required in the image sequence, and the resolution required in the image. Combined video and audio is quite sensitive to differential delays. Human perception is highly attuned to the correct correlation of audio and video, as is readily apparent in some foreign-language dubbed films.

On the other hand, broadcast video can tolerate much longer delays, since a large playback buffer compensates for variations in delay. However, the loss rate must be small, since lost or distorted frames degrade playback quality. One way to compensate for loss is via error correction using AAL1, as described in Chapter 12. A good benchmark for broadcast video transmission is the quality of cable television transmission systems.

The human ear is also sensitive to such differences in delay on a similar time scale. This shows up in telephony, where the reception of a speaker's echo becomes objectionable within less than the blink of an eye; indeed, 50 ms is the round-trip delay where standards require echo cancellation. Delay variation also affects the perception of audio and video. Audio is the most sensitive, since the human ear perceives even small delay variations as changes in pitch. The human ear-brain combination is less sensitive to short dropouts in received speech, being able to accept loss rates on the order of a percent or so.

Many data protocols respond to delay and loss through retransmission strategies to provide guaranteed delivery. One example is the transmission control protocol (TCP), which is widely used to flow control the transfer of World Wide Web text, image, sound, and video files. Data applications are extremely sensitive to loss because they respond by retransmitting information. A user perceives this as increased delay if, for example, retransmissions due to loss extend the time required to transfer large files carrying video or audio images. Since most data networks exhibit significant delay variation (especially over the Internet), applications insert a substantial delay before starting playback. This technique works fine for one-way communication but impedes two-way, interactive communication if the round-trip delay exceeds 300 ms.

Satellite communications for a voice conversation illustrates the problem with long delays: since the listener can't tell whether the speaker has stopped or merely paused, and simultaneous conversation frequently occurs. Most data communication protocols remain relatively insensitive to delay variations, unless the delay varies by values on the order of a large fraction of a second, which causes a retransmission time-out.

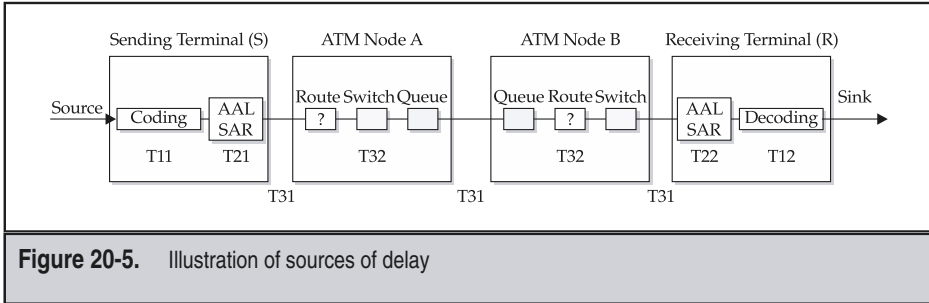
Distributed computing and database applications are also sensitive to absolute delay, loss, and variations in delay. The ideal for these types of applications is access to infinite capacity, together with delay near the speed of light in fiber. A more practical performance model is one comparable to a locally attached disk drive, or CD-ROM, with access times ranging from 10 to 100 ms. In other words, the goal of high-performance networking is to make the remote resource appear as if it were locally attached to the user's workstation.

ATM QoS Parameters

This section details the ATM QoS parameters defined in references ATM Forum TM 4.1 and ITU-T I.356.

Delay and Cell Delay Variation (CDV) Defined

Two key components of ATM QoS are the *cell transfer delay (CTD)* and *cell delay variation (CDV)*. Various components within ATM devices contribute to the statistics of delay within an ATM network, as illustrated in Figure 20-5. In general, fixed and variable delays occur on the sending and receiving sides of the end terminal, in intermediate ATM nodes, as well as on the transmission links connecting ATM nodes. Note that the internal structure of an ATM node need not be identical.



The detailed delay components indicated in Figure 20-5 are

- ▼ T1 = Coding and decoding delay:
 - T11 = Coding delay
 - T12 = Decoding delay
- T2 = Segmentation and reassembly delay:
 - T21 = Sending-side AAL segmentation delay
 - T22 = Receiving-side AAL reassembly/smoothing delay
- ▲ T3 = Cell transfer delay (end-to-end):
 - T31 = Inter-ATM node transmission propagation delay
 - T32 = Total ATM node processing delay (queuing, switching, routing, and so on.)

The principal statistical fluctuations in delay occur due to the random component of queuing embodied in the T32 variable. Other terms contribute to a fixed delay, such as the terminal coding/decoding delays T11 and T12, along with the propagation delay T31. Part 6 covers the effects of statistical cell arrivals, queuing, and switching techniques, on the loss and delay parameters. The interaction of the sources of random and fixed delay depicted in Figure 20-5 result in a probabilistic representation of the likelihood that a particular cell experiences a specific delay. Mathematicians call such a plot a *probability density function*. Figure 20-6 shows this generic function and also indicates the particular ATM Forum QoS parameters for delay and delay variation. Of course, no cells arrive sooner than the fixed delay component, but cells arriving after the peak-to-peak cell delay variation (peak-to-peak CDV) interval are considered late. The user may discard cells received later than this interval; hence, the cell loss ratio (CLR) QoS parameter bounds this area under the probability density curve. The maximum cell transfer delay (maxCTD) is the sum of the fixed delay and peak-to-peak CDV delay components as indicated at the bottom of the figure.

Standards currently define cell delay variation (CDV) as a measure of cell clumping. Standards define CDV either as a single point against the nominal intercell spacing, or as

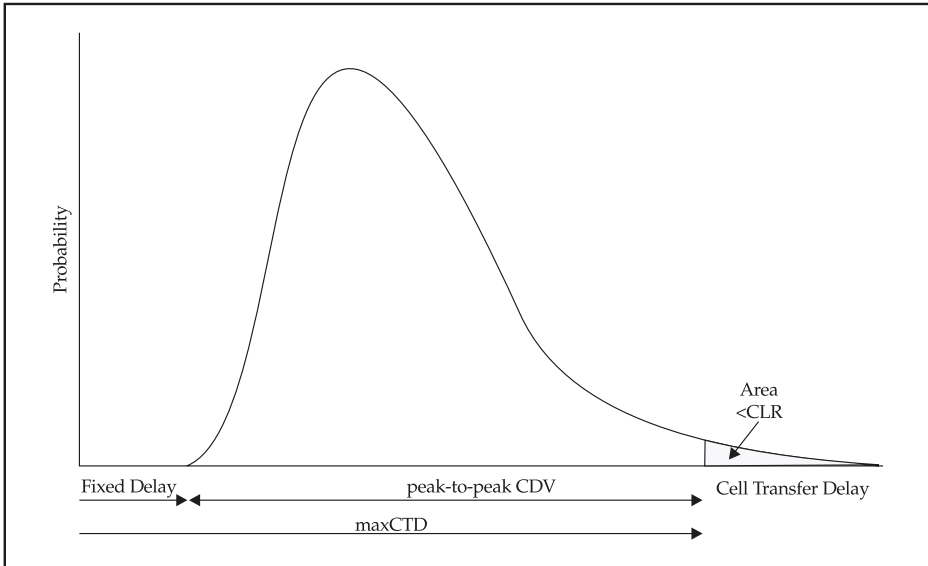


Figure 20-6. Transfer delay probability density model

the variability in the pattern measured between an entry point and an exit point. ITU-T Recommendation I.356 and ATM Forum TM 4.1 cover details on computing CDV and its interpretation. Cell clumping is of concern because if too many cells arrive too closely together, then cell buffers may overflow. Cell dispersion occurs if the network creates too great a gap between cells, in which case the playback buffer would underrun. Chapter 25 illustrates the cases of playback buffer overflow and underrun via a simple example. An upper bound on the peak-to-peak CDV at a single queuing point is the buffer size available to the particular QoS class or connection. This bound results from observing the fact that the worst case (i.e., peak-to-peak) variation in delay occurs between the buffer empty and buffer full conditions. The worst-case end-to-end CDV is the aggregation of the individual bounds across a network.

ATM Cell Transfer Outcomes

Standards define ATM QoS on an end-to-end basis. An end user may be a workstation, a customer premises network, a private ATM UNI, or a public ATM UNI. The following cell transfer outcomes define the various QoS parameters:

- ▼ A *transmitted cell* by the originating user enters the network.
- A *successfully transferred cell* is delivered by the network to the destination user.

- A *lost cell* does not reach the destination user.
- An *errored cell* arrives at the destination but has errors in the payload.
- A *misinserted cell* arrives at the destination, but was not sent by the originator. This can occur due to an undetected cell header error or a configuration error.
- ▲ A *severely errored cell block* occurs when M or more lost, errored, or misinserted cells occur within a received block of N cells.

See the discussion in Chapter 28 regarding ATM performance measurement for further details on cell transfer outcomes.

ATM QoS Parameter Definitions

Table 20-1 lists the ATM layer QoS parameters defined in this section along with their commonly used acronyms. The last column provides an indication of whether the ATM Forum's UNI 4.1 and PNNI 1.1 specifications define a means for the user to negotiate the QoS parameter with a network.

Whether a network will support a desired set of QoS parameters depends on network and switch design, as well as the variation of the network load. Propagation delay dominates the fixed delay component in wide area networks, while queuing behavior contributes to delay variations in heavily loaded networks. The effects of the queuing strategy and buffer sizes dominate loss and delay variation performance in congested networks. A large single shared buffer results in lower loss, but greater average delay and delay variation. The transmission network error rate defines a lower bound on loss. Transmission network characteristics are the leading cause for errors, and hence CER is a QoS parameter common to all QoS classes. Undetected errors in the cell header or configuration errors are the principal cause for misinserted cells. Bursts of errors or intermittent failures are the likely causes of errored blocks.

For all applications, the CER and the CMR must be extremely small, on the order of one in a billion or less. Therefore, the principal QoS parameters are delay (CTD), variation

ATM QoS Term	QoS Parameter Name	Negotiated?
Peak-to-peak CDV	Cell delay variation	Yes
MaxCTD	Maximum CTD	Yes
CLR	Cell loss ratio	Yes
CER	Cell error ratio	No
SECBR	Severely errored cell block ratio	No
CMR	Cell misinsertion rate	No

Table 20-1. Quality of Service (QoS) Parameter Terminology

in delay (CDV), and loss ratio (CLR). As discussed earlier, human sensory perceptions determine the acceptable values of these major QoS parameters, while data communication protocol dynamics define the rest.

Table 20-2 lists the major causes of these QoS impairments [AF TM 4.1]. Note that the offered traffic load and functions performed by the switch primarily determine the delay, variation in delay, and loss parameters. Error statistics largely involve cell errors, but they also include misinsertion events when a cell header is corrupted by several errors so that the cell erroneously appears as valid. Of course, the more switching nodes a cell traverses, the more the quality degrades. All QoS parameters accrue in approximately a linear fashion except for delay variation, which grows at a rate no less than the square root of the number of nodes traversed as discussed in Chapter 25.

The following formulae define the CLR, CER, SECBR, and CMR in terms of the cell transfer outcomes defined earlier. Chapter 28 defines the protocols defined to measure these QoS parameters.

$$\text{Cell Loss Ratio} = \frac{\text{Lost Cells}}{\text{Transmitted Cells}}$$

$$\text{Cell Error Ratio} = \frac{\text{Errored Cells}}{\text{Successfully Transferred Cells} + \text{Errored Cells}}$$

$$\text{Severely Errored Cell Block Ratio} = \frac{\text{Severely Errored Cell Blocks}}{\text{Total Transmitted Cell Blocks}}$$

$$\text{Cell Misinsertion Rate} = \frac{\text{Misinserted Cells}}{\text{Time Interval}}$$

Impairment	CTD	CDV	CLR	CER	CMR
Propagation delay	✓				
Switch queuing architecture	✓	✓	✓		
Switch buffer capacity	✓	✓	✓		
Switch resource allocation/ admission control	✓	✓	✓		
Variations in traffic load	✓	✓	✓		✓
Switch and link failures			✓	✓	

Table 20-2. Mapping of Network Impairments to ATM QoS Parameters

IP Performance Metrics (IPPM)

ITU-T Recommendation I.380 describes a general methodology similar to that summarized in the preceding section for measuring IP packet errors, loss, and availability. The IETF's IP performance metrics (IPPM) working group has precisely defined the following IP QoS metrics within a framework defined in RFC 2330. The general framework is similar to that described at the beginning of this chapter where the performance of an end-to-end application or any network segment is subject to measurement.

- ▼ **Connectivity** As defined in RFC 2678, "Connectivity is the basic stuff from which the Internet is made." More precisely, it determines whether pairs of hosts are instantaneously reachable in either a one-way or bidirectional manner. The metrics defines the basis for determining whether pairs of hosts are reachable over specific time intervals.
- **One-way delay** As defined in RFC 2679, this is the period of time elapsed from the time that a source node sends the first bit of a packet until the destination node receives the last bit of that packet. An important motivation for measuring one-way delay is the fact that packets flowing over an IP network between pairs of hosts may travel over different paths or, if traveling over the same path, may have different performance in opposite directions. Note that measurement of this metric requires not only an accurate frequency reference, but also a synchronized time-of-day clock at both the transmitter and the receiver, for example, as provided by the Global Positioning System (GPS).
- **Round-trip delay** As defined in RFC 2681, this is the period of time elapsed from the time that a source node sends the first bit of a measurement packet until the source node receives the last bit of that packet. The definition assumes that immediately after receipt of the measurement packet, the destination node packet sends it back toward the source node. Although this measurement method cannot account for asymmetric path delays, it does not require time-of-day clock synchronization.
- **One-way packet loss metric** As defined in RFC 2680, a singleton observation of loss may be measured, or a sample based upon a series of singleton transmissions may be collected. Statistics may be derived from a series of measurements of one-way packet loss between a source node and a destination node over a specific interval. Other work is in progress in the IETF to better characterize the loss patterns, for example, a means to quantify the degree of loss bursts.
- ▲ **One-way delay variation** As defined in [ID IPDV], this measurement applies to either a pair of packets or a packet inside a stream of packets. The delay variation for a pair of packets is simply the difference between the one-way-delay measurements for the corresponding packets. Delay variation for an individual packet inside a stream of packets is the difference between the one-way delay of that packet and the one-way delay of the preceding packet in the stream.

TRAFFIC PARAMETERS AND CONFORMANCE DEFINITIONS

This section defines two more components of the traffic contract: the parameters composing the traffic descriptor and the definitions of conformance. Chapter 21 describes the conformance checking (e.g., policing) method, as well as means that a source can ensure conformance (e.g., shaping).

ATM Traffic Descriptor

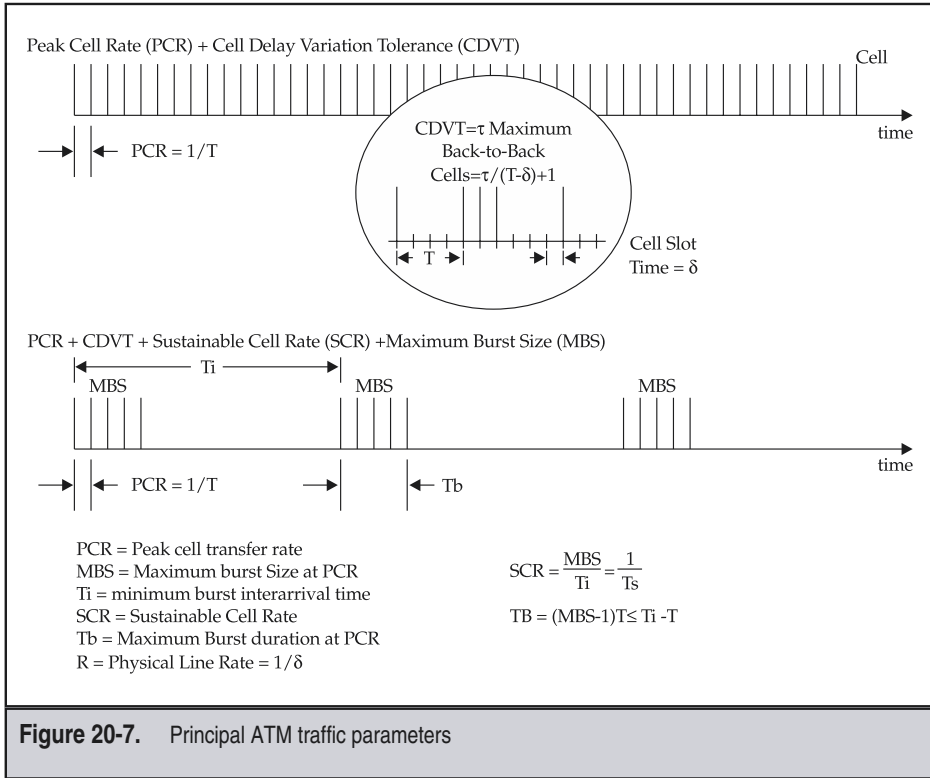
The ATM traffic descriptor is a list of parameters that captures intrinsic source traffic characteristics. It must be understandable and enforceable. This section describes the following traffic parameters defined by the ATM Forum Traffic Management 4.1:

- ▼ A mandatory Peak Cell Rate (PCR) in cells/second in conjunction with a CDV tolerance (CDVT) in seconds
- ▲ An optional Sustainable Cell Rate (SCR) in cells/second (always less than or equal to PCR) in conjunction with a Maximum Burst Size (MBS) in cells

The following statements summarize the ATM Forum's traffic parameter specification. Figure 20-7 illustrates the following traffic contract parameters:

- ▼ Peak Cell Rate (PCR) = $1/T$ in units of cells/second, where T is the minimum intercell spacing in seconds (i.e., the time interval from the first bit of one cell to the first bit of the next cell).
- Cell Delay Variation Tolerance (CDVT) = τ in seconds. This traffic parameter normally cannot be specified by the user but is set instead by the network. The number of cells that can be sent back-to-back at the physical interface rate $R = \delta^{-1}$ (in units of cells/second) is $\tau/(T - \delta) + 1$, for $T > \delta$.
- Sustainable Cell Rate (SCR) = $1/T_s$ is the rate that a bursty, on-off traffic source can send. The worst case is a source sending MBS cells at the peak rate for the burst duration T_b as depicted in Figure 20-7.
- ▲ Maximum Burst Size (MBS) is the maximum number of consecutive cells that a source can send at the peak rate. A Burst Tolerance (BT) parameter, formally called τ_s (in units of seconds), defines MBS in conjunction with the PCR and SCR cell rate parameters, as detailed in Chapter 21.

Figure 20-7 also depicts the minimum burst interarrival time as T_i , which relates to SCR and MBS according to the equations at the bottom of the figure. Specifically, SCR is equivalent to sending MBS cells within an interval of T_i seconds. The maximum burst duration in seconds is given by T_b , which an equation at the bottom of the figure also defines. These definitions are intended only to help the reader in understanding the traffic



parameters—they are not part of the formal traffic contract. See the ATM Forum TM 4.1 specification for the detailed definitions. The ATM Forum UNI 3.0 and 3.1 specifications first defined the Sustainable Cell Rate and Maximum Burst Size in a manner patterned after the PCR definition to better model bursty data traffic. In 1996, ITU-T Recommendation I.371 added a specification for the Sustainable Cell Rate to the previous standard, which defined operation at only the Peak Cell Rate. Modeling of the peak, average, and burst length characteristics enables ATM networks to achieve statistical multiplex gain with a specified loss rate as detailed in Chapter 24.

Figure 20-7, however, does not represent a rigorous definition of the traffic parameters. Standards define a formal, rigorous definition called the Generic Cell Rate Algorithm (GCRA). The next chapter defines this formal algorithm (which has the informal name of leaky bucket).

Traffic Descriptors and Tolerances

ITU-T Recommendation I.371 and the ATM Forum TM 4.1 specification formally define the Peak Cell Rate (PCR) and Sustainable Cell Rate (SCR) traffic parameters in terms of a virtual scheduling algorithm and the equivalent leaky bucket algorithm representation detailed in the next chapter. The user specifies these traffic parameters either in a signaling message or at PVC subscription time, or else the network implicitly defines these parameters according to default rules.

The Peak Cell Rate (PCR) is modeled as a leaky bucket drain rate, and the Cell Delay Variation Tolerance (CDVT) defines the bucket depth as $CDVT + T$ for peak rate conformance checking on either the $CLP = 0$ or the combined $CLP = 0 + 1$ flows.

The Sustainable Cell Rate (SCR) is modeled as a leaky bucket drain rate, and $BT + T_s + CDVT_s$ defines the bucket depth for sustainable rate conformance checking on either the $CLP = 0$, $CLP = 1$, or $CLP = 0 + 1$ flows. $CDVT_s$ is the tolerance parameter associated with the peak rate for the SCR conformance definition. The burst tolerance defines the SCR bucket depth by the following formula from Appendix B of the ATM Forum TM 4.1 specification:

$$\text{Burst Tolerance} = (\text{MBS} - 1) \left(\frac{1}{\text{SCR}} + \frac{1}{\text{PCR}} \right)$$

The burst tolerance, or bucket depth, for the SCR is not simply the MBS because the sustainable rate bucket drains at the rate SCR. The bucket depth allows for MBS cells to arrive at a rate equal to PCR.

Allocation of Tolerances

There are several considerations involved in setting the leaky bucket depths (i.e., traffic parameter tolerances). These differ for the peak rate and the sustainable rate. For the peak rate, the bucket depth should not be much greater than that of a few cells; otherwise, cells may arrive too closely together. Recommendation I.371 defines the minimum CDVT at a public UNI. For a single bucket depth of $CDVT$ seconds and a nominal cell interarrival spacing T , note that approximately $CDVT/T$ cells can arrive back-to-back.

For the sustainable rate, the burst tolerance (or equivalent MBS) should be set to a value greater than the longest burst generated by the user. Of course, the source should shape its traffic to this parameter. The burst length is at least as long as the number of cells corresponding to the longest higher layer Protocol Data Unit (PDU) originated by the user. Furthermore, some transport protocols, such as TCP, may generate bursts that are many PDUs in length; however, the network may not guarantee transfer of extremely large bursts. Chapter 24 describes the relationship between burst length and loss for statistically multiplexed sources. Also, some additional tolerance should be added to allow for the effect of multiplexing and intermediate networks prior to the point that checks traffic conformance.

IP Traffic Descriptor

As described in Chapter 14, since MPLS can use RSVP-derived signaling, the traffic descriptors from RSVP are relevant [RFC 2211, RFC 2212, RFC 2215]. RSVP uses the token bucket algorithm to describe traffic parameters corresponding to a specific flow of IP packets. Two parameters completely specify the token bucket: an average rate r and a bucket depth b . RFC 2215 defines the token-controlled average rate r as the number of bytes of IP datagrams per second permitted by the token bucket. The maximum value of r can be 40 terabytes per second. RFC 2215 defines the bucket depth b in units of bytes with values ranging from 1 byte to 250 gigabytes. The RFCs intentionally specify a large range for these parameters to support capacities achievable in future networks.

The full RSVP traffic specification starts with the token bucket specification and adds three additional parameters: a *minimum-policed unit* m , a *maximum packet size* M , and a *peak rate* p . The packet size parameters, m and M , include the application data and all protocol headers at or above the IP level (e.g., IP, TCP, UDP, RTP, etc.). They exclude the link-level header size. The minimum-policed unit requires that the device remove at least m token bytes for each conforming packet. The parameter m also allows a device to compute the peak packet-processing rate as b/m . It also lower bounds the link-level efficiency by $H/(H + m)$ for a link-level header of H bytes.

Figure 20-8 illustrates the preceding concepts. The worst-case conforming burst is one that begins with a bucket full of tokens and continues with back-to-back packets arriving at the line rate R , and continuing for $T = b/(R - r)$ seconds. In the specific example in the figure, $b = 1000$ bytes, $r = 500$ bytes/second, and $R = 2500$ bytes per second. The lines at the bottom of the figure indicate arriving packets of variable length, while the plot in the figure shows the level of the token bucket. Starting at time 0, the bucket is full of 1000 bytes worth of tokens, and packets arriving back-to-back at the line rate R remove tokens

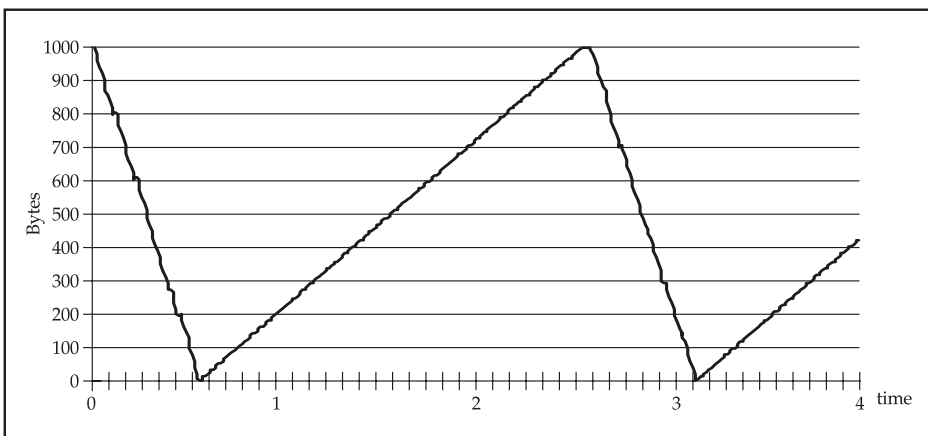


Figure 20-8. Packet-oriented traffic parameters

that are replenished at the rate r . After a period of transmission at the line rate for approximately 0.5 seconds, the burst of packets has emptied to token bucket so that any subsequent packets would be considered non-conforming. After an idle interval of approximately 2 seconds, the bucket is completely replenished with tokens, and another burst of packets at the line rate of approximately 0.5 seconds in duration would be considered conforming. As in the ATM case, this is the worst-case alternating on-off arrival pattern. Other arrival patterns that are less bursty, but still have an average rate less than that determined by the token bucket algorithm would also be considered conforming.

ATM Conformance Definitions

The ATM Forum TM 4.1 specification defines conformance definitions for combinations of leaky buckets. This specification defines the associated traffic parameters; use of the Cell Loss Priority (CLP) bit for tagging non-conforming cells; and as a declaration of the cell flow to which the Cell Loss Ratio (CLR) QoS parameter applies. As noted previously, the terminology of "0" means that the parameter applies to only the CLP = 0 flow, while the terminology "0 + 1" means that the parameter applies to the combined CLP = 0 + 1 flow. Table 20-3 summarizes these conformance definitions.

Conformance Definition	PCR Leaky bucket Flow	SCR Leaky bucket flow	CLP Tagging option active	MCR	CLR On
CBR.1	0 + 1	ns (not specified)	n/a (not applicable)	ns	0 + 1
VBR.1	0 + 1	0 + 1	n/a	ns	0 + 1
VBR.2	0 + 1	0	No	ns	0
VBR.3	0 + 1	0	Yes	ns	0
ABR	0	ns	n/a	Yes	0 ₍₃₎
GFR.1	0 + 1	ns	No	Yes	0 ₍₄₎
GFR.2	0 + 1	ns	Yes ₍₂₎	Yes	0 ₍₄₎
UBR.1	0 + 1	ns	No	ns	Unspecified CLR
UBR.2	0 + 1	ns	Yes ₍₁₎	ns	Unspecified CLR

- 1 With the tagging option, the network may set the CLP bit to 1, but such action does not necessarily imply nonconformance.
- 2 Tagging applicable to all cells of frames deemed ineligible for the service guarantee. Tagging to be applied uniformly to all cells of a frame.
- 3 CLR is low for sources that adjust cell flow in response to feedback. Specific CLR is a network decision.
- 4 CLR is low for frames eligible for the service guarantee. Specific CLR is a network decision.

Table 20-3. ATM Forum TM 4.1 Conformance Definitions

Figure 20-9 illustrates some of these ATM Forum leaky bucket configurations. When there are two buckets, the conformance-checking rule pours an equal amount of cell “fluid” into both buckets when the figures show a diagonal line and a two-headed arrow. The analogy for the diagonal line that the fluid pours over into the buckets is a “rough board” that creates enough turbulence in the fluid such that a single cup of fluid from an arriving cell fills both buckets to the same depth as if the fluid from a single cell was smoothly poured into one bucket.

Tagging is the action of changing the CLP = 0 cell header field to CLP = 1 when the ATM network detects a nonconforming cell. This controls how a bucket acting on CLP = 0 only interacts with a bucket that operates on both CLP = 0 and CLP = 1 flows, referred to as CLP = 0 + 1. We cover tagging in more detail in Chapter 21.

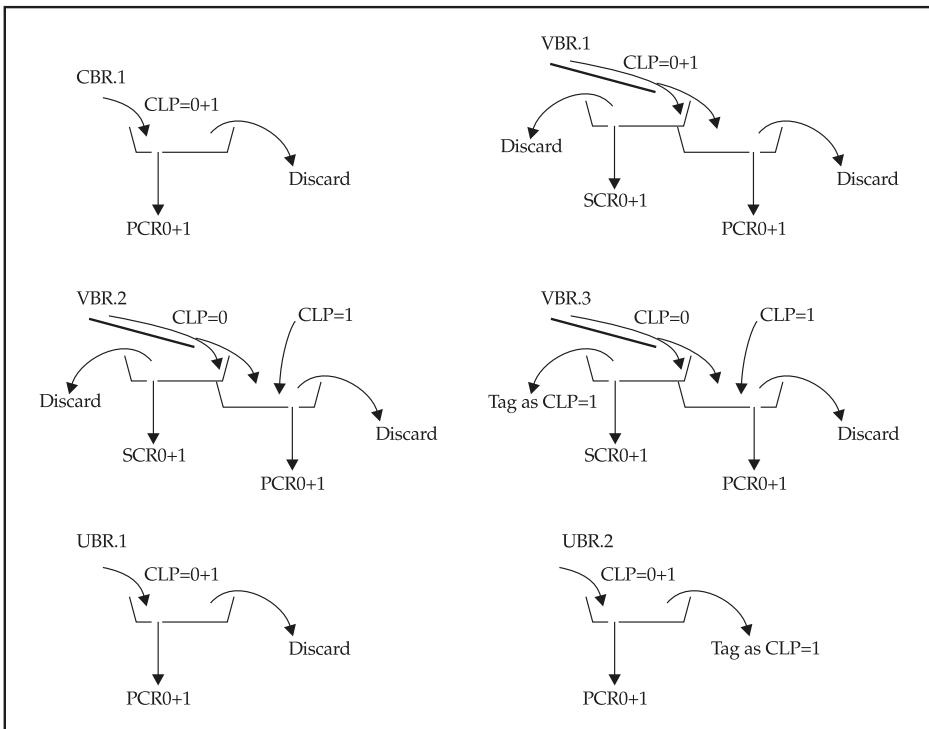


Figure 20-9. ATM Forum leaky bucket configurations

Note that all of these configurations contain the Peak Cell Rate (PCR) on the aggregate $CLP = 0 + 1$ cell flow to achieve interoperability with the minimum requirement from ITU-T Recommendation I.371.

IP Traffic Conformance Definitions

In a manner similar to ATM's combinations of leaky buckets, a few information IETF RFCs have defined combinations of token buckets. RFC 2697 describes two token bucket, both with the same committed information rate (CIR) in bytes per second. The two token buckets have different depths, one has a depth defined by a committed burst size (CBS) and the second defined by an excess burst size (EBS), both parameters defined in bytes. There are two modes of operation: color-blind and color-aware. In the color-blind mode, the meter assumes that arriving packets are uncolored. In color-blind mode, the marker colors an arriving packet of B bytes as follows: it is colored green if B bytes can be removed from the CBS bucket, otherwise it is colored yellow if B bytes of tokens can be removed from the EBS bucket, otherwise it is colored red. In the color-aware mode, the meter assumes that the arriving packets were previously colored as either green, yellow, or red. In color-aware mode, green-colored packets are checked against the CBS and EBS buckets and it remains green if it does not exceed CBS, but are colored yellow if they exceed the CBS, but not the EBS, and red otherwise. Yellow-colored packets are only checked against the EBS bucket and remain yellow if they do not exceed the EBS, but are colored red if they exceed the EBS. Red-colored packets are not checked against either of the token buckets.

RFC 2698 defines a two rate, three color marker. It also uses two token buckets, a first with a CIR fill rate and CBS depth as defined previously, along with a second that has a peak information rate (PIR) with a peak burst size (PBS) depth. In color-blind mode, an arriving B byte packet is marked as follows. It is marked red if B bytes cannot be removed from the PIR bucket, otherwise it is marked yellow if B bytes cannot be removed from the CIR bucket, otherwise it is marked green and B bytes are removed from both the PIR and CIR buckets. In color-aware mode, red-colored packets are unchanged. Any arriving packet is marked red if B bytes cannot be removed from the PIR bucket, otherwise if the packet is marked yellow it is passed, any arriving green packet is marked yellow if B bytes cannot be removed from the CIR bucket, otherwise it is marked green and B bytes are removed from both the PIR and CIR buckets.

Note that the three colors resulting from these markers requires two bits in the DSCP (e.g., as defined for Diffserv AF). Since the MPLS EXP field is only three bits wide, it could carry colored marking—but this would leave only one remaining bit for mapping other Diffserv indications in an E-LSP. On the other hand, an L-LSP has all three EXP bit available to carry markings such as the ones described previously.

CLASSES OF SERVICE

Both ATM and IP standards have employed the notion of a relatively small number of QoS classes. In ATM these are called service categories or transfer capabilities, while in IP Diffserv, these are called per-hop and per-domain behaviors (PHBs and PDBs). The remainder of this section discusses and summarizes these approaches. The coverage on ATM concentrates on the current ATM Forum TM 4.1 and ITU-T I.356 specifications, while the standards for IP and MPLS are based upon the Diffserv standards introduced earlier.

ATM Forum QoS Classes and Service Categories

In order to make things simpler on users, the ATM Forum UNI 3.1 specification defined QoS classes. Each network specified particular values for the QoS parameters for each class, and optionally, an implementation could support individually negotiated QoS parameters on a per connection basis. The ITU-T I.356 still specifies QoS classes, whereas in the ATM Forum UNI 4.1 and TM 4.1 specifications QoS classes are retained only for backward descriptive compatibility, and individual QoS parameters are specified instead.

The ATM Forum's definition of service categories puts all of these acronyms together in a manner meaningful to end-user applications. Each service category definition uses the QoS and traffic parameter terms defined earlier in this chapter. The ATM Forum Traffic Management 4.1 specification defines the following ATM layer service categories:

- ▼ **CBR Constant Bit Rate** The Constant Bit Rate service category is used by connections that request a static amount of bandwidth that is continuously available during the connection lifetime. This amount of bandwidth is characterized by a Peak Cell Rate (PCR) value. The source can emit cells at the Peak Cell Rate at any time and for any duration, and the QoS commitments still pertain. The CBR service category supports real-time applications requiring a fixed amount of bandwidth defined by the PCR. CBR supports tightly constrained CTD and CDV for applications that cannot tolerate variations in delay. Example applications are voice, constant-bit-rate video, and Circuit Emulation Services (CES).
- **rt-VBR Real-time variable bit rate** The real-time VBR service category is intended for applications requiring tightly constrained delay and delay variation. Thus, rt-VBR connections are characterized in terms of a Peak Cell Rate (PCR), a Sustainable Cell Rate (SCR), and a Maximum Burst Size (MBS). Sources are expected to transmit at a rate that varies with time. Equivalently, the source can be described as "bursty." The rt-VBR service may support statistical multiplexing of real-time sources. The rt-VBR service category supports time-sensitive applications, which also require constrained delay and delay variation requirements, but which transmit at a time-varying rate constrained to a PCR and an "average" rate defined by the SCR and MBS.

- **nrt-VBR Non-real-time variable bit rate** The non-real-time VBR service category is intended for non-real-time applications that have bursty traffic characteristics and that are characterized in terms of a PCR, an SCR, and an MBS. For cells that are transferred within the traffic contract, the application expects a low-cell-loss ratio. No delay bounds are associated with this service category. The nrt-VBR service category supports applications that have no constraints on delay and delay variation, but that still have variable-rate, bursty traffic characteristics. The traffic contract is the same as that for rt-VBR. Applications include packet data transfers, terminal sessions, and file transfers. Networks may statistically multiplex these VBR sources effectively.
- **UBR Unspecified Bit Rate** The Unspecified Bit Rate (UBR) service category is intended for non-real-time applications. UBR service does not specify traffic-related service guarantees. No numerical commitments are made with respect to the CLR, or to the CTD. A network may or may not apply PCR to the CAC and UPC functions. Congestion control for UBR may be performed at a higher layer on an end-to-end basis. UBR with MDCR (minimum desired cell rate) can give UBR connections bandwidth assurance. The ATM Forum also calls the UBR service category a “best effort” service, which does not require tightly constrained delay and delay variation, and provides no specific Quality of Service or guaranteed throughput. This traffic is therefore “at risk,” since the network provides no performance guarantees for UBR traffic. Most LANs and IP implementations provide a “best effort” service today. The Internet and local area networks are examples of this “best effort” delivery performance. Example applications are LAN emulation, IP over ATM, and non-mission-critical traffic.
- **ABR Available Bit Rate** ABR is an ATM layer service category utilizing a flow control mechanism that supports several types of feedback to control the source rate in response to changing ATM layer transfer characteristics. This feedback is conveyed to the source through specific control *resource management cells*, or RM-cells. The ABR service does not require bounding the delay or the delay variation experienced by a given connection. ABR service is not intended to support real-time applications. The end system shall specify both a maximum required bandwidth by specifying a PCR, and a minimum usable bandwidth by specifying a MCR. The MCR may be specified as zero. The bandwidth available from the network may vary but shall not become less than the MCR. The ABR service category works in cooperation with sources that can change their transmission rate in response to rate-based network feedback used in the context of closed-loop flow control. The aim of ABR service is to dynamically provide access to bandwidth currently not in use by other service categories to users who can adjust their transmission rates. In exchange for this cooperation by the user, the network provides a service with very low loss. ABR service does not provide bounded delay variation; hence, real-time applications are not good candidates for ABR. Example applications for ABR are LAN interconnection, high-performance file transfers, database archival, non-time-sensitive traffic, and Web browsing. Chapter 22 covers the subject of ABR in detail.

- ▲ **GFR Guaranteed Frame Rate** The GFR service category is intended to support non-real-time applications. It does not require adherence to a flow control protocol. The service guarantee is based on AAL5 PDUs (frames), and, under congestion conditions, the network attempts to discard complete PDUs instead of discarding cells without reference to frame boundaries. The end system specifies a PCR, and a Minimum Cell Rate (MCR) that is defined along with a Maximum Burst Size (MBS) and a Maximum Frame Size (MFS). The GFR traffic contract can be specified with an MCR of zero. The user may always send cells at a rate up to PCR, but the network commits to carry cells only in complete frames at MCR. There are no delay bounds associated with this service category.

The TM 4.0 specification advanced the concept of service categories begun in UNI 3.1 by adding a new category, ABR; splitting the VBR category into real-time (rt) and non-real-time (nrt) components; and better defining the UBR (also known as best-effort) service. Note that the UNI 3.1 Specification contained the traffic management specification; but with UNI 4.0, this content was moved into a separate document: TM 4.0. UNI 4.0 and UNI 4.1 are also referred to as SIG 4.0 and SIG 4.1, indicating that only signaling is specified. Video service requirements largely drove the distinction between rt-VBR and nrt-VBR. The TM 4.1 specification continued enhancements by adding the GFR service category, adding helpful ABR application examples, and specifying differentiated UBR by adding two additional attributes to the UBR service category: a Behavior Class Selector (BCS) and a Minimum Desired Cell Rate (MDCR). This will be further discussed at the end of this chapter.

The ATM service categories use the following QoS parameters: peak-to-peak cell delay variation (peak-to-peak CDV), maximum cell transfer delay (maxCTD), and cell loss ratio (CLR). Each of the service categories has one or more conformance definitions, as defined in Table 20-3. These conformance definitions are distinguished by the manner in which the QoS parameters (particularly CLR) and the traffic parameters apply to the $CLP = 0$ or $CLP = 0 + 1$ cell flows.

Table 20-4 summarizes the QoS, traffic parameters, and feedback attributes for these service categories. Many world-renowned traffic engineering experts participated in the definition and specification of ATM-layer traffic management and service classification taxonomies in the ITU-T, the ATM Forum, and standards bodies around the world. The definition of ATM traffic management and congestion control is now stable within the ITU-T, as defined in conjunction with the ATM Forum, which now enables interoperability in multivendor implementations.

ITU-T ATM QoS Classes

An ATM connection (a VCC or a VPC) is provided with one Quality of Service class (in the ITU-T terminology). A VPC could carry virtual channel links with various specified QoS parameters, and the QoS of the VPC must then meet the most demanding QoS of the virtual channel links carried [ITU-T I.150]. A QoS class can have specified performance parameters (a specified QoS class) or no specified performance parameters (an unspecified QoS class).

ATM-Layer Service Category						
Attribute	CBR	rt-VBR	nrt-VBR	UBR	ABR	GFR
TRAFFIC PARAMETERS						
PCR and CDVT ₁	Specified			Specified ₂	Specified ₃	Specified
SCR, MBS, CDVT ₁	N/A	Specified		N/A		
MCR	N/A				Specified	N/A
MCR, MBS, MFS, CDVT ₁	N/A					Specified
QoS PARAMETERS						
Peak-to-peak CDV	Specified		Unspecified			
MaxCTD	Specified		Unspecified			
CLR	Specified			Unspecified	See note 1	See note 5
CONGESTION CONTROL						
Feedback	Unspecified				Specified	Unspecified
OTHER ATTRIBUTES						
BCS	Unspecified			Optional	Unspecified	
MDCR	Unspecified			Optional	Unspecified	
<p>1 CLR is low for sources that adjust cell flow in response to control information. Whether a quantitative value for CLR is specified is network specific.</p> <p>2 May not be subject to CAC and UPC procedures.</p> <p>3 Represents maximum rate at which the ABR source may send. Actual rate subject to control information.</p> <p>4 CDVT is not signaled. In general, CDVT need not have a unique value for a connection. Different values may apply at each interface along the path of a connection.</p> <p>5 CLR is low for frames that are eligible for the service guarantee. Whether a quantitative value for CLR is specified is network specific.</p>						

Table 20-4. ATM Service Category Attributes begin table footnotes

Specified QoS Classes

A specified QoS class enumerates a set of performance parameters and specifies an objective value for each of these parameters. A QoS class defines at least the following parameters:

- ▼ Cell loss ratio for the CLP = 0 flow
- Cell loss ratio for the CLP = 1 flow
- Cell delay variation for the aggregate CLP = 0 + 1 flow
- ▲ Delay for the aggregate CLP = 0 + 1 flow

The CLP = 0 flow refers to only cells that have the CLP header field set to 0 (high-priority cells), while the CLP = 1 flow refers to only cells that have the CLP header field set to 1. The aggregate CLP = 0 + 1 flow refers to all cells in the virtual path or channel connection.

The initial QoS classes for ATM connections over public networks are defined in ITU-T Recommendation I.356 (Section 8, Table 2/I.356). It is important to emphasize that the QoS classes defined in I.356 apply only to public networks, as opposed to private networks. The I.356 Recommendation considers a private network to be part of customer equipment or networks. In the case where a user is connected to a public network through a private network, the private network should qualitatively interpret the user-specified QoS class in a manner consistent with the user's expectations. In particular, this means that

- ▼ The choice of QoS class 1 by the user indicates that the user desires bounds on the delay parameters and a cell loss ratio commitment on the aggregate cell stream.
- The choice of QoS class 2 by the user indicates that the user desires a cell loss ratio commitment on the aggregate cell stream.
- ▲ The choice of QoS class 3 by the user indicates that the user desires a cell loss ratio commitment on the CLP = 0 cell stream.

A specified QoS class provides performance criteria to an ATM virtual connection (VCC or VPC) defined by a subset of QoS parameters. For each specified QoS class, the network specifies an objective value for each QoS parameter. Note that a particular parameter may be essentially unspecified depending upon the actual objective value assigned—for example, any network meets the objective of a cell loss ratio of 100 percent. In general, an ATM virtual connection may use any one of these QoS classes. However, some higher-layer protocols won't operate very well if the QoS provided by the network is too poor. For example, circuit emulation generally requires QoS class 1 for proper operation.

Unspecified QoS Class

In the unspecified QoS class, no objective is specified for the performance parameters. Therefore, for the unspecified QoS class, there is no explicitly specified QoS commitment on either the CLP = 0 or the CLP = 1 cell flow. Services using the unspecified QoS class may, however, explicitly specify traffic parameters.

An example of the unspecified QoS class is the support of "best effort" service (like UBR), where the user effectively specifies no traffic parameters and does not expect a performance commitment from the network.

QoS in International ATM Networks

ITU-T Recommendation I.356 defines QoS objectives for a reference configuration spanning multiple carrier networks in a related, yet different set of QoS classes. The reference configuration considers transmission distances of up to 27,500 km, which creates a propagation delay of approximately 170 ms. The informative guidelines of Appendix II de-

scribe reference configurations traversing up to five networks containing up to 17 VP switches or 25 VC switches. The analysis assumes that most of the interswitch connections occur at 155 Mbps or higher. However, the objectives apply even with a number of 34/45 Mbps circuits. The I.356 objectives do not include the delay contributed by geostationary satellites.

Table 20-5 shows the numerical values cited in ITU-T Recommendation I.356 for three QoS classes. These are provisional values and not firm requirements, since operational experience may indicate that better (or worse) performance is realistic. Even with this caveat, having quantitative numbers for QoS parameters on a global scale is a significant step forward. QoS class 1 meets the stringent requirements of constant bit rate traffic, while tolerant QoS class 2 addresses applications that do not differentiate between CLP = 0 and CLP = 1 loss. The bilevel QoS class 3 targets applications that expect guaranteed performance on CLP = 0 cells but don't expect guarantees on CLP = 1 cells. For example, QoS class 3 addresses applications where the network uses the CLP bit to tag nonconforming cells. Recommendation I.356 also defines an unspecified class similar to the ATM Forum's definition.

Mapping Between ATM Forum and ITU-T QoS definitions

The service categories of the ATM Forum can be mapped to many of the ATM transfer capabilities in I.371. Some of the ATM service categories of the ATM Forum are equivalent to some of the ATM transfer capabilities in I.371 but have different names: Constant Bit Rate (CBR) is called deterministic bit rate (DBR) in I.371, while Variable Bit Rate (VBR) is

QoS PARAMETER	Notes	QoS Class 1	QoS Class 2	QoS Class 3
CTD	Mean value	400 ms	Unspecified	Unspecified
CDV	2 point at 10^{-8} quantile	3 ms	Unspecified	Unspecified
CLR (0 + 1)	Applies to CLP = 0 + 1	3×10^{-7}	10^{-5}	Unspecified
CLR(0)	Applies to CLP = 0	N/A	Unspecified	10^{-5}
CER	Upper Bound	4×10^{-6}	4×10^{-6}	4×10^{-6}
CMR	Upper Bound	Once per day	Once per day	Once per day
SECBR	Upper Bound	10^{-4}	10^{-4}	10^{-4}

Table 20-5. Recommendation I.356 QoS Class Objectives

called statistical bit rate (SBR) in I.371. The discontinued Recommendation I.362 containing the ITU-T definitions for service classes A, B, C, and D provided a convenient way to associate the ATM Forum and ITU-T QoS classes. However, since QoS classes are no longer specified in ATM Forum specifications and the former ITU-T I.362 service classes are no longer in use, the ATM Forum to ITU-T mapping is not so simple anymore. However, a plausible association has been suggested between ATM Forum service categories [AFTM 4.1] with the ITU-T I.356 classes and the transfer capabilities as shown in Table 20-6, which compares the ATM Forum's TM 4.1 service category terminology in the first column with the analogous ATM Transfer Capability from ITU-T Recommendation I.371 in the third column. The second column indicates the applicable ITU-T QoS class (if one exists). The fourth column lists characteristics of a typical application of the category, or capability. Although these groups chose different names, the close relationships established between the groups resulted in compatible definitions in most major areas. Although the most recent definitions in the ITU-T's Recommendation I.371 and the ATM Forum's TM 4.1 are closely aligned in many areas, they also have important differences, as shown in the table.

ATM Forum Service Category (Conformance Definition)	ITU-T QoS Class	ITU-T I.371 Transfer Capability	Representative Applications
CBR (CBR.1)	1	DBR	Circuit emulation, voice, and video
rt-VBR (VBR.1)	1	SBR.1	Voice and video
rt-VBR (VBR.2)	N/A	N/A	
rt-VBR (VBR.3)	N/A	N/A	
nrt-VBR (VBR.1)	2	Nrt-SBR.1	Packet traffic
nrt-VBR (VBR.2)	3	Nrt-SBR.2	
nrt-VBR (VBR.3)	3	Nrt-SBR.3	
ABR	3, U	ABR (partial)	Adaptable rate sources
UBR (UBR.1, UBR.2)	U	N/A	Best-effort LAN traffic
GFR (GFR.1, GFR.2)	3, U	N/A (Under study)	TCP/IP and Frame Relay-type traffic
N/A	N/A	ABT	Native ABT

Table 20-6. Plausible Association of ATMF Service Categories and ITU QoS Class and Transfer Capabilities

In general, a mapping between the ATM Forum service categories and the ATM transfer capabilities can be made according to Table 20-6, with some discrepancies:

The ATM Forum distinguishes between real-time VBR.1, VBR.2, and VBR.3 and non-real-time VBR.1, VBR.2, and VBR.3, while I.371 specifies non-real-time SBR.1, SBR.2, and SBR.3 and real-time SBR.1 only. The ATM Forum has a unspecified bit rate (UBR) service category that has no equivalent ATM transfer capability in I.371. I.371 partially specifies an ATM block transfer (ABT) transfer capability that has no equivalent in TM 4.1. ABR is also specified in TM 4.1, but the ABR transfer capability is only partially specified in I.371. The ATM Forum finally defines a GFR service category. A GFR ATC is also under study in the ITU-T.

An ATM service category or, interchangeably, ATM-layer transfer capability, represents a class of ATM connections with similar characteristics. Resource allocation defines how much bandwidth a network must allocate to the service category/capability. Possible strategies are assignment of a fixed amount to each connection; statistical multiplexing of connections; or a dynamic, also called elastic, allocation based upon the current state of network utilization.

Table 20-7 illustrates another mapping of ATM Forum service category against the ITU-T transfer capability according to the resource allocation strategy, traffic parameter

ATM Service Category	ATM Transfer Capability	Resource Allocation	Traffic Parameters	QoS Requirement
CBR	DBR	Constant	Peak, or maximum rate	Low CDV, low loss
rt-VBR		Statistical	Peak, average rates and burst size	Moderate CDV, low loss
nrt-VBR	SBR	Statistical	Peak, average rates and burst size	Moderate loss
ABR	ABR	Dynamic	Minimum rate	Low loss
UBR		None, statistical with MDCR	Peak, minimum desirable	No guarantees
	ABT	Per block, or burst	Block size, burst rate	Low loss
GFR		Statistical, complete AAL5 PDU	Minimum rate, peak, burst size, frame size	Low loss

Table 20-7. Mapping of ATM Forum and ITU-T Traffic Management Terminology

characterization, and generic QoS requirements. This mapping illustrates another dimension of the similarities and differences between these two important ATM standards.

Diffserv Per-Hop Behaviors (PHBs)

This section summarizes the IETF standardized Diffserv PHBs, their code points, and important service attributes. Table 20-8 illustrates the mapping of the Diffserv PHB to the DSCP in the IP packet header, as detailed in Chapter 6. The remainder of this section now briefly describes each PHB in Table 20-8 with reference to RFCs that provide further details.

The Expedited Forwarding (EF) PHB [RFC 3246] can be used to build a service with low loss, low delay, and low delay variation. The EF PHB is a forwarding treatment for a particular BA or service class where the configured service rate must equal or exceed the average arrival rate. EF traffic should receive the configured service rate independent of the intensity of any other traffic attempting to transit the same network device. It should average at least the configured rate when measured over any time interval equal to or longer than the time it takes to send an output link MTU-sized packet at the configured rate.

The Assured Forwarding (AF) PHB group [RFC 2597] can be used to provide different levels of forwarding assurance for IP packets transmitted over DS domains. Four AF classes are defined. Each AF class is allocated a certain amount of network resource (buffer space and bandwidth). IP packets in an AF class are marked with one of three possible drop precedence levels. A congested network device tries to protect packets

PHB	DSCP
Expedited Forwarding (EF)	101110
Assured Forwarding (AF) Class 1 (low/medium/high drop precedence)	001010/001100/001110
Assured Forwarding (AF) Class 2 (low/medium/high drop precedence)	010010/010100/010110
Assured Forwarding (AF) Class 3 (low/medium/high drop precedence)	011010/011100/011110
Assured Forwarding (AF) Class 4 (low/medium/high drop precedence)	100010/100100/100110
Class Selector (CS)	xxx000
Precedence Forwarding (PF)	111000 or 110000
Default (Best Effort)	000000

Table 20-8. Diffserv Per-Hop Behavior (PHB) and Code Point (DSCP) Mapping

with a lower drop precedence value in the same PHB scheduling class from being lost by preferably discarding packets with a higher drop precedence values. The packet order must be preserved for AF packets of the same flow.

The set of eight Class Selector (CS) PHBs [RFC 2474] must yield at least two relative forwarding priorities. A PHB selected by a CS DSCP should give packets a probability of timely forwarding that is not lower than that given to packets marked with a CS DSCP of lower value, under reasonable operating conditions and traffic loads. PHBs selected by DSCPs '11x000' must give packets a preferential forwarding treatment over packets receiving the default PHB to preserve the common usage of IP Precedence values '110' and '111' for routing traffic. In addition, the CS PHB requirements on DSCP '000000' are compatible with those listed for the default PHB.

The IP Precedence Forwarding (PF) PHB [RFC 2474] satisfies the IPv4 precedence queuing requirements. Note from Table 20-8 that the PF PHB is a proper subset of the CS PHBs. In the current IP networks, routing packets are usually marked with the IP precedence values of PF PHB.

The Default (Best-Effort) PHB [RFC 2474] is required in a DS-compliant network device. When no other agreements are in place, it is assumed that packets receive the default PHB, which is the common, best-effort forwarding behavior available in existing routers. The default DSCP marking for the default PHB is '000000'. However, when a DSCP is not mapped to any standardized or locally used PHB, it should be mapped to the default PHB, too. The DS-compliant network device must ensure that the default PHB does not starve for bandwidth.

MPLS Support for Diffserv

At the time of writing, the IETF had defined two methods for MPLS to support the Diffserv flavor of IP QoS [ID MPLSDS]. These approaches are called label-based (L-LSP) and EXP field-based (E-LSP). The L-LSP approach basically sets up a separate LSP for each (group of) Diffserv classes; while in the E-LSP approach, the source of the LSP maps DSCPs to the three-bit experimental field in the MPLS shim header, and each MPLS node then performs QoS-related forwarding based upon these bits. Figure 20-10 illustrates a simple example of L-LSP and E-LSP for two groups of Diffserv classes, each served by a separate queue, one for high-priority traffic (e.g., EF) and one for low-priority traffic (e.g., best effort). The L-LSP example of Figure 20-10a shows separate LSPs from node A to node C. The LSP for high-priority traffic traverses the path A-B-C and uses the high-priority queue, as shown by shading in the figure. The LSP for low-priority traffic traverses the path A-D-E-F-C and uses the low-priority queue. In the E-LSP example of Figure 20-10b, a single LSP traversing the path A-B-C carries both high and low priority traffic. Node A maps the DSCPs to the EXP bits and performs queuing based upon the EXP field. In a similar manner, node B also performs queuing based upon the EXP field, shown in the figure by the dark black circle. Note that L-LSP allows a network operator the flexibility to route different traffic over different paths, for example, to meet different latency objectives and balance load in a more granular manner as compared with E-LSP. An advantage of E-LSP is that a smaller number of LSPs is required to support Diffserv.

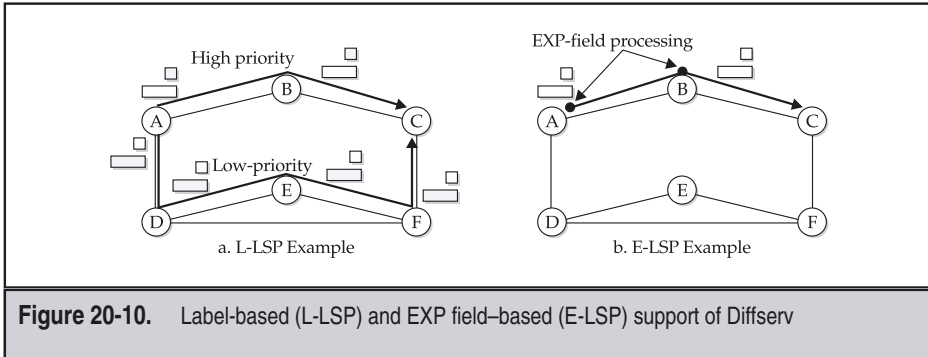


Figure 20-10. Label-based (L-LSP) and EXP field-based (E-LSP) support of Diffserv

COMPARISON OF ATM AND IP QOS AND TRAFFIC PARAMETERS

So, if the IETF's token bucket and ATM's leaky bucket algorithms are simply different ways of expressing a deterministic description of a traffic flow, can they interoperate? The answer is yes, they can, as defined in IETF RFC 2381. This specification defines the mapping of the various parameters and the procedures to invoke them. Table 20-9 illustrates the mapping

Parameter/Protocol	IP	ATM
Signaling protocol	RSVP	SVC (Q.2931/UNI 4.1)
Traffic parameters	Token Bucket peak (p), rate (r), bucket (b) Maximum packet size (M) Minimum policed unit (m)	Leaky Bucket PCR, SCR, MBS
QoS parameters	Delay	CLR, CTD, CDV Loss, Delay, Jitter
Service classes	Best Effort Controlled Load Guaranteed Service	UBR nrt-VBR, ABR CBR, rt-VBR
Conformance	Treat excess as best effort	Mark excess using CLP

Table 20-9. Parameters and Protocol Interworking Between IP's RSVP and ATM

of the signaling protocols, algorithms, and parameters. As seen from the table, most parameters have a one-to-one correspondence. The next chapter details the topics of QoS parameters, service classes, and conformance checking against the traffic parameters.

ATM SERVICE CATEGORIES OPTIMIZED FOR PACKET SWITCHING

The ATM Forum has defined two service categories optimized for the support of packet switching applications, for example, IP and Ethernet. We introduced these categories earlier as Guaranteed Frame Rate (GFR) and Unspecified Bit Rate (UBR).

Guaranteed Frame Rate (GFR)

GFR was envisioned as providing a service that is aware of the ATM AAL5 frame boundaries in order to efficiently support TCP/IP traffic. In general, UBR is already a good match for handling TCP/IP traffic, but it is only suitable for networks where congestion is handled by an upper-layer protocol; otherwise, there is a risk of congestion collapse, as described in Chapter 22.

This potential problem was anticipated early in the UBR specification process, and therefore work was started on the Available Bit Rate (ABR) service, which we detail later in Chapter 22. ABR is based on a rate-based congestion control mechanism implemented in the ATM layer, where the end system periodically sends special resource management (RM) cells within its flow of data cells, providing a feedback loop limiting the source transmission rate. ABR service can provide best-effort service at a Peak Cell Rate (PCR) together with a Minimum Cell Rate (MCR) providing a minimum guaranteed bandwidth, and allowing the end system to always transmit at this rate. The ABR service only supports $CLP = 0$ cells and can provide loss-free operation.

GFR is one of the most recent ATM service categories to address TCP/IP traffic needs. It was first proposed in December 1996, and the final traffic management specification was done in 1999 [AF TM.4.0]. The GFR signaling specification to establish GFR VCs with signaling messages lingered on and was finally released in 2001 [AF GFR1.0].

The main motivation for introducing this new service category was to provide a service that is as easy to use as the UBR service for the end systems, while also providing bandwidth guarantees. Even though ABR fulfilled this need, it was considered complex. GFR does not impose the implementation of complex shapers inside end systems as required in ABR. This, however, might be at the expense of more complex switch implementations relying on per-VC scheduling. Although ABR can be used with simple FIFO switches, the network operator needs to select values for a large number of operational parameters (rate increase factor, RIF; rate decrease factor; RDF; and so on) that influence the behavior of the sources and destinations. The optimal selection of these ABR parameters in a heterogeneous network is not always a simple task. The GFR service category retains the simplicity of UBR (from the end system's point of view) by allowing the end system to transmit cells at the line rate of their ATM adapter. One rather significant drawback

is that the support of GFR in ATM networks will require modifications to existing ATM switches. We will discuss several mechanisms that have been proposed to efficiently support GFR inside ATM switches.

The GFR service category requires the network elements to be aware of the AAL5 frame boundaries and to discard entire AAL5 frames when congestion occurs. UBR does not specify such a strong requirement, even though it is also mainly used for AAL5-based traffic. Most ATM switches today implement frame discard strategies such as early packet discard (EPD) to discard entire AAL5 frames instead of individual ATM cells when congestion occurs. The main distinction from other ATM service categories, is that discard is based on full frames, and GFR attempts to drop entire frames instead of dropping individual cells from several possible different frames. Several studies have confirmed that better TCP/IP throughput can be achieved with GFR [for example see: Hellstrand 98, Ellaumi 98, Bonaventure 98].

The main advantage of GFR is that the quality of service guarantees are provided at the frame level, while they are only provided at the cell level in VBR.3, the conformance used and modified for GFR frames, and its simplicity for the end systems.

The GFR traffic contract is composed of four main parameters (neglecting the cell delay variation tolerances):

- ▼ **Peak cell rate (PCR)** The pcr has the same meaning as in ubr: the maximum rate at which an end system is allowed to transmit. It can be expected that the pcr will often be set at the line rate of the atm adapter.
- **Minimum cell rate (MCR)** The MCR, expressed in cells per second, corresponds to the long-term average bandwidth reserved for the VC inside the network. It is similar to the sustainable cell rate (SCR) in VBR, modified to provide guaranteed bandwidth for AAL5 frames.
- **Maximum frame size (MFS)** The MFS is the largest size of AAL5 frame the end systems can send.
- ▲ **Maximum burst size (MBS)** The MBS places an upper bound on the burstiness of the traffic to which the minimum guaranteed bandwidth applies. This parameter must always be at least equal to

$$1 + [(MFS \times PCR) / (PCR - MCR)]$$

In the GFR setup signaling messages, two additional parameters are defined:

- ▼ **Burst Cell Tolerance (BCT)** A burst cell tolerance (BCT) is to provide a measure on the maximum number of cells eligible for the service guarantee when the connection is served at a rate of MCR cells/s.

$$\begin{aligned} \text{Burst Cell Tolerance (BCT)} &= \text{Minimum Cell Rate} * \text{Burst Tolerance} \\ &= (MBS - 1) * (1 - MCR/PCR) \end{aligned}$$

- ▲ **Acceptable Burst Cell Tolerance (AccBCT)** The AccBCT is the largest Acceptable Burst Cell Tolerance for GFR connections and is a required routing topology attribute. The algorithm used to determine a significant change for AccBCT on a specific network link, and thereby forcing the new calculated value to be advertised in the topology update messages, is identical to the one used for maxCTD [AF GFR1.0].

Table 20-10 summarizes the GFR service parameters, their meanings, and the default values as defined by the ATM Forum.

The logical unit of GFR information is an AAL5 frame. GFR imposes that all the cells of a frame have the same CLP bit. The CLP = 1 AAL5 frames are considered low-priority frames and can be transmitted by the network on a best-effort basis. The minimum guaranteed bandwidth (MCR) is only applicable to the CLP = 0 frames. The intuitive meaning of MCR is that if the end system transmits CLP = 0 AAL5 frames at a rate smaller than or equal to the MCR, these frames should be correctly received by the destination.

There are two GFR conformance definitions: GFR.1 and GFR.2. The only difference between them is whether an F-GCRA, using the VBR.3 GCRA modified to apply to frames, is used to explicitly set the CLP bit to one in the ineligible frames at the ingress of the network or not.

GFR.1

With GFR.1, the network is not allowed to modify the CLP bit of the frames sent by the end systems. The end systems can send CLP = 0 frames in excess of the minimum guaranteed bandwidth; this may include MCR-ineligible frames as well. Therefore, no distinction between an eligible and an ineligible AAL5 frame can be made inside the network; each ATM switch individually must determine which CLP = 0 frames should be transmitted to fulfill the MCR, and what is excess traffic that can be discarded during congestion. Note that GFR does not require that all the frames admitted at the ingress of the network are exactly those that must be delivered to the destination in providing the MCR.

ACRONYM	Meaning	Default Value
PCR	Source's peak cell rate policed by the network	-
MCR	Minimum cell rate guaranteed by the network	0 cells
MBS	Maximum burst size	-
MFS	Maximum frame size	-
BCT	Burst cell tolerance	$2^{24}-1$ CELLS

Table 20-10. GFR Service Parameters

There is a fair bit of ambiguity on what GFR.1 really specifies for frame discard. In one interpretation, all the $CLP = 0$ frames, including the ineligible ones, have greater “importance” than the $CLP = 1$ frames. This implies that all $CLP = 1$ frames should always be discarded before an ineligible $CLP = 0$ frame. Another interpretation is that the $CLP = 1$ frame is always less important than a $CLP = 0$ frame *belonging to the same VC*, but it could be more important than a $CLP = 0$ frame belonging to a different VC. In this case, AAL5 frame discard would be based on the CLP bit of the first cell of the frame, while also accounting for the resource consumption of the corresponding VC.

A user considering using GFR.1 should consult the implementer about what interpretation is used in their solution, since unpredictable results will occur if both implementations exist on an end-to-end GFR connection.

GFR.2

With GFR.2 conformance, the policing function at the ingress of the network uses an F-GCRA to tag the noneligible AAL5 frames. Only the eligible AAL5 frames are accepted as $CLP = 0$ AAL5 frames inside the network. There is now a clear distinction between the eligible ($CLP = 0$) and ineligible ($CLP = 1$) AAL5 frames, and a network can rely on this to decide if an AAL5 frame must be delivered to fulfill the MCR. The F-GCRA algorithm in TM 4.1 is an “ideal” model and may be difficult to implement in policing functions, shapers, or schedulers; therefore, a simple-F-GCRA was created as a slightly simplified version equivalent to the F-GCRA (T,L) for connections containing only conforming frames. Conforming frames are frames containing, at most, MFS number of cells, with the same CLP bit setting, that pass the F-GCRA test.

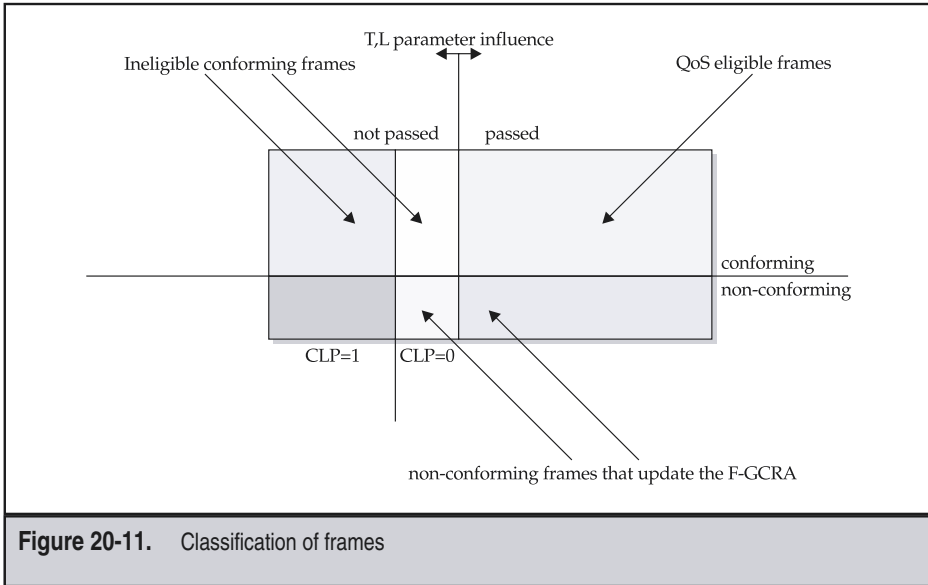
A simple implementation can only perform the AAL5 frame discard mechanisms, relying on the tagging of the ineligible frames performed by an F-GCRA at the ingress of the network. The main advantage of such an implementation is low complexity. The drawback is that it supports only GFR.2 conformance.

A more complex implementation could use one logical queue for each established VC and would then rely on a weighted fair queuing (WFQ)-like scheduler to provide the minimum guaranteed bandwidth.

If a user sends frames with less than MFS cells in a burst below MBS, then the network should deliver the associated frames with minimal loss. GFR allows a user to send traffic exceeding MCR and the associated MBS; but the network does not provide performance guarantees for this excess traffic. The idea is that excess traffic should also be sharing available resources fairly across all GFR users in proportion to the traffic contract. Currently, the GFR service applies only to virtual channel connections, since this is the level at which AAL5 delineates frames.

As shown in Figure 20-11, the set of frames belonging to a particular connection may be classified with regard to the basic concepts of

1. Marking/ CLP tagging
2. Frame conformance
3. Passing the F-GCRA(T,L)



A subset of QoS-eligible frames is built by intersecting passed and conforming subsets. All cells in nonconforming frames whose first cell has $CLP = 0$ are counted by the F-GCRA(T,L) algorithm. These frames count against the MCR, but since they are nonconforming frames, the network may, but is not obliged to, deliver them. The GFR service guarantee is to deliver, with high probability, a number of cells in complete unmarked frames at least equal to the number of cells in conforming frames that pass the F-GCRA(T,L) with parameters $T = 1/MCR$ and $L = BT + CDVT$.

Switch Modifications to Support GFR

TM 4.1 suggests three mechanisms that could be used by a network to provide the per-connection minimum rate guarantees for GFR: tagging, buffer management, and scheduling.

Tagging

Tagging frames at the ingress of the network can be used as a means of reducing the priority of frames that are not eligible for the QoS guarantee before they enter the network. Per-connection network-based tagging requires some per-connection state information to be maintained. This tagging isolates QoS-eligible and non-QoS-eligible traffic of individual connections tagging that other rate-enforcing mechanisms can use in the preferential treatment of QoS-eligible traffic. This would allow buffer thresholds to be used to discard tagged frames and allowing a greater proportion of QoS-eligible frames to enter the network at impending congestion.

Buffer Management

Buffer management is typically used to control the number of frames entering switch or router buffers. Where multiple connections share common buffer space, a per-connection buffer management strategy may use per-connection accounting to keep track of the buffer occupancies of each connection, adding some implementation complexity. Examples of per-connection buffer management schemes are *selective drop* and *fair buffer allocation*.

Scheduling

While tagging and buffer management control the entry of frames into the network, queuing strategies determine how frames are scheduled toward the next hop. In a FIFO queue, frames are scheduled according to entry order, and FIFO queuing therefore cannot isolate frames from multiple connections at the egress of the queue. Per-connection queuing maintains a separate queue for each connection in the buffer and can select between these queues at each scheduling time, again at a cost of added complexity.

Simple Tagging Implementation for GFR.2

We will now look at a simple switch implementation proposed in the TM 4.1 specification based on a modification to a simple buffer acceptance algorithm frequently used to support VBR service in ATM switches. This switch implementation can provide the MCR guarantee by discarding CLP = 1 frames earlier than CLP = 0 frames. The number of CLP = 0 frames, however, needs to be bounded with GFR.2 conformance. The switch would provide MCR guarantees by simply avoiding dropping CLP = 0 frames as much as possible. Figure 20-12 shows this simple switch implementation as an AAL5-aware buffer acceptance algorithm that relies on two buffer thresholds, where the low threshold is used to limit the amount of ineligible (CLP = 1) frames inside the buffer. When the buffer queue occupancy goes above this threshold, the more recently arriving CLP = 1 frames are entirely discarded. The value of the low threshold is chosen as a function of the traffic contract of the established VCs. CLP = 0 frames are entirely discarded when the buffer occupancy reaches the high threshold, but the connection admission control (CAC) algorithm should ensure that this is a rare event. The high threshold is used only to ensure that the switch will not drop individual cells from a frame; its value will usually be close to the buffer size and should not impact performance.

As indicated in Figure 20-12, all the GFR VCs are multiplexed in a single FIFO buffer directly attached to an output link. The advantage of this implementation is that it requires only a global counter for the number of cells in the buffer, and two bits of state information for each VC to discard entire CLP = 1 frames and not individual cells when congestion occurs.

Buffer Management: Counter-Based Implementations

Another similar buffer acceptance algorithm proposed in TM 4.1, still using a single FIFO buffer for all the GFR VCs but maintaining a separate counter for each VC, is used to decide if a new frame can be accepted inside the FIFO buffer or not. A simultaneous update of the counters is performed as a rate function of the MCR of each VC, and the

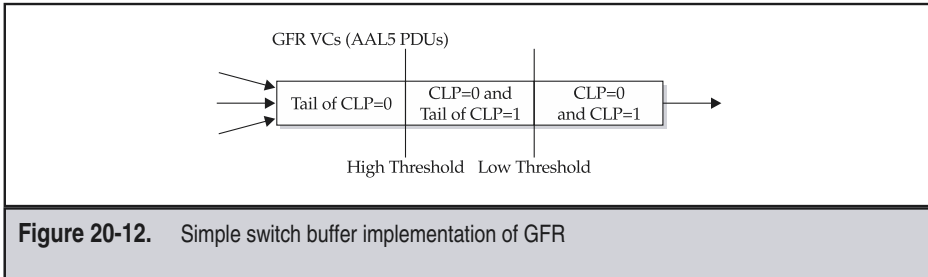


Figure 20-12. Simple switch buffer implementation of GFR

current utilization of the output link. When the first cell of a frame arrives, it is accepted inside the buffer provided that the counter associated with its VC is large enough; otherwise, the entire frame is rejected. Compared with UBR, the main advantage of GFR with a zero MCR is that GFR must take the frame boundaries into account. Although UBR switches should also implement frame-based strategies, this is not a strong requirement. GFR also has a strong conformance requirement with an MCR. UBR does not have any bandwidth guarantees, at least not by adhering to the letter of the UBR specifications. We will look at UBR with a minimum desired cell rate (MDCR) in the next section, and we will find that in practice this implementation is a compelling alternative to GFR.

UBR with BSC and MDCR

The UBR service category has been enhanced to support additional functionality. The first improvement allows the end system to optionally associate a minimum desired cell rate (MDCR) to UBR VCs [AF MDCR 1.0]. This MDCR is similar in spirit to the MCR of GFR, but not normative with a QoS commitment. This means that even when the MDCR is specified, UBR normatively still provides only best-effort service. The MDCR is thus only an indication that the end system can provide to the network. This contrasts with the utilization of the MCR in GFR, since a strong QoS commitment is associated with the MCR. Now you noticed I carefully said “normative.” The UBR with MDCR specification includes an appendix that really suggests a stronger interpretation of QoS, and it is most likely the only method that is useful to implement. We therefore will examine this option closer later in this chapter.

A second improvement allows the end system to attach a *behavior class* to each UBR VC. In such an association, the behavior class is indicated via the behavior class selector (BCS) parameter [AF BCS1.0 AF DIFF1.0]. UBR connections for which no behavior class is indicated are associated with a network-specific default behavior. This information can be used by the network to provide different “classes of service” to VCs with different behavior classes. This modification is intended to allow ATM switches to better support the Internet Engineering Task Force (IETF) differentiated services model and the IEEE 802.1D user priorities used in 802.x LANs. Network resources (e.g., queuing and scheduling resources) can also be associated with a behavior class, as a means of enabling service differentiation among behavior classes.

The BCS capability applies both to VCCs and VPCs, and to point-to-point and point-to-multipoint connections. The behavior class of a UBR can also be different in each direction. The intention is that the behavior class of a UBR connection will be consistent at all queuing points in a given direction, and therefore it is recommended that BCS values be translated where necessary at boundaries between administrative domains, for example, to ensure consistent treatment end to end. The BCS parameter differs from the parameters associated with other ATM service categories in that the impact to the end-to-end service is intentionally not specified and is network specific. The use of BCS in differentiated UBR is also not only applicable to IETF Diffserv, or IEEE 802.1D user priority. Any type of differentiated service policy of UBR connections is possible. The BCS mapping for a connection in such cases can be determined without examining an IP or MAC header, but by looking at other things like port configuration or physical and logical interface identifiers.

Use of Differentiated UBR to Support Diffserv

Various mechanisms exist to establish interconnections of IP devices attached to a common ATM network, including classical IP and ARP over ATM, LAN emulation (LANE), MPOA, and manual configuration, that we cover in other parts of this book. The Diffserv architecture enables IP networks to support QoS differentiation between aggregated flows. As we described earlier in this chapter, the Diffserv model is based on traffic classification and conditioning (*e.g.*, policing and marking) at the network edge, and a set of per-hop behaviors (PHBs) at queuing points within the network. The differentiated services code point (DSCP) is used to select the PHB that a packet experiences at a given queuing point, where the DSCP is encoded in the DS field of the IP header.

Another thing to remember is that Diffserv and ATM differ with respect to the nature of their respective QoS objectives. ATM QoS objectives are always absolute, that is, QoS measures must be specifically defined in terms of loss and delay, and the assurances offered to a connection are independent of those offered to other connections. Diffserv can be used to provide services with absolute QoS objectives, and services with relative service objectives. Absolute QoS objectives when used with Diffserv are similar to the ATM service categories in that the QoS measures must be specifically defined. When relative service objectives are applied to Diffserv, the service level provided to one connection may be dependent on that provided to another.

To support Diffserv, multiple UBR virtual channel connections may be established between each pair of ATM-attached IP devices. A behavior class representative of the desired treatment within the ATM network is assigned to each UBR connection. Knowledge of this assignment is retained by the IP layer of the devices at the end points of the connection. A mapping table then translates the DSCPs to behavior classes and loss priority. In order to honor packet-ordering constraints, the translation should be from DSCP to PHB scheduling class (PSC), and then from PSC to BCS. A PSC is the set of PHBs that share the same ordering constraints, while each PHB is specified by the DSCP. The loss priority is communicated to the ATM adaptation layer via the *CPCS-loss priority* parameter of the

CPCS-UNITDATA.Invoke primitive [AF BCS1.0], and coded into the CLP bit of the headers of the corresponding cells by the segmentation and reassembly (SAR) sublayer.

Queuing points within the ATM network then treat a cell according to the loss priority indicated in the header and the behavior class of the UBR connection.

Use of Differentiated UBR to Support IEEE 802.x

As described in Chapter 9, IEEE 802.1D supports packet classification and service differentiation via eight user priority levels. User priorities are encoded in three bits. Some LANs convey the user priority as an intrinsic part of their MAC frame (*e.g.*, token ring and FDDI), while others encode the user priority in a tag header inserted into the MAC frame (*e.g.*, Ethernet 802.1 Q VLAN tag or an 802.1D user priority field).

MAC bridging over ATM is typically implemented by meshing several MAC layer bridges using ATM virtual connections. Each virtual connection is modeled as a logical interface. The outgoing interface through which to forward a given frame is determined from a lookup of the destination address in its MAC layer header in the forwarding tables associating the MAC header to a logical interface.

To implement 802.1D user priorities within this framework, multiple UBR virtual connections are provisioned between pairs of MAC layer bridges. Just like before, a BCS value representative of the desired treatment within the ATM network is assigned to each UBR connection, and also using a mapping table that translates user priority values to BCS values.

To summarize the procedures that would be used to forward a given frame: An ATM-attached MAC layer bridge first performs a destination address lookup. This will return a set of virtual connections. The user priority in the MAC layer header is used in conjunction with a user priority-to-BCS mapping table, to determine the BCS. Finally, the BCS is used to select a UBR connection from among the set returned by the address lookup.

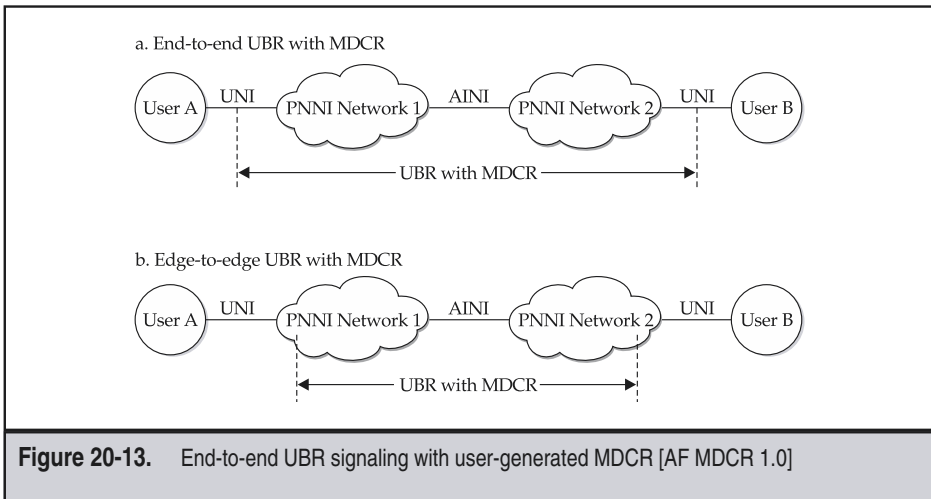
UBR Service Category with Optional MDCR Parameter

The MDCR parameter identifies an optional minimum bandwidth objective for UBR connections. One example application that may find this extension useful is interconnecting IP routers with VCCs or VPCs, where a minimum flow of availability, diagnostic, or other system data is desirable. This service allows the end user to indicate an objective for the lower bound on the bandwidth available for user data. The UBR with MDCR specification [AF UBR 1.0] does not specify in the normative text that the MDCR parameter is a commitment to support a minimum bandwidth for a virtual circuit, but it does include an appendix that strongly suggests that this is the way to implement the feature. This curious way to specify the MDCR feature is the result of discussions within the ATM Forum of the implications of providing minimum bandwidth guarantees to the UBR service category, changes that in effect would be changing the general understanding of this “best effort” service category. The net effect of the normative part of the specification is that UBR remains a “best effort” service, but it also provides a document where a “standard” implementation suggestion providing bandwidth guarantees to UBR is available. It is

hard to imagine that any real implementation of this specification would not follow the “suggested” option, since otherwise no real benefit is achieved. We will discuss this option, but keep in mind that the actual specification does not require bandwidth guarantees. UNI signaling of the MDCR indicates only a bandwidth request, and if you want to make this UBR enhancement useful, the PNNI routing protocol needs to advertise the MDCR impact on available bandwidth, as we discuss next.

If the user transmits at a rate less than or equal to MDCR, a low-cell-loss ratio can be expected with this service. If the user sends cells at a rate in excess of MDCR, the excess traffic will be delivered only within the limits of the additional available resources. Note that the MDCR bandwidth commitment does not have to be end-to-end on any connection, and two scenarios are shown in Figure 20-13. In Figure 20-13a, the MDCR is committed end-to-end; while in Figure 20-13b, the MDCR does not extend across the UNI.

To increase the usefulness of connection establishment, it is preferable that connections requesting a given MDCR be routed through nodes and links that support the MDCR feature and that also have sufficient capacity to provide the requested cell rate commitment. Nodes should advertise the effective available capacity for MDCR commitments using the AvCR in the PNNI UBR RAIG (see Chapter 15 on PNNI RAIGs). AvCR is a measure of effective available link resource capacity for CBR, real-time VBR, and non-real-time VBR service categories. For the ABR and GFR service categories, the AvCR is a measure of link resource capacity available for Minimum Cell Rate (MCR) reservation. The MDCR should be accounted for by decrementing the Available Cell Rate (AvCR) when accepting new UBR connections with a nonzero MDCR, rather than by increasing the Best-Effort Cell Rate (BeCR). In general, MDCR should be accounted for either by decrementing the AvCR in the UBR RAIG, or by accounting for it in the BeCR, but



not both, as indicated in Figure 20-14a. The real useful option is to decrement the MDCR from the reserved bandwidth, as shown in Figure 20-14b, which in effect provides hard bandwidth guarantees to the UBR connection, similar to the ABR and GFR reservation of MCR. The difference between these options is clearly shown in Figure 20-14. As we indicated at the end of the GFR details, with UBR with MDCR implemented as indicated in Figure 20-14b, you have to ask yourself why you should spend much time implementing GFR at all. Looking at the one incremental benefit associated with GFR not as efficiently addressed in ABR or UBR, which is a more efficient frame discard solution throughout the network, at a cost of the increased complexity needed to make GFR really deliver an optimal solution, it starts to look less and less attractive. The ubiquity of UBR adapters that practically can deliver almost the same exact functionality as GFR, with only minor software changes at the user and inside the network, makes this a very attractive solution.

REVIEW

This chapter introduced several key concepts and some basic terminology used in the remainder of this book. A traffic contract is an agreement between a user and a network regarding the Quality of Service (QoS) that the network guarantees to a cell or packet flow that conforms to a set of traffic parameters defined by a leaky bucket or token bucket conformance rule. The principal QoS parameters for ATM are: Maximum Cell Transfer Delay (maxCTD), Cell Delay Variation (CDV), and Cell Loss Ratio (CLR). The traffic parameters define at least the Peak Cell Rate (PCR) but may optionally define a Sustainable

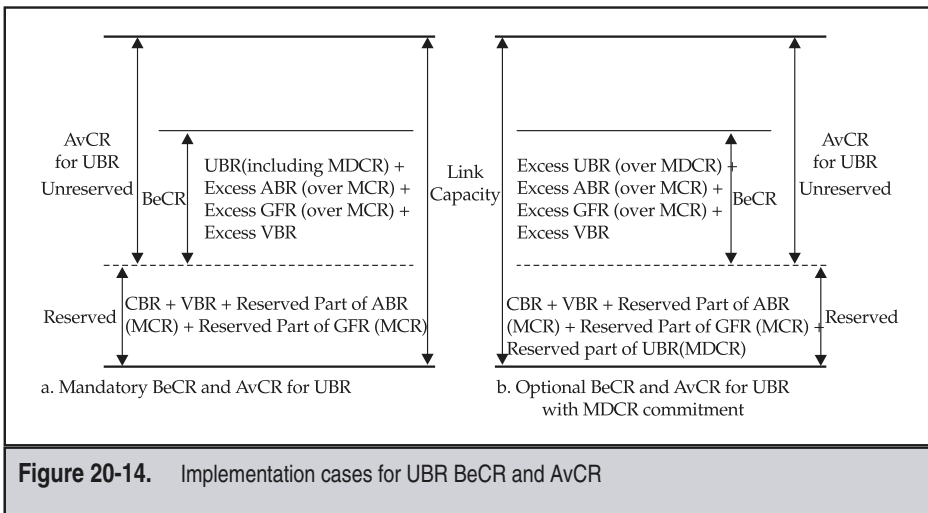


Figure 20-14. Implementation cases for UBR BeCR and AvCR

Cell Rate (SCR) and Maximum Burst Size (MBS) for specific cell flows identified by the Cell Loss Priority (CLP) bit in the ATM cell header. The conformance rule also associates a CDV Tolerance (CDVT) parameter with the peak rate, which the network specifies. IP defines QoS measurements for an end-to-end application based on connectivity, one-way delay, round-trip delay, one-way packet loss rate, and one-way delay variation. The text then defined the ATM Forum service categories and ITU-T transfer capabilities in terms of these QoS parameters and traffic parameters, and summarized the interoperable set of leaky bucket conformance-checking combinations defined by the ATM Forum. We also described the IP Diffserv Per-Hop Behaviors (PHBs), which play an analogous role in IP and MPLS networks. The chapter highlights the similarities between the ATM and IP/MPLS QoS and traffic management approaches by reviewing the interworking defined for Intserv by the IETF. Finally, we concluded the chapter by detailing recent ATM Forum enhancements in support of packet switching, namely, the GFR service category and recent enhancements to the UBR service category.

CHAPTER 21



Traffic Control, QoS Mechanisms, and Resource Management

This chapter introduces, compares, and contrasts some basic concepts from standards groups and the industry regarding traffic control. *Traffic control* provides the means for a user to ensure that offered load conforms to the rate specified in a traffic contract, as well as the means for networks to check conformance against this same rate. This fundamental balance allows a network to deliver the negotiated QoS performance across all users at the agreed-to traffic load. Checking conformance involves policing. ATM employs the leaky bucket algorithm, windowing techniques, and the Generic Cell Rate Algorithm (GCRA), while IP and MPLS define the policing functions in terms of a token bucket. We then describe how the user can meet these stringent network requirements using traffic shaping with either a leaky bucket for ATM or a token bucket for IP and MPLS. The chapter then describes the methods that ATM switches or IP/MPLS routers can use to deliver the required QoS. This includes the concepts of queuing prioritization and scheduling, as well as loss thresholds. The chapter concludes with a discussion of techniques used to allocate and manage resources such that overloads do not occur, namely, admission control and ATM VP or MPLS LSP path-based traffic aggregation.

ACHIEVING CONFORMANCE

This section provides an overview of traffic control as an introduction to the topics covered in this chapter. A generic ATM-centric reference model from I.371 as shown in Figure 21-1 illustrates the placement of various traffic and congestion control functions. Starting from the left side, user terminal equipment and private network switches may shape cell flows to conform to traffic parameters. The ingress port of the network then checks conformance of this cell flow to traffic parameters with a Usage Parameter Control (UPC) function at the public User-Network Interface (UNI). In a similar manner, networks may check the arriving cell flows from a prior network using Network Parameter Control (NPC) at a public Network-Node Interface (NNI). Networks may employ additional traffic control functions such as Connection Admission Control (CAC), priority control, resource management, ABR flow control, and traffic shaping, as indicated in the figure.

In traffic and congestion control, the time scale over which a particular control is applicable is important [Hui 88, Awater 91, Hong 91]. Figure 21-2 illustrates the time scales of various traffic and congestion control methods. The minimum time scale that traffic or congestion control acts on is a cell time (which is approximately 10 μ s on a DS3, and 3 μ s on an OC3/STS-3c). For example, UPC traffic control (commonly called policing) and traffic shaping act on this time scale. Also, selective cell discard and priority queue servicing, as described in the next chapter, act at the cell time level. Other functions operate at the packet or frame time scale, such as frame-level policing or selective frame discard. The next time scale that makes sense is that of round-trip propagation time, as covered in the section on closed-loop flow control in Chapter 22, which includes the ATM Available Bit Rate (ABR) service and protocol. The next major event that changes over

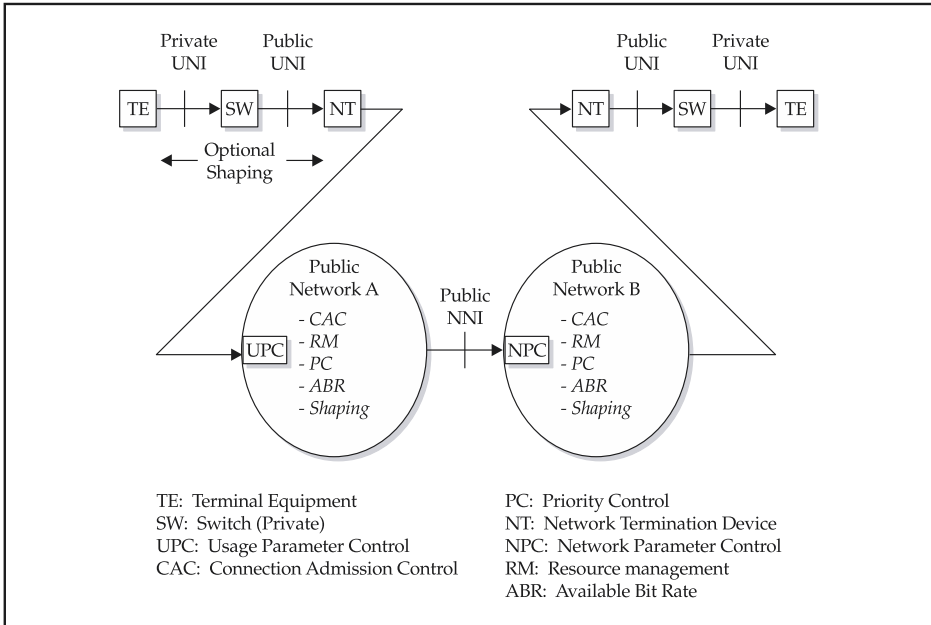


Figure 21-1. Overview of traffic and congestion control functions

Example Traffic Control and Congestion Control Functions	Time Scale
Cell-level policing, shaping, discard, scheduling and queuing	Cell Time
Frame-level policing, shaping, discard, scheduling and queuing	Frame/Packet Time
Closed-loop flow control, Available Bit Rate (ABR)	Round-trip Propagation Time
Routing, Admission Control, Resource Allocation	Call/Connection Inter-Arrival Time
Centralized Network Management Controls	Switch or Circuit Provisioning Interval
Long Term Network Engineering	

Figure 21-2. Time scales of traffic and congestion control functions

time is the arrival of either connection provisioning or signaling requests to either establish or relinquish connections. Typically, this time scale is much greater than the round-trip delay. Finally, long-term network engineering and network management procedures operate on much longer time scales, on the order of hours to days, and extending to months and years for long-range network planning.

The IETF Diffserv working group defined similar concepts in RFC 2475 for the functions performed at a node, as shown in Figure 21-3. Starting from the left-hand side, the classifier determines what traffic function should be applied to arriving packets. The classifier may use fields from the MPLS header, the IP packet header, the transport protocol header, or even the application layer. Packets may then be subject to metering (e.g., policing or counting) or marking, as shown in the center of the figure. Prior to departure, packets may be subject to shaping or may be dropped, depending upon the results of the metering and marking operation.

CHECKING CONFORMANCE: POLICING

Recall the old adage about how some people view a glass as being half full, while others view it as half empty. An analogous situation exists between the methods defined for ATM and IP/MPLS when measuring conformance to traffic parameters. Although these points of view differ, many of the basic concepts are the same, and what differs is the terminology and the specifics of the algorithms involved.

In IP and MPLS, a bucket collects tokens that measure the average rate and burst duration. A device periodically adds tokens to the bucket(s) corresponding to each flow at a rate specified by the traffic parameters. If an arriving packet finds sufficient tokens in the bucket, it is considered conformant to the traffic parameters; otherwise, it is not. In ATM, arriving cells fill a bucket, which leaks at a rate as specified by the traffic parameters. If an arriving cell would overflow any of the buckets, then the network considers the cell noncompliant to the traffic parameters. Let's examine each of these methods in more detail via some examples, and then map their measurement methods and conformance determination back to the generic traffic parameters of peak rate, average rate, and burst duration.

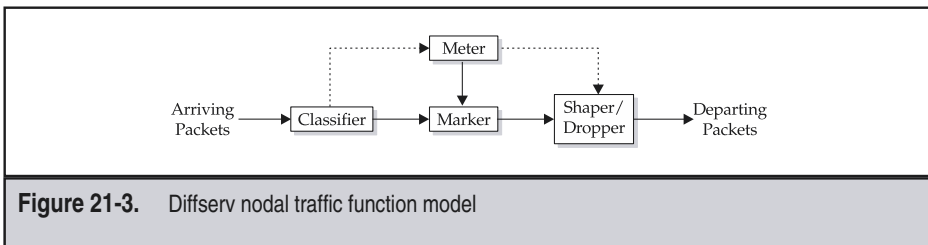


Figure 21-3. Diffserv nodal traffic function model

ATM Policing

ATM networks employ Usage Parameter Control (UPC) and Network Parameter Control (NPC) to check conformance of cell flows from a user or another network, respectively, against negotiated traffic parameters. Another commonly used name for UPC/NPC is *policing* [Rathgeb 91]. This is a good analogy because UPC and NPC perform a role similar to the police in society. Police enforce the law, ensuring fair treatment for all people. The ATM UPC/NPC algorithms enforce traffic contracts while Connection Admission Control (CAC) fairly allocates bandwidth and buffering resources among the users. Without UPC/NPC and CAC, unfair situations where a single user “hogs” resources can occur. Most of the functions and discussion in the following sections apply equally to UPC and NPC, with any differences identified explicitly.

Standards do not specify the precise implementation of UPC and NPC functions; instead, they bound the performance of any UPC/NPC implementation in relation to a Generic Cell Rate Algorithm (GCRA), which is essentially a fancy name for the leaky bucket algorithm. Indeed, the compliant connection definition part of the traffic contract identifies how much nonconforming traffic a network will still provide a QoS guarantee. The other requirement is that the UPC should not take a policing action (i.e., tag or discard) on more than the fraction of cells that are nonconforming according to the leaky bucket rule; or in other words, the UPC cannot be too tight and over-police user cell flows.

Also note that the UPC may police different cells than a leaky bucket algorithm does due to inevitable differences in initialization, the latitude defined for a compliant connection, or the fact that the UPC implementation of a particular device is not the leaky bucket algorithm.

This section begins with a few examples of policing that illustrate the operation of the leaky bucket and how it relates to conformance definitions. We then give three examples of UPC/NPC implementations; one using the leaky bucket algorithm, and two windowing schemes to illustrate differences in how cell flows are compliance checked by different algorithms using the same traffic parameters.

Examples of Leaky Bucket Policing

The leaky bucket algorithm is key to defining the meaning of conformance in ATM. The leaky bucket analogy refers to a bucket with a hole in the bottom that causes it to “leak” at a certain rate, corresponding to a traffic cell rate parameter (e.g., PCR or SCR). The “depth” of the bucket corresponds to a tolerance parameter (e.g., CDVT or BT). A subsequent section details these tolerance and traffic parameters. Each cell arrival creates a “cup” of fluid flow “poured” into one or more buckets for use in conformance checking. The Cell Loss Priority (CLP) bit in the ATM cell header determines into which bucket(s) the cell arrival fluid pours.

In the leaky bucket analogy for policing, the cells do not actually flow through the bucket; only the check for conformance to the contract does. On the other hand, as described later, one implementation of traffic shaping does actually have cells flow through the bucket. The operation of the leaky bucket is described with reference to the following

figures for examples of conforming and nonconforming cell flows. In all of the following examples, the nominal interval between cell arrivals is four cell times, which is the bucket increment, and the bucket depth is six units. A cell time is the amount of time required to transmit a cell at the physical line rate. Many of our examples employ the notion commonly employed in queuing theory of a fictional “gremlin” performing some action or making an observation. Real devices don’t have gremlins (even though certain users disgruntled with early ATM devices may disagree here), but this treatment helps give an intuitive insight into the operation of ATM traffic controls.

For each cell arrival, the gremlin checks to see if adding the increment for a cell to the current bucket contents would create overflow. If the bucket would not overflow, then the cell is *conforming*; otherwise, it is *nonconforming*. The gremlin pours the fluid for nonconforming cells on the floor. Fluid from a cell arrival is added to the bucket only if the cell is conforming; otherwise, accumulated fluid from nonconforming cells might cause later cells to be identified as nonconforming. The hole in the bucket drains one increment each cell time. Each cell arrival adds a number of units specified by the increment parameter. The fluid added by a cell arrival completely drains out after a number of cell times given by the leaky bucket increment. We now look at detailed examples of cell arrivals containing conforming and nonconforming cells.

For the conforming cell flow example shown in Figure 21-4, the first cell arrival finds an empty bucket, and hence conforms to the traffic contract. Thus, the first cell arrival fills the bucket with four units, the bucket increment in our example. The gremlin indicates that this cell is conforming, or okay. At the third cell time, two units have drained from the bucket, and another cell arrives. The gremlin determines that the fluid from this cell would fill the bucket to the brim (i.e., to a depth of six); therefore, it also conforms to the traffic contract so that the gremlin adds its increment to the bucket, filling it completely at cell time 3. Now the next earliest conforming cell arrival time would be four cell times later (i.e., cell time 7), since four increments must be drained from the bucket in order for a

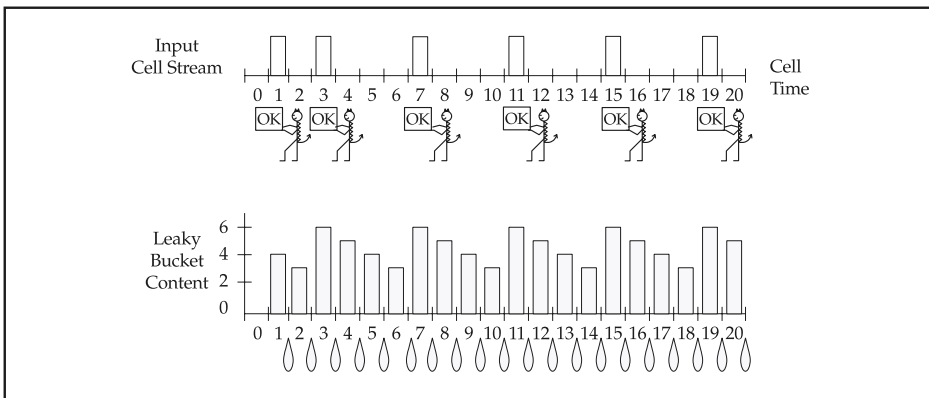


Figure 21-4. A conforming cell flow

cell arrival not to cause the bucket of depth equal to six units to overflow. This worst-case arrival of conforming cells continues in cell times 15 and 19 in the example.

In the example in Figure 21-5, which contains nonconforming cells, the first cell arrival at cell time 1 finds an empty bucket, is therefore conforming, and fills the bucket to a depth of four units. Over the next four cell times, the bucket drains completely—one unit per cell time. At the fifth cell time, another cell arrives and fills the empty bucket with four units of fluid. At the sixth cell time, a cell arrives, and the gremlin determines that adding the arriving cell's fluid would overflow the bucket. Therefore, this cell is nonconforming and the gremlin pours the fluid for this cell onto the floor, bypassing the bucket. Since the gremlin did not pour the fluid for the nonconforming cell into the bucket, the next conforming cell arrives at cell time 7, completely filling the bucket to the level of six units. The next cell arrives at cell time 13 and fills the empty bucket with four units. The next cell arrival at cell time 15 fills the bucket to the brim. Finally, the cell arrival at cell time 17 would cause the bucket to overflow; hence, it is by definition nonconforming, and the gremlin pours the nonconforming cell's fluid on the floor, bypassing the bucket again.

The leaky bucket example of Figure 21-6 uses the same parameters as in the previous examples. Now, three more gremlins, "Tag," "Discard," and "Monitor," join the "Dump" gremlin in this example to illustrate the range of UPC actions.

Cell arrivals occur along the horizontal axis at the top of Figure 21-6, with the "Dump" gremlin pouring the fluid from nonconforming cells (indicated by cross-hatching in the figure) past the leaky bucket (the other gremlins, "Tag," "Discard," and "Monitor," all operate in conjunction with "Dump's" identification of a nonconforming cell. "Tag" sets the

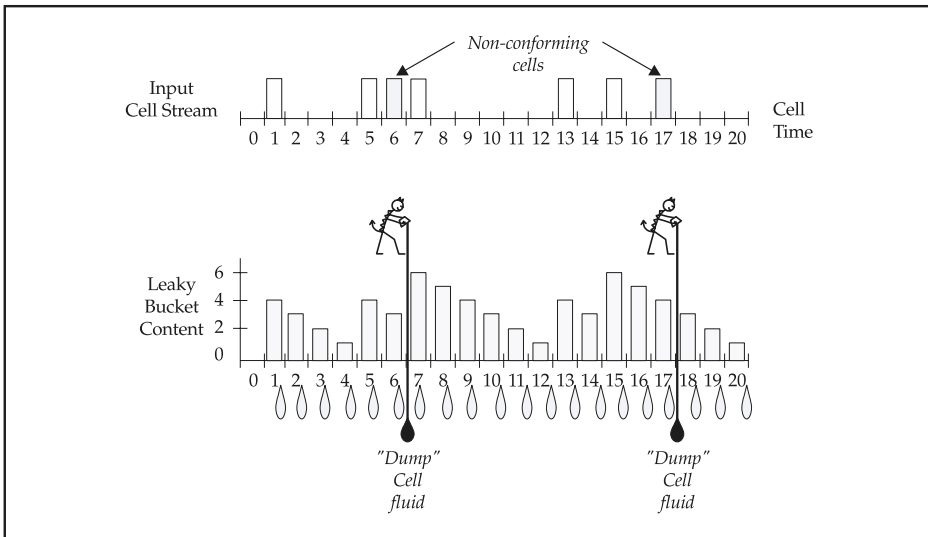


Figure 21-5. Nonconforming cell flow

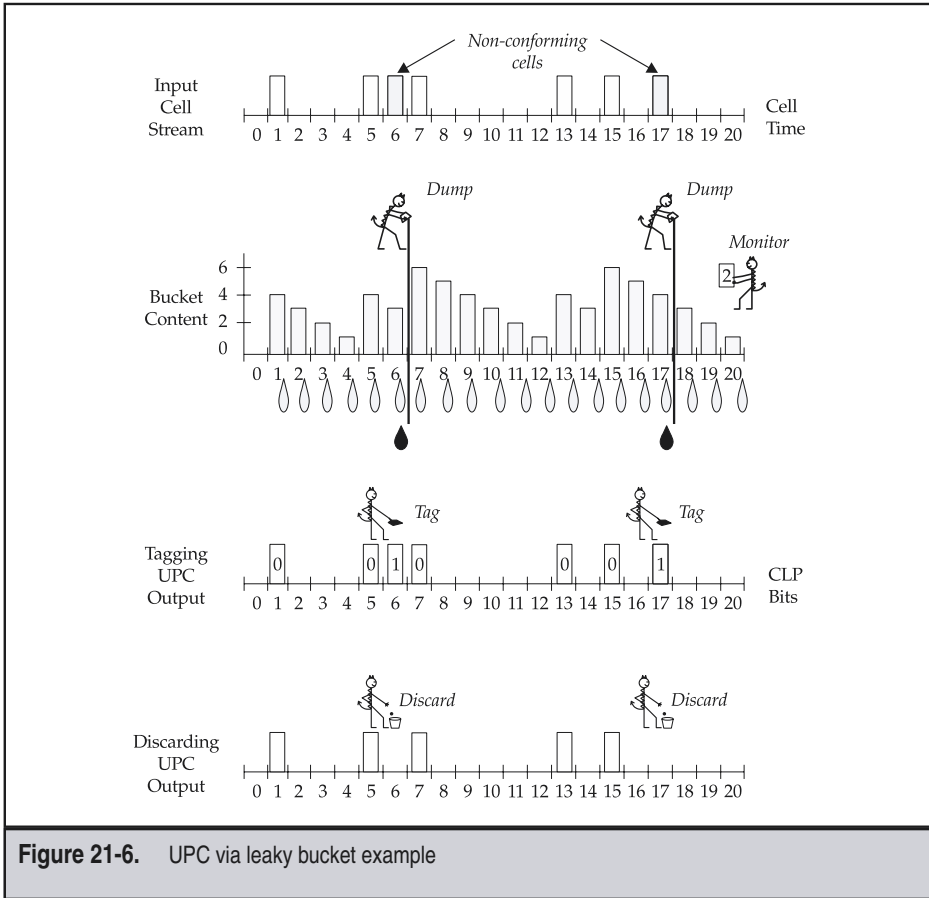


Figure 21-6. UPC via leaky bucket example

CLP bit to 1 (regardless of its input value) and lets the cell pass through, with its position unchanged. “Discard” simply blocks the cell from transmitting. “Monitor” simply keeps track of how many cells were nonconforming on his notepad. There is a fourth possible UPC action in the standard—namely, do nothing, which corresponds to all of the gremlins being out to lunch in this example.

Sliding and Jumping Window Policing

This section compares two windowing UPC mechanisms to the leaky bucket mechanism to illustrate how different UPC implementations, designed to police the same proportion of nonconforming cells, yield different results. Some older ATM implementations used

these techniques; however, most modern devices use the leaky bucket for reasons demonstrated in this section. Observe from the previous examples that the worst-case conforming cell flow could have at most three cells in ten cell times. For any set of single leaky bucket parameters, an apparently equivalent relationship is at most M cells in a window of N cell times. In the following examples, $M = 3$ and $N = 10$. Two types of windowing UPCs are considered: a sliding window method and a jumping window method. We use two more gremlins to describe these UPCs: "Slide" and "Jump."

For reference, the top of Figure 21-7 shows the same arrival sequence containing nonconforming cells as input to the leaky bucket in the previous example. In the sliding window protocol, the gremlin "Slide" moves a window covering N cell times to the right each cell time, as shown by the arrows below the first axis. If no more than M cells exist in the window covering N cell times, then no UPC action occurs. However, if by sliding the window one unit to the right, more than M cells exist within the window, then UPC takes action on the cells that violate the " M out of N " rule, as shown along the axis in the middle

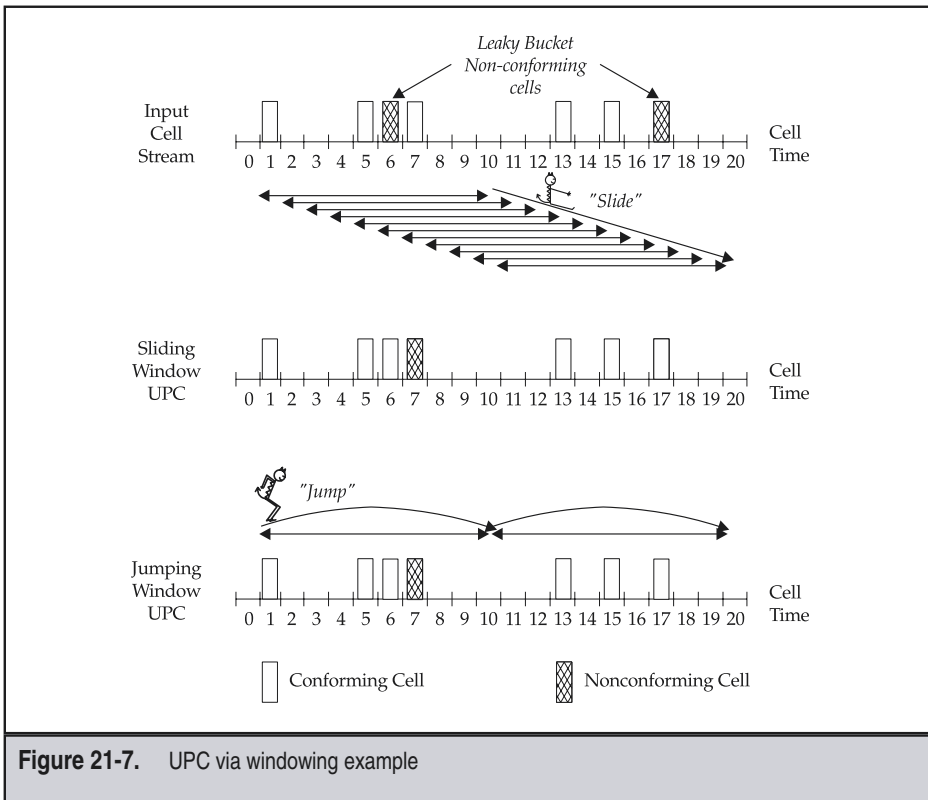


Figure 21-7. UPC via windowing example

of the figure. As shown there, the cell arrival in time slot 7 is considered nonconforming. The sliding window algorithm does not include cells previously classified as nonconforming in subsequent computations. The cell in time slot 7 is considered nonconforming. The sliding window and leaky bucket UPCs act upon the same number of cells; however, they do not act on the same cells! In the jumping window scheme, "Jump" moves the window N units to the right every N cell times as shown at the bottom of the figure. "Jump" applies the same M out of N count rule but detects only one nonconforming cell as compared with two in the leaky bucket UPC example.

In the example of Figure 21-7, the jumping and sliding window UPCs were looser than the leaky bucket UPC methods. The fact that different UPC algorithms, or even the same algorithm, may police different cells is called *measurement skew*. This is the reason that standards state UPC performance in terms of the fraction of conforming cells according to a specific reference algorithm. However, in some cases the sliding and jumping window UPC algorithms may indicate lack of conformance for a smaller or larger proportion of cells than the leaky bucket algorithm.

Figure 21-8 illustrates a pathological case where the difference between leaky bucket, sliding window, and jumping window UPCs is even more pronounced. Each algorithm has parameters chosen to admit one cell every three cell times on the average and to allow at most two back-to-back cells. The leaky bucket UPC has an increment of three cells and a bucket depth of five cells, while the sliding and jumping window algorithms have parameters $M = 2$ and $N = 6$. Figure 21-8 illustrates the arrival of 10 cells. The leaky bucket

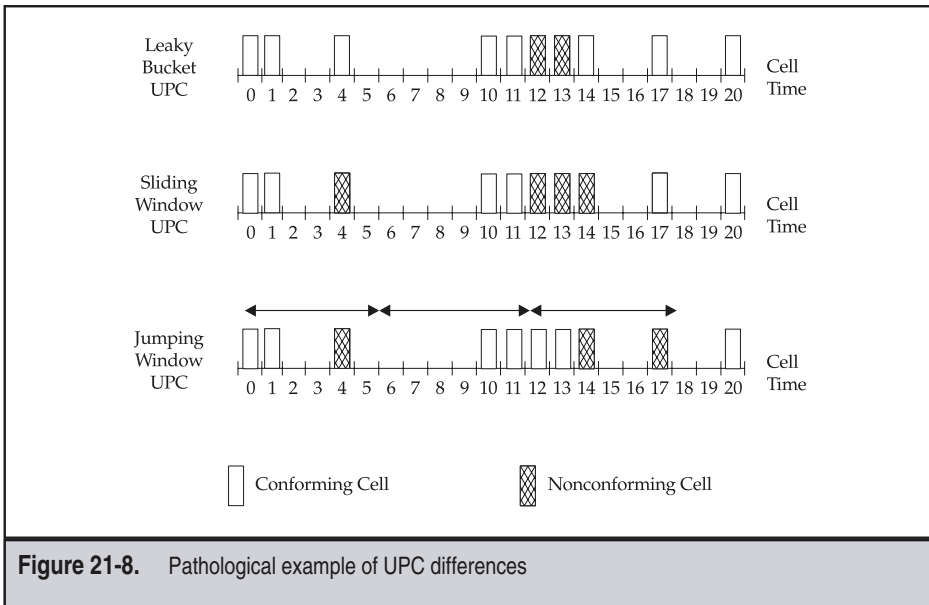


Figure 21-8. Pathological example of UPC differences

UPC identifies 20 percent of the cells as nonconforming, the sliding window identifies 40 percent as nonconforming, and the jumping window identifies 30 percent as being nonconforming.

Generic Cell Rate Algorithm (GCRA) and Virtual Scheduling

As seen from the previous example, because different UPC implementations can result in markedly different proportions of policed cells, the ATM Forum specified a formal algorithm as a reference model in the UNI 3.1 specification in 1994 [AF UNI 3.1] (now part of the TM 4.1 specification [AF TM 4.1]). Figure 21-9 illustrates the two equivalent interpretations of the ATM Forum and the ITU-T Recommendation I.371 Generic Cell Rate Algorithm (GCRA): the virtual scheduling algorithm and the leaky bucket algorithm. Each of these algorithms utilizes the two traffic parameters that define either the peak rate or the sustainable rate and the associated tolerance parameters: an Increment (I) and a Limit (L), both expressed in units of time. As stated in the standards, these representations are equivalent. The virtual scheduling representation appeals to time sequence-oriented people, while the leaky bucket method appeals to the mathematical and accounting types among us.

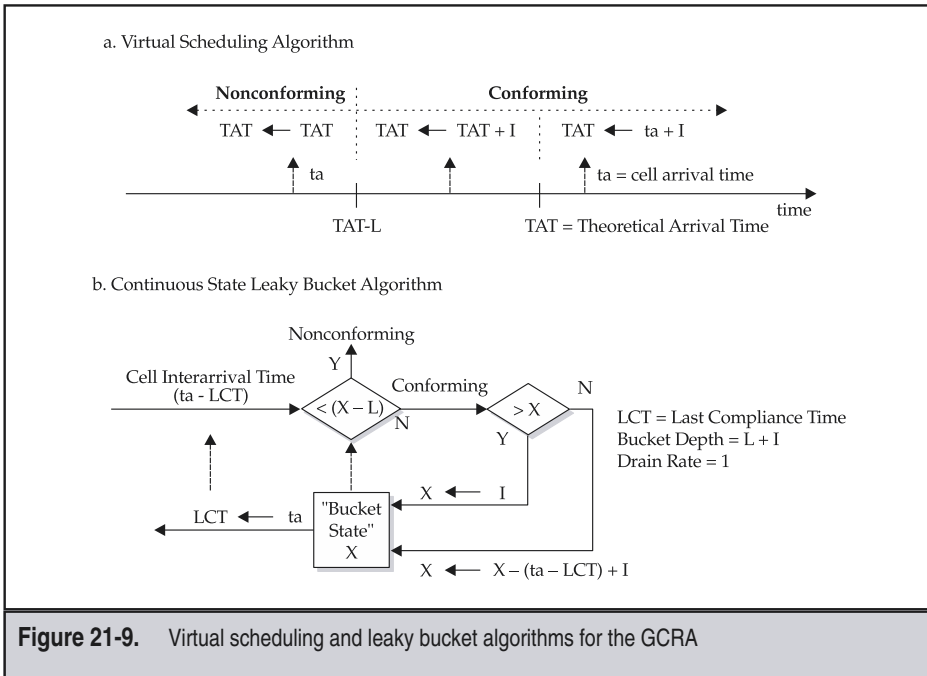


Figure 21-9. Virtual scheduling and leaky bucket algorithms for the GCRA

Figure 21-9a illustrates the virtual scheduling algorithm that utilizes the concept of a *theoretical arrival time (TAT)* for the next conforming cell. If a cell arrives more than the tolerance limit, L , earlier than the TAT, then it is nonconforming as shown in the figure. Cells arriving within the tolerance limit are conforming and update the TAT for the next cell by the increment, I , as indicated in the figure. Cells arriving after the TAT make the tolerance available to subsequent cells, while cells arriving within the tolerance interval reduce the tolerance available to subsequent cells.

The flowchart in Figure 21-9b shows the detailed workings of the leaky bucket algorithm. Here, the GCRA parameters have the interpretation that the bucket depth is $L + I$ and the bucket drain rate is one unit per cell time. The leaky bucket algorithm uses the concepts of a Last Compliance Time (LCT) variable to hold the arrival time of the last conforming cell, as well as a bucket fill state variable, X . The algorithm compares the amount that drained from the leaky bucket since the time of the last compliant cell arrival (i.e., $ta - LCT$) to the amount that had to drain from the bucket in order for the next cell to be considered conforming (i.e., $X - L$) to determine whether the cell arrival would overflow the bucket. If overflow would occur, then the cell is deemed nonconforming. Otherwise, the flowchart checks to see if the bucket completely drained since the LCT before updating the bucket state variable X accordingly (i.e., X is never negative). The algorithm then substitutes for the current values of the LCT and bucket state X in preparation for processing the next cell arrival.

The ATM Forum TM 4.1 specification uses the GCRA(I, L) rule to formally define the conformance checks for the peak and sustainable rate conformance checks as follows:

Peak rate: GCRA(T, τ), where $PCR = 1/T$ and $CDVT = \tau$

Sustainable rate: GCAR(T_S, τ_S), where $SCR = 1/T_S$ and $BT = \tau_S$

The reader interested in more details on the formal GCRA algorithm should download the ATM Forum's TM 4.1 specification or consult normative Annex A of ITU-T Recommendation I.371.

IP and MPLS Policing

This section begins with some simple examples of token bucket policing given in a similar style to that used for the leaky bucket in order to highlight the similarities in concept, yet differences in implementation. We then give the details of the token bucket algorithm itself.

Token Bucket Example

The token bucket algorithm operates in a manner opposite to that of the leaky bucket, yet produces a similar result. A token bucket holds credits that are supplied at a rate r , up to the bucket depth b . Credits in excess of the bucket depth are not held. An arriving packet must find sufficient credits in the token bucket to be considered conforming. At the time of this writing, several interpretations and implementations of the token bucket algorithm were under consideration in the IETF [ID DSIM]. We describe the interpretation

here of a “strict” token bucket where credits are added at discrete points in time, since it is one of the simpler interpretations and maps most directly to the leaky bucket examples described previously.

Let’s look at a simple example with reference to Figure 21-10 to illustrate the concept of how a token bucket checks compliance of an arriving sequence of packets. In order to simplify the presentation, we assume that time is slotted in fixed units (for example, according to a minimum policed unit) and that all packets are multiples of this time slot unit. The top of the figure shows a series of packet arrivals, and the bottom of the figure shows the contents of a token bucket used to determine whether an arriving packet conforms to a traffic specification. In this example, the token bucket has a depth b of six units and tokens are added at a rate of one credit once every four time slots. Starting at time slot 0, we assume that the token bucket is full of $b = 6$ credits. A packet arrives during time slots 1 and 2, as shown at the top of the figure. At the end of time slot 2, there are sufficient credits in the token bucket, and the gremlin declares that the arriving packet is conforming (i.e., okay), as shown in the center of the figure, and two credits are deducted from the token bucket. In time slot 3, one credit is added to the token bucket, as shown by the text “+1” above the token bucket content bar. Four time slots later (i.e., time slot 7), another credit is added. Similarly, at time slot 11, it is time to add another credit; but the token bucket is already full (i.e., it contains six units), so no credit is added.

Continuing the example, the next packet begins arriving at time slot 10 and continues for four time units. Upon completion of this second packet arrival at time slot 13, there are six credits in the token bucket, and the gremlin finds it to be conforming and deducts four credits (i.e., the length of the packet) from the token bucket. The third packet begins to arrive immediately afterward, beginning in time slot 14. During this arrival, another credit is added to the token bucket at time slot 15. Upon completion of the third packet arrival at time slot 16, the gremlin finds that the token bucket contains exactly the

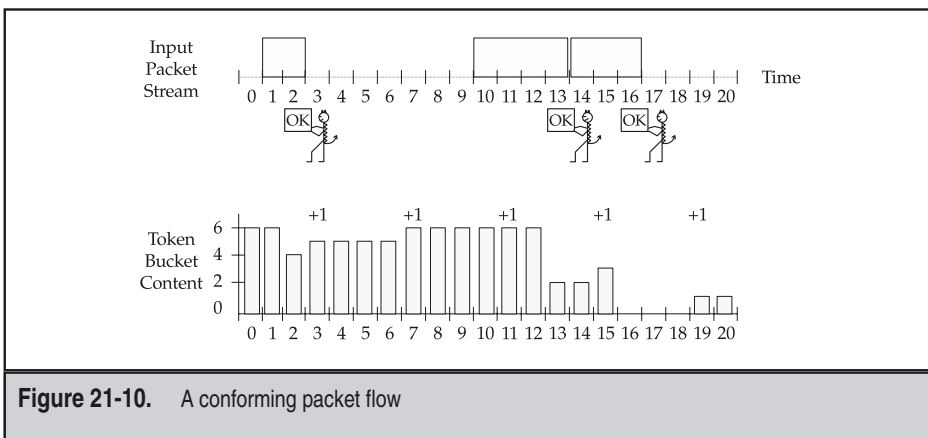


Figure 21-10. A conforming packet flow

number of credits corresponding to the packet length, indicates that the packet is conforming, and empties the token bucket. Finally, at time slot 19, another credit is added to the token bucket. Note that any subsequent packets of length greater than one time unit would be considered nonconforming until more credits accumulate in the token bucket.

The next example illustrates the operation of the token bucket when handling packets that are nonconforming with reference to Figure 21-11. This figure uses the same convention of packet arrivals shown on the line above the token bucket contents with the gremlins in the middle indicating conformance on a packet-by-packet basis. The example begins in time slot 0 with a token bucket full of $b = 6$ credits, where a credit is added once every four time slots, as before. The first packet arrives in time slot 1 and continues for two time slots. In time slot 2, the gremlin determines that the packet is conforming (i.e., okay) and deducts two credits from the token bucket. In time slot 3, a credit is added to the token bucket. Beginning in time slot 3, a packet of length equal to four time units arrives. At time slot 6, the gremlin determines that the packet is conforming and deducts four credits from the token bucket. In time slot 7, a credit is added to the token bucket. A third packet of length equal to three time units begins arriving in time slot 8 and completes in time slot 10. At this point, there are only two credits in the token bucket, and the gremlin determines that this packet is nonconforming (i.e., not okay). Note that no credits are deducted from the token bucket for the third packet.

Immediately afterward, a fourth packet of length equal to three time units begins arriving at time slot 11 and completes at time slot 13. At this point, there are now three credits in the token bucket since a credit was added at time slot 11, and the gremlin identifies this packet as conforming, deducting all three credits from the token bucket and leaving it empty. In time slot 15, a credit is added to the token bucket. At time slot 17, another

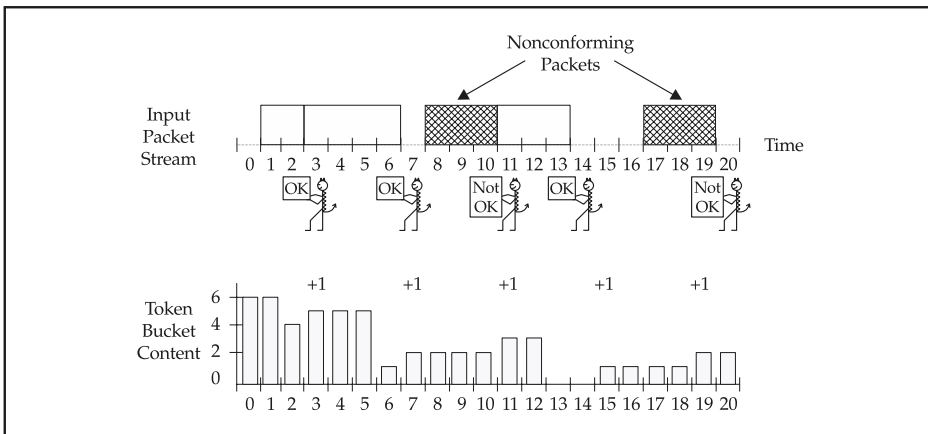


Figure 21-11. A nonconforming packet flow

packet of length equal to three time units begins arriving. At time slot 19, this packet arrival completes, and the gremlin determines that it is nonconforming because only two credits are present in the token bucket. As before, no credits are deducted from the token bucket for a nonconforming packet.

Token Bucket Algorithm

The Internet's resource reservation protocol (RSVP) [RFC 2211, RFC 2212, RFC 2205] uses the token bucket algorithm to describe traffic parameters corresponding to a specific flow of IP packets. Two parameters completely specify the token bucket: an average rate r and a bucket depth b . Figure 21-12 depicts the basic operation of the token bucket algorithm [Partridge 94, Stallings 98]. Conceptually, a device measures the conformance of a sequence of packet arrivals using the token bucket that contains up to b bytes worth of tokens. The device adds tokens to the bucket at a rate of r bytes per second, as shown in the figure. An arriving packet conforms to the token bucket traffic specification if the level of the tokens in the bucket equals or exceeds the packet length. Specifically, when a packet arrives, the device checks the current level of tokens in the bucket X against the length L of the arriving packet. If $L \leq X$, then the packet conforms to the token bucket traffic specification; otherwise, the packet is considered nonconforming. Normally, conforming packets remove the number of tokens (e.g., bytes) equal to their length. Nonconforming packets do not remove any tokens. Typically, a network guarantees QoS for only conforming packets because resources are allocated according to the traffic parameters.

The basic effect of the token bucket parameters r and b is that the amount of data sent $D(T)$ over any interval of time T obeys the rule:

$$D(T) \leq rT + b$$

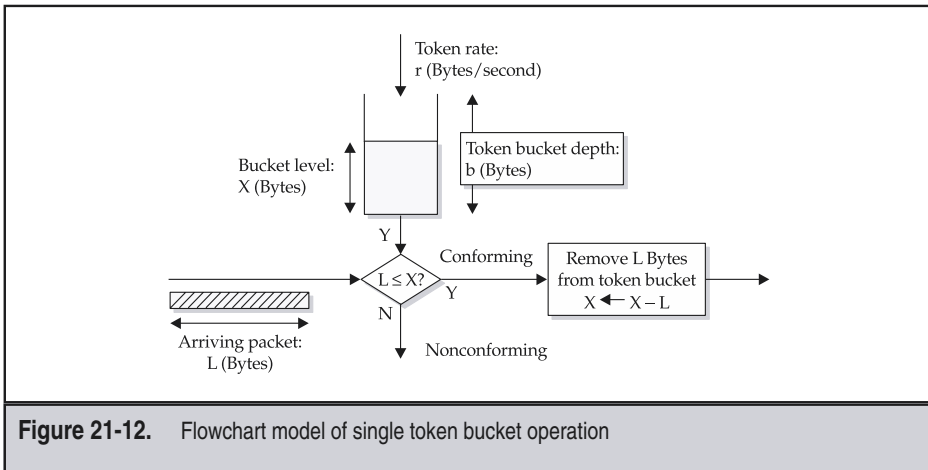


Figure 21-12. Flowchart model of single token bucket operation

Note that this rule means that the actual average rate $A(T)$ over a time interval T is actually somewhat greater than r , namely,

$$A(T) = D(T)/T = r + b/T > r$$

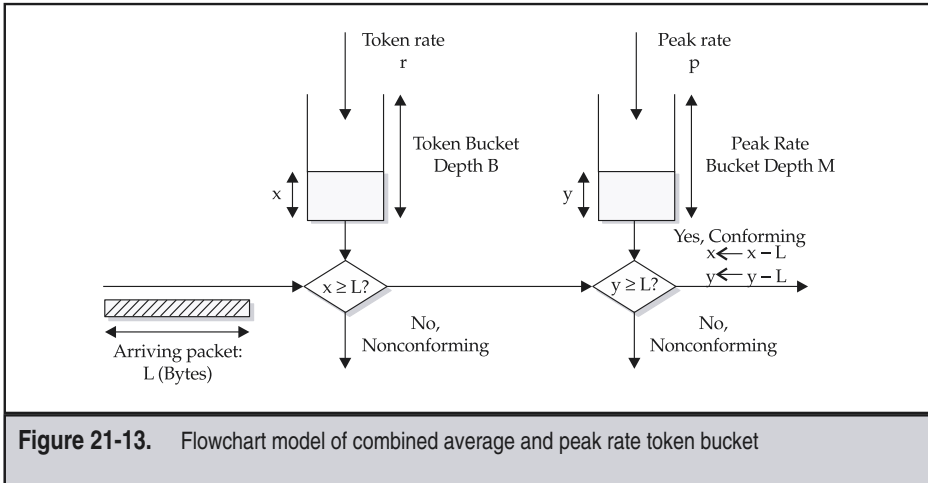
The full RSVP traffic specification starts with the token bucket specification and adds three additional parameters: a minimum-policed unit m , a maximum packet size M , and a peak rate p . The measure for the peak traffic rate p is also bytes of IP datagrams per second, with the same range and encoding as the token bucket parameter r . When the peak rate equals the link rate, a node may immediately forward packets that conform to the token bucket parameters. For peak rate values less than the link rate, a peak rate token bucket operating in parallel with the average rate token bucket implements conformance checking that ensures that the transmitted data $D(T)$ over any interval of time T satisfies the following inequality:

$$D(T) \leq \text{Min}[pT+M, rT+b]$$

Figure 21-13 depicts the block diagram model for an average rate token bucket with parameters r and b operating in conjunction with a peak rate token bucket with parameters p and M . Although the flowchart shows a serial operation, the token bucket tests for the (average) token rate, and the peak rate can be performed in parallel. In a manner similar to ATM, if either the average or peak rate check fails, the arriving packet is considered nonconforming. Furthermore, only if both checks succeed are tokens removed from the bucket, as shown in the right-hand side of the figure.

ENSURING CONFORMANCE: SHAPING

For a user to derive maximum benefit from guaranteed QoS, then the device connecting to the network should ensure that the cells or packets sent to the network conform to the parameters in the traffic contract. Standards call the method to achieve this goal *traffic shaping*. In other words, the user equipment processes the source stream such that the resultant output toward the network conforms to the traffic parameters according to the applicable conformance algorithm (i.e., leaky bucket or token bucket). Although the standards make traffic shaping optional, recall that the network definition of a compliant connection need not guarantee QoS performance for nonconforming cells or packets. Therefore, a user wanting guaranteed QoS must shape traffic to ensure conformance to the traffic parameters in the contract. A network may employ shaping when transferring a packet flow to another network in order to meet the conditions of a network-to-network traffic contract, or in order to ensure that the receiving user application operates in an acceptable way.



Overview of Possible Shaping Methods

Various papers in the literature as well as the standards propose the following traffic shaping implementations:

- ▼ Peak rate reduction
- Burst length limiting
- Source rate limitation
- Shaping: buffering with leaky bucket and token bucket
- ▲ Scheduling

We give a brief summary for each of the proposals listed and cover some of them in detail in the remainder of this section. Peak rate reduction involves operating the sending terminal at a peak rate *less* than that in the traffic contract, reducing the possibility of conformance violation. Burst length limiting constrains the transmitted burst length to a value *less* than the maximum burst size in the traffic contract. Source rate limitation is an implicit form of shaping that occurs when the actual source rate is limited in some other way; for example, in DS1 circuit emulation, the source rate is inherently limited by the TDM clock rate. Buffering operates in conjunction with algorithms like a leaky bucket or token bucket, to ensure that packets do not violate the traffic parameters of the contract by buffering packets until the algorithm will admit them. Scheduling cov-

ers the order and frequency in which a switch or router services packets that are in queue. Other proposals for a UPC/NPC traffic control function include those of spacing and framing. In a spacing implementation, the resultant output never violates the nominal intercell interval, but may discard additional cells. Framing overlays a synchronous structure of frame boundaries on an asynchronous cell stream and is a method of controlling delay variation for delay-sensitive virtual connections. Neither spacing nor framing were ever widely implemented, and they are not further discussed here. Some additional detail on these methods and historical references are in [McDysan 98].

Leaky Bucket Buffering

This section gives an example of traffic shaping using buffering and a leaky bucket implementation to transform a nonconforming cell flow into a conforming cell flow. This example uses the same notation for cell arrivals over time along the horizontal axis, the same nominal interarrival time of four cell times, and the same leaky bucket depth of six as in the earlier ATM policing examples. We're pleased to introduce two new gremlins, "Stop" and "Go," to illustrate the buffering and scheduling operation.

The gremlin "Stop" replaces "Dump" in the earlier nonconforming example. "Stop" commands the ATM hardware genie to buffer the cell if its fluid flow would cause bucket overflow, and "Go" allows a cell transmission as soon as the bucket drains far enough to admit the latest cell. When "Stop" and "Go" are out of synch, then cells build up in the shaping buffer, as shown in Figure 21-14. The figure illustrates this operation with the individual cells labeled A through G, and the nonconforming cells from the previous examples indicated by shading. Cell arrivals A and B are conforming and leave the bucket in a state such that arrival C at cell time 6 is nonconforming, and hence "Stop" stores cell C in the shaping buffer. Cell D arrives immediately after C, so the gremlin "Stop" also buffers D. In the same cell time, the bucket empties enough that "Stop's" partner, "Go," transmits cell C and adds its flow to the bucket. At cell time 11, "Go" sends cell D and fills the bucket. At cell time 13, cell arrival E would cause the bucket to overflow; hence "Stop" buffers it. Cells F and G are similarly buffered by "Stop" and transmitted at the earliest conforming time by "Go," as illustrated in the figure. For the reader wishing to continue the example, cell G would be transmitted at cell time 23 (not shown). Note that the output cell flow from this process is conforming, as can be checked from the conformance test of the GCRA defined earlier.

The leaky bucket shaper smoothes the input stream and will not drop any cells unless its buffer overflows. A leaky bucket policing algorithm would find this shaped output stream conforming.

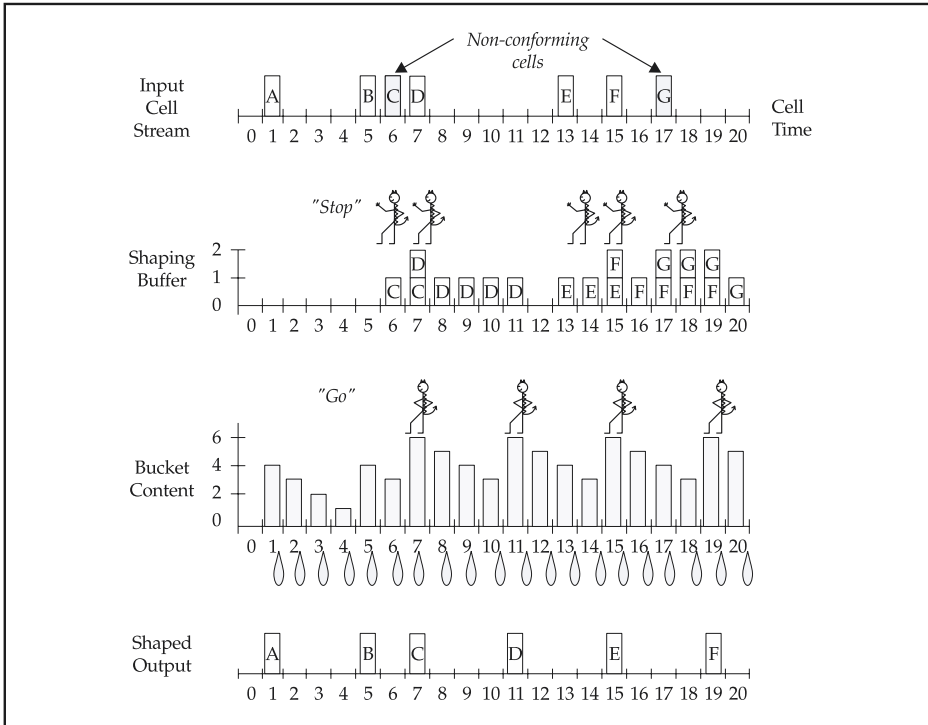


Figure 21-14. Traffic shaping example (buffering)

Token Bucket Shaping

Figure 21-15 illustrates a token bucket shaper. Arriving packets are first stored in a buffer, unless it is full, in which case the arriving packet is discarded. A regulator checks the length of the packet at the head of the buffer L , against the contents of the token bucket X . If there are sufficient tokens, the shaper transmits the packet and decrements the token bucket by the sent packet length L . As usual, tokens are added to the bucket at a rate r , up to a maximum total number of tokens b . Effectively, this combination of buffer and token bucket regulator delays packets until packet transmission would be in conformance with the token bucket parameters.

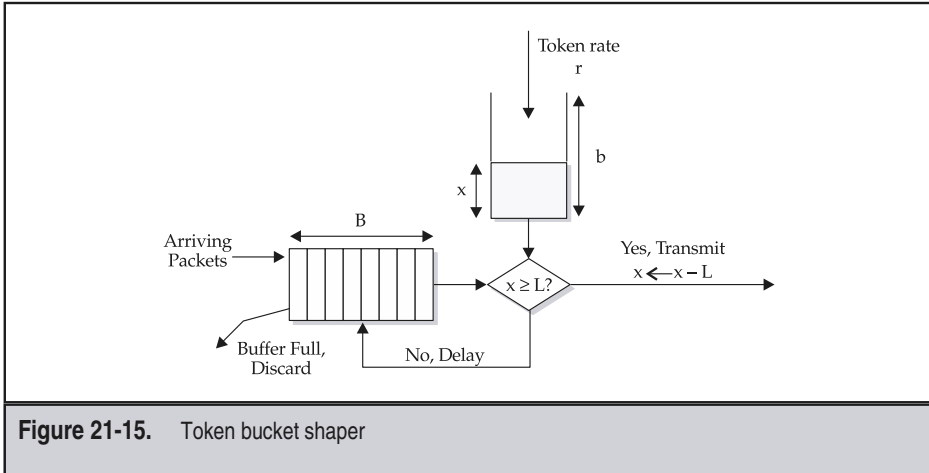


Figure 21-15. Token bucket shaper

Figure 21-16 illustrates a combined peak and average rate token bucket shaper utilizing two buffers with a token bucket and a peak rate regulator as indicated in the figure. The combination of buffers and regulators delays packets until transmission of a packet is in conformance with both the token bucket and peak rate parameters. The logic that compares the number of tokens in the bucket with the length of the first packet in the buffer achieves this objective. The buffer for the peak rate regulator is still b bytes for the maximum length burst M allowed by the token bucket regulator. Note that any arriving packets that find a token bucket reshaping buffer full are nonconforming. This means

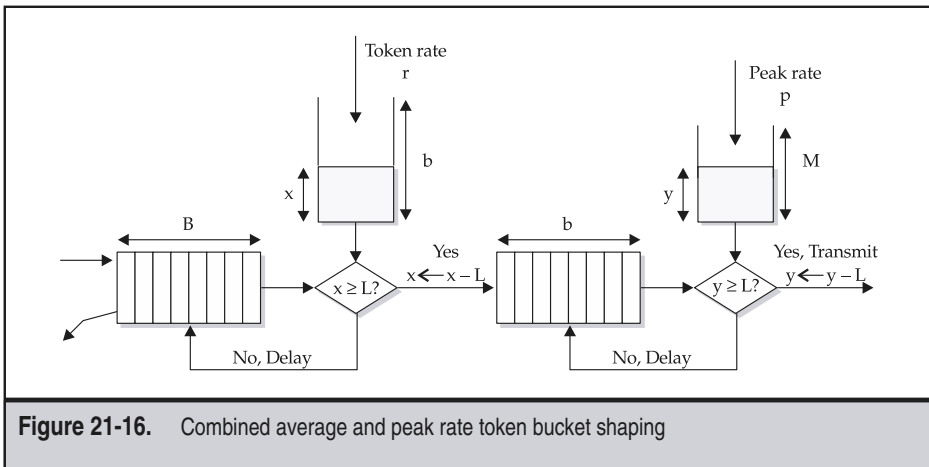


Figure 21-16. Combined average and peak rate token bucket shaping

that a reshaper is effectively a policer as well. The shaping buffer must be of size b bytes to admit a burst of packets conforming to the token bucket. The peak rate shaper operates only on packets conforming to the token bucket parameters.

The next example illustrates a worst-case example of how the peak rate shaper affects the scheduling of packet transmission with reference to Figure 21-17. Starting on the left-hand side at $t = 0$, the peak rate shaper bucket contains M tokens. A conforming packet of maximum length M arrives from the token bucket, followed shortly thereafter by the arrival of another conforming packet of length L . The peak rate shaper transmits the maximum-length packet at the line rate R , which requires M / R seconds. The peak rate shaper must now wait until the token bucket contains enough tokens to begin transmitting the packet of length L . Since the shaper bucket fills at the peak rate p , the required level is L minus the number of tokens that will arrive during transmission of the L byte packet, as indicated in the figure.

The peak rate shaper delays the transmission of the next packet of length L in order to satisfy the peak rate conformance check. This means that $(pt + M) = (L + M)$, which implies that $t = (L / p)$, the time required to transmit a packet at the peak rate. Note that $t = W + L / R$ in the figure, resulting in the solution that the shaper may have to wait up to W seconds before transmitting the next packet of length L as follows:

$$W = \frac{L}{p} - \frac{L}{R}$$

Shaping implies that some arriving packets must wait in order to comply with traffic parameters specified by the peak and average rate and the maximum burst duration.

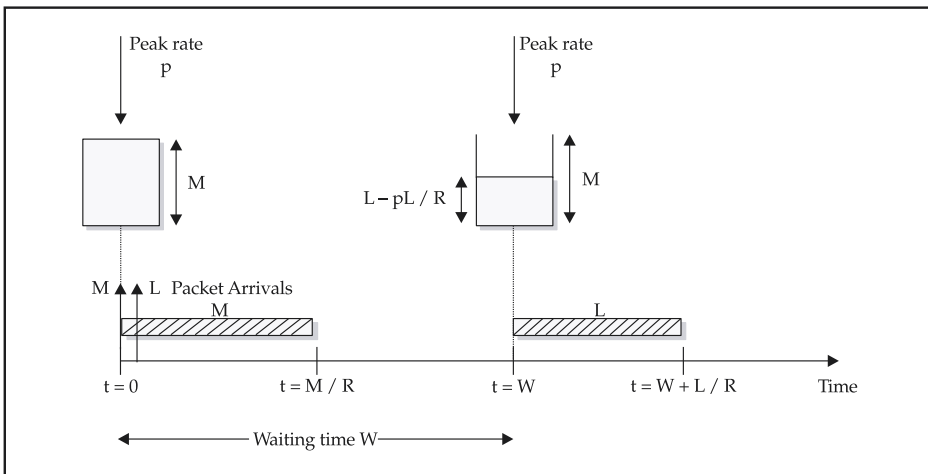


Figure 21-17. Example of packet scheduling for a token bucket peak rate shaper

DELIVERING QoS: PRIORITIZATION, QUEUING, AND SCHEDULING

Priority control helps achieve the full range of QoS loss and delay parameters required by a diverse set of high-performance applications. Generally, prioritization operates to control delay or loss using one of two mechanisms. First, the use of separate queues served in a prioritized manner results in different delay performance for each queue. Second, thresholds within each buffer for different traffic types and marked cells or packets result in different levels of loss performance for each threshold. Operating in concert, these two mechanisms allow ATM and MPLS devices to meet the delay, delay variation, and loss QoS parameters specified in the traffic contract.

Prioritized Queuing and Scheduling

Priority queuing and weighted fair queuing all basically implement multiple queues in the switch, such that delay-intolerant connections or flows can “jump ahead” of those that tolerate delay. IP routers and ATM switches employ *prioritized queuing* to meet different delay and loss priorities for different flows and connections. Switches and routers may perform class-based queuing, or even implement a queue for each connection or flow. We describe prioritized queuing with reference to the block diagram of Figure 21-18. In our example, the prioritized queuing function conceptually resides on the output port of an IP router or ATM switch. The router/switch takes arriving packet/cell streams from multiple input ports, looks up an internal priority value, and directs the packets/cells to the queue on the output port corresponding to the class or individual

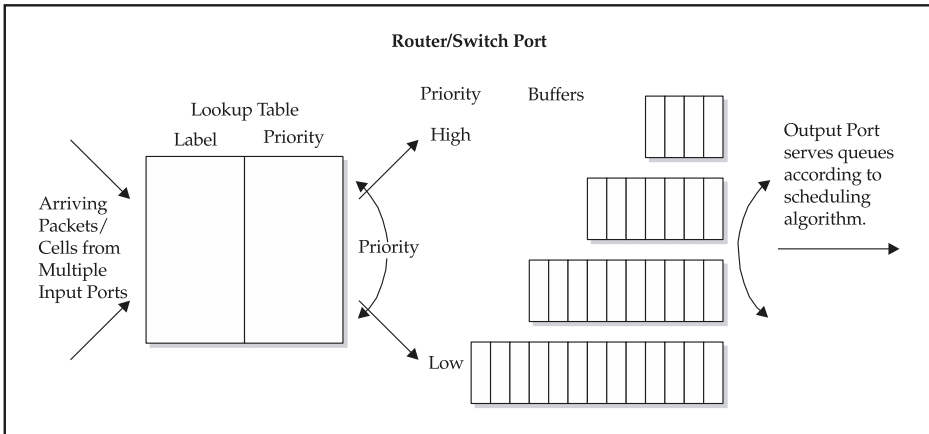


Figure 21-18. Priority queuing operation

flow/connection. The output side of the port serves the queues according to a particular scheduling algorithm.

A simple scheduling algorithm serves the highest-priority, nonempty queue to exhaustion and then moves on to the next-highest-priority queue. This process repeats for each successively lower-priority queue. This scheduling function ensures that the highest-priority queue has the least loss, delay, and delay variation. Consequently, lower-priority queues may experience significant delay variation and delays, as well as loss.

Actual switch and router designs may dedicate a set of buffers to each output port or split the buffers between the input and output ports. Some switches share memory between multiple priorities. Switches employing a shared memory approach usually limit the individual queue size for each port, service class, or—in some cases— individual virtual connections. Although theoretically ideal from a fairness point of view, per-connection queuing does add implementation complexity. The principal benefits of aggregating multiple connections or flows into a smaller number of classes are the reduction in the required amount of state information, and hence reduced complexity. Most queue service disciplines are work conserving; that is, if a packet or cell is in any queue, then the scheduler services it.

Priority Discard Thresholds

Policing allows the ingress network node to either discard traffic that fails to conform to the traffic parameters or else mark nonconforming traffic at a lower priority. The ATM cell header uses the Cell Loss Priority (CLP) bit to indicate whether a cell is of high priority (CLP = 0) or low priority (CLP = 1). The IP Diffserv standard allows implementations to selectively mark nonconforming packets. Specifically, the Diffserv Assured Forwarding (AF) Per Hop Behavior (PHB) supports multiple levels of discard threshold. The experimental field in the MPLS header may also support selective tagging of nonconforming packets. Additionally, the IP integrated services architecture document [RFC 1633] recommends tagging nonconforming packets if such a means is available.

Selective discard gives preferential treatment to higher-priority cells or packets over lower-priority cells or packets during periods of congestion. ATM standards define selective cell discard as the mechanism whereby the network may discard lower-priority flows while meeting Quality of Service (QoS) for higher-priority flows. In ATM, selective discard is an important standardized network equipment function for recovering from severe congestion. In Diffserv AF, discarding occurs in a similar manner for packets that have been marked at a higher discard probability during intervals of congestion. The network may use selective discard to ensure that compliant flows receive a guaranteed QoS. If the network is not congested, then the network may provide higher throughput by also transferring noncompliant traffic, but never less than the reserved amount.

The application may also tag packets or cells as lower priority if it considers them to be of lesser importance. However, user-tagged packets or cells create an ambiguity because intermediate network nodes have no way to discern whether the user or the network policer set the priority indication. If the user sets the priority indication, and the network policer performs tagging, then it may not be possible to guarantee a loss probability for low-priority flows.

Figure 21-19 shows an example implementation of the selective discard mechanism. The router or switch fills a buffer with a fixed number of positions with high- and low-priority packets or cells arriving from the left. The physical layer empties the buffer from the right. When clumps of arrivals occur from other ports on the switch or router, the buffer becomes congested. One simple way of implementing selective discard is to set a threshold above which the port discards any incoming low-priority traffic but still admits high-priority traffic. Note that high-priority packets or cells may occupy any buffer position, while low-priority packets or cells may occupy only the portion of the buffer to the right of the threshold as indicated in the figure. Therefore, by controlling the buffer threshold, the network controls low-priority loss performance. A refinement of this idea involves flushing out low-priority traffic after crossing another threshold, but this increases the complexity of implementation.

Performance Implications of Priority Control

In some configurations, high- and low-priority traffic must share the same queue. A commonly used technique to deliver different levels of QoS to different types of traffic is to use a threshold in a queue, as illustrated in Figure 21-20a. High-priority traffic (as indicated by 0) can occupy the entire queue, while lower-priority traffic (as indicated by 1) can occupy only the portion of the queue below the threshold. If there are multiple queues in a switch or router, each queue may or may not have a discard threshold. For example, in ATM CBR and Diffserv EF, there is no concept of selective discard, while the ATM VBR service categories and Diffserv AF do have such a concept. Queuing has a significant impact on the values of packet loss and delay variation, which differs for the thresholded queues, as shown in Figure 20b. Since buffer-length bounds delay variation

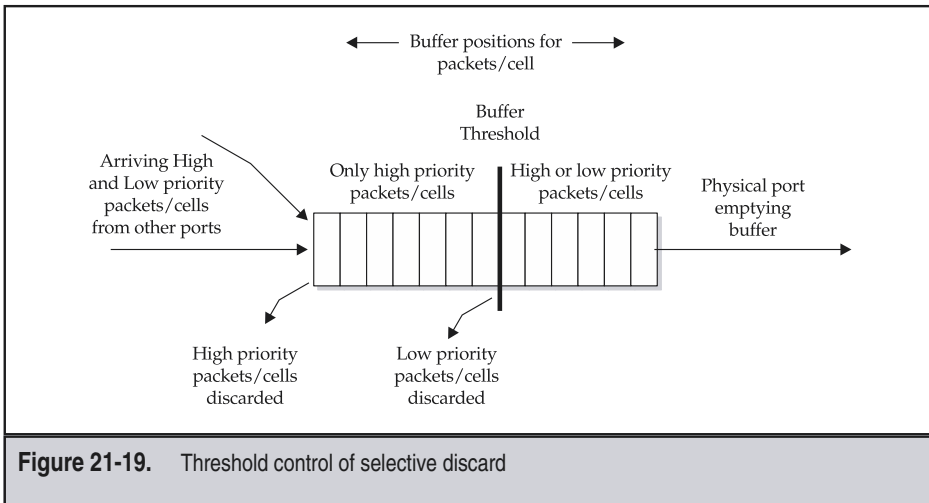


Figure 21-19. Threshold control of selective discard

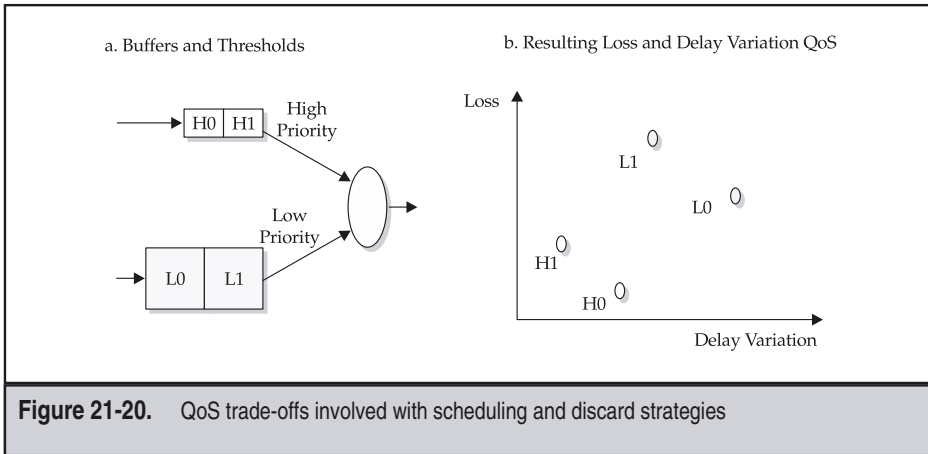


Figure 21-20. QoS trade-offs involved with scheduling and discard strategies

and the port services the shorter buffer at the highest priority, the traffic served by this buffer experiences the best delay variation and loss performance. Furthermore, the traffic that can occupy the entire buffer (H0) experiences lower loss but greater delay variation than the traffic that can occupy only part of the buffer (H1), as determined by the admission threshold. Since the port services the longer buffer at a lower priority, loss and delay variation are both greater. In an analogous fashion, the traffic served by the entire buffer (L0) has greater delay variation but lower loss than traffic that can utilize only the portion of the buffer (L1) determined by the admission threshold. Note that although L1 and L0 both have worse loss and delay variation performance than H1 and H0 in this example, this relationship is not true in general.

Another important consideration in a shared buffer approach is the impact on higher-layer protocols. TCP traffic has the highest throughput when the buffer size is comparable to the product of the round-trip delay and the bottleneck bandwidth. Unfortunately, if TCP is assigned the lower-loss priority (1), it receives a smaller amount of buffer. Therefore, as recommended in the earlier sections of this document, class-based, IP-QoS-aware queuing (i.e., a separate queue per class) is essential at all potential bottleneck points in the network.

Overview of Weighted Scheduling Algorithms

Other scheduling algorithms spread out the variation in delay across the multiple queues. For example, a Weighted Fair Queuing (WFQ) scheduler sends out cells just before reaching the maximum delay variation value for cells in the higher-priority queues [Parekh 93, Keshav98, Stallings 98, McDysan 00]. As we shall see, this action decreases delay variation in the lower-priority queues. The idealized Generalized Processor Sharing (GPS) model assumes that service can be broken down into infinitesimal amounts. A simpler explanation follows from the use of the notion of bit-by-bit service to

express the units of a virtual clock. Packetized GPS (PGPS), commonly called WFQ, works on whole packet boundaries instead of the idealized bit-by-bit method described in this section.

The definition of GPS is closely tied to the definition of the token bucket defined earlier, which associates an average rate r_x and token bucket depth b_x with QoS class x , which may be an individual flow or a traffic aggregate. The router/switch port has a separate queue for each class. A scheduler makes the rounds across all connection queues, one bit at a time.

If $N(t) > 0$ classes are active at time t , then the rate that the virtual clock services each queue is inversely proportional to $N(t)$. The scheduler weight ϕ_x determines the guaranteed minimum service rate g_x out of a total available capacity R for the queue servicing class x as follows:

$$g_x = \frac{\phi_x}{\sum_{i=1}^{\phi} \phi_i} R$$

If $g_x \geq r_x$, then the performance of GPS (and also WFQ) for a single node and a network of nodes has bounded delay with no loss [McDysan 02]. WFQ can also be applied to best-effort traffic in conjunction with performance guarantees available to QoS classes with token bucket-limited traffic.

MEETING THE TRAFFIC CONTRACT: RESOURCE MANAGEMENT

This section discusses several important methods widely used in ATM and MPLS networks to manage the allocation of resources so that the required QoS is delivered. The first is that of admission control, which aims to prevent congestion by ensuring that new connections are admitted only up to the point where QoS can still be met. The second involves a simplification of traffic engineering by aggregating many smaller bundles of traffic into larger ones in order to simplify resource management.

Admission Control

Admission control ultimately decides whether to admit or reject the request to add a new flow or connection on the basis of whether the newcomer would violate delivering on QoS for the existing flows or connections.

This section describes an important means that connection-oriented protocols employ to deliver QoS. ATM, RSVP, MPLS, and Diffserv all effectively operate in a connection-oriented paradigm, and therefore they have similar admission control policies. Admission control involves each node checking every request against available capacity and current QoS capabilities. The node admits the request only if it can provide the requested QoS after adding the traffic corresponding to the existing connections.

One commonly encountered technique for implementing admission control is that of schedulable regions [Keshav 98, Hyman 91]. Admission control reserves scheduling resources to ensure QoS for each connection. In simple admission control systems, each connection, or flow, can be assigned to a class with other connections or flows that have the same (or at least similar) QoS and traffic parameters. Figure 21-21 illustrates a simple example for a switch or router serving only two traffic classes. Class 1 has 25 percent of the capacity requirement of class 2. The shaded area indicates the schedulable region for combinations of class 1 and class 2 connections or flows. If a connection request would cause the combination of class 1 and 2 traffic to fall outside of the schedulable region, then admission control would deny the request, since allowing it could degrade the QoS of the already admitted connections.

A problem with the schedulable region approach is complexity. Since the traffic classes are a function of QoS and traffic parameters, the number of classes is potentially quite large. Therefore, precomputing and storing the schedulable region for all of the possible combinations of QoS and traffic parameters is not implemented in real-world switches or routers. Approximating the schedulable region via a simpler algorithm is an important practical design consideration. We give several examples of such designs for ATM and MPLS networks as an illustration of this approach.

ATM Connection Admission Control

Connection Admission Control (CAC) is a function commonly implemented by software in ATM switches that determines whether to admit or reject connection requests. A connection request includes traffic parameters, along with either the ATM service category, the requested QoS class, or the user-specified QoS parameters. ATM switches use CAC to determine whether admitting the connection request at permanent virtual circuit (PVC) provisioning time or SVC call origination time would violate the QoS already guaranteed to active connections. In other words, CAC admits the request only if the network can still

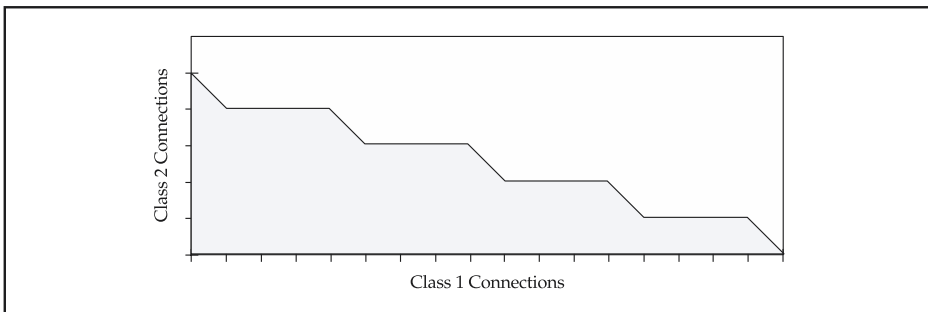


Figure 21-21. Simple example of a schedulable region

guarantee QoS for all existing connections after accepting the request. Frequently, each node performs CAC for SVCs and soft permanent virtual circuits (SPVCs) in a distributed manner for performance reasons. A centralized system may perform CAC for PVCs. For accepted requests, CAC determines policing and shaping parameters, routing decisions, and resource allocation. Resources allocated include trunk capacity, buffer space, and internal switch resources like VPI/VCI lookup table ranges.

CAC must be simple and rapid to achieve high SVC call-establishment rates. On the other hand, CAC must be accurate to achieve maximum utilization while still guaranteeing QoS. CAC complexity is related to the traffic descriptor, the switch queuing architecture, and the statistical traffic model.

The simplest CAC algorithm is peak rate allocation, where the ATM switch simply keeps a running total of the peak rate of all admitted connections. Peak rate allocation CAC denies a connection request if adding the peak rate of the candidate connection to the counter indicating capacity already allocated exceeds the trunk capacity. It adds the peak rate of admitted requests to the running counter, and decrements the peak rate of released connections from the running counter.

Figure 21-22 illustrates peak rate allocation. Starting in the upper-left corner, the ATM device's CAC logic receives a request with a peak cell rate R . The trunk capacity is P , of which a certain portion A is already assigned according to peak rate allocation. If the request R exceeds the available capacity ($P - A$), then CAC denies the request; otherwise, CAC accepts the connection request, incrementing the allocated capacity by R . Actually,

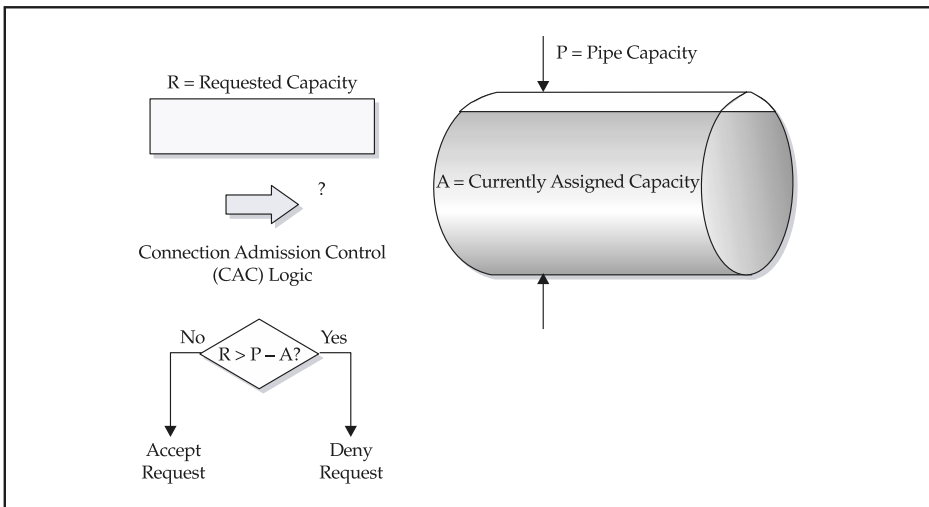


Figure 21-22. Peak rate connection admission control

the admission threshold may be somewhat less than $(P - A)$ due to the slack implied by the compliant connection definition, the CDV tolerance parameter, and the buffer size available for a certain cell loss and delay QoS objective.

Observe that if a network sets the Usage Parameter Control (UPC) (i.e., policing) action to discard cells in excess of the peak rate, and the switches allocate all trunk capacity and buffer resources for the peak rate, then congestion simply cannot occur. The worst that can happen is that arrivals of various streams will randomly clump at highly utilized ATM network nodes, creating some cell delay variation (CDV). A modest amount of buffering keeps the probability of loss quite small in most applications of peak rate allocation, as analyzed in Chapter 24. Extending this design to reserve enough capacity to handle likely network failure scenarios is straightforward.

Although this approach avoids congestion completely, the resulting utilization of the network may be quite low if the average rate is much less than the peak rate, making this a potentially expensive proposition. Note that loose resource allocation policies can make sense in a local area where transmission and ports are relatively inexpensive. The literature refers to the practice of consistently allocating more resource than required as “overengineering.” In other words, the network designer allocates more than enough capacity to the problem at every potential bottleneck point. As stated before, this approach always works—if you can afford it.

In general, a network uses the peak cell rate, sustainable cell rate, and maximum burst size for the two types of CLP flows (0 and 1) as defined in the traffic contract to allocate the buffer, trunk, and switch resources. Peak rate allocation ensures that even if all sources send the worst-case, conforming cell streams, the network still achieves the specified Quality of Service (QoS). Similar CAC algorithms using the SCR and MBS parameters also achieve lossless multiplexing. CAC algorithms may also use a concept called *equivalent capacity*, as introduced in Chapter 15 for PNNI and further detailed in Chapter 24 in an admission algorithm based upon a combination of the PCR, SCR, and MBS as described earlier when used in conjunction with weighted fair queuing. Networks that oversubscribe certain service categories usually employ some form of congestion avoidance or recovery procedures, as described in Chapter 22.

A more aggressive CAC than that of peak rate allocation described earlier is where the network allows a certain degree of oversubscription. Users frequently request traffic parameters, such as the sustainable cell rate, that exceed their typical usage patterns. Hence, if a network takes these traffic parameters at face value, it allocates too much capacity and creates blocking of connection attempts. Furthermore, when a large number of these connections share a common resource, it is unlikely that they all simultaneously use the resource at their peak demand level. Therefore, the network may admit more connections than the traffic parameters indicate could be supported and still achieve the specified Quality of Service (QoS) due to the statistical properties of averaging traffic from many users. These statistical QoS guarantees achieve a good balance between efficiency and quality in well-run networks. Chapter 24 describes the equivalent bandwidth model and its use in predicting statistical multiplexing gain.

MPLS Admission Control

As discussed in Chapter 14, the RSVP-TE and CR-LDP MPLS signaling protocols employ token bucket traffic parameters to communicate the capacity required for an MPLS LSP. Mapping the MPLS terminology to the ATM equivalent results in a similar set of admission control schemes. Support of admission control in MPLS networks serving as a backbone for an ISP is critical in order to meet the traffic engineering requirements described in RFC 2702. Many LSRs implement admission control using a form of bookkeeping, while some actually configure scheduler weights based upon the traffic parameters in the ATM signaling message. One aspect of MPLS signaling that is unique is the concept of negotiating downward the values of the traffic parameters in the event that an LSP cannot be established. This reflects the policy that it is better to have some traffic-engineered connectivity rather than none at all.

ATM VPs and Label Stacked MPLS LSPs

There is a need to manage critical resources in the nodes of an ATM or MPLS network. Two critical resources are buffer space and trunk capacity. One way of simplifying the management of the trunk capacity is through the use of aggregation. ATM and MPLS both allow design of hierarchical traffic aggregation. ATM's fixed-format cell header allows only two levels of hierarchy: the virtual path (VP) and the virtual channel (VC). MPLS, on the other hand, allows for an essentially unlimited level of hierarchy using label stacking. Nodes in MPLS and ATM networks employ label switching as defined in Chapter 4. This means that the packet header labels need only be unique on an individual link. Recall that label switching involves mapping an incoming label to an outgoing label on a per-connection basis. An end-to-end connection is then a concatenation of such label-switching actions. Label stacking occurs when a switch maps a number of connections into another aggregate connection at a higher hierarchical level. Thus, the next higher-level flow contains the aggregate of many lower-level flows.

Figure 21-23 illustrates the generic operation of two-level label stacking for a simple ATM or MPLS network. This example illustrates the operation of label stacking and the resulting flow aggregation. The shading and pattern of the packets illustrate the end-to-end connections between lowest-level nodes. Highest-level nodes can either encapsulate multiple lower-level flows within a common label, or else handle the flow without stacking any labels (see the flow from A.1 to B.1 via nodes A and B). Such flow aggregation eases the task of network routing and traffic engineering by reducing the number of required connections.

Recall from Chapters 10 and 11 that an ATM VP contains many VCs, and that VP cell relaying operates only on the VPI portion of the cell header. If every node in a network is interconnected to every other node by a VPC, then only the total available entry-to-exit VPC bandwidth need be considered in admission control decisions. A VPC is easier to manage as a larger aggregate than multiple, individual VCCs. The complexity and number of changes required to implement routing, restoration, and measurement also are reduced by VPCs as compared to VCCs.

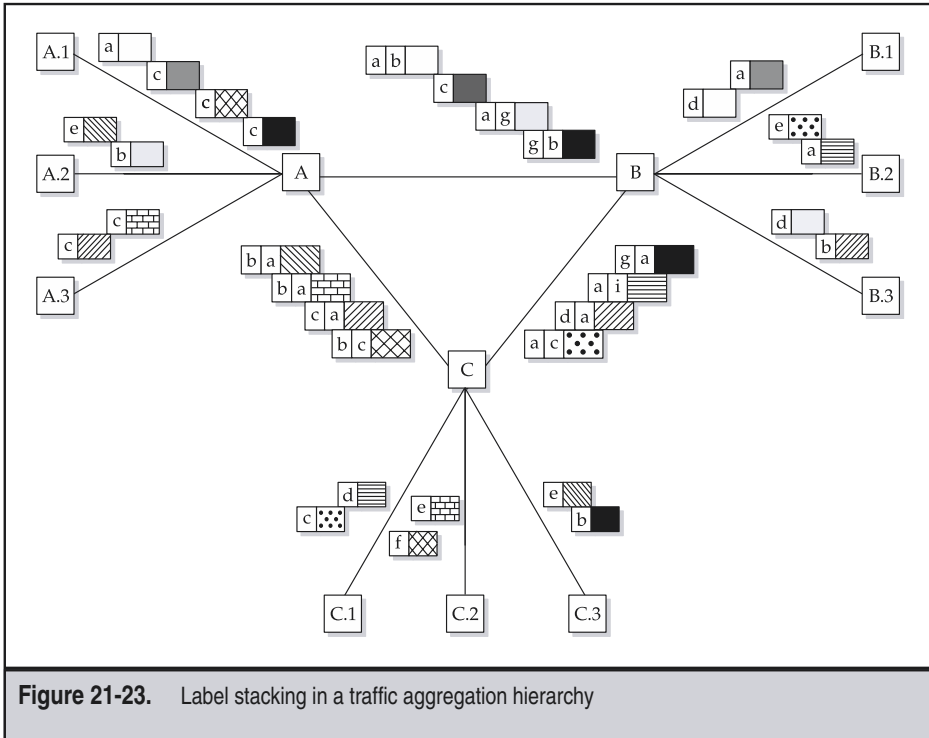


Figure 21-23. Label stacking in a traffic aggregation hierarchy

Note that QoS is determined by the VCC with the most stringent QoS requirement in a VPC. One could envision a network of nodes interconnected by a VPC for each QoS class; however, this could quickly exhaust the VPI address space if there are more than a few QoS classes. Unfortunately, a full-mesh design does not scale well. Even in partial mesh networks, allocating VPC capacity efficiently is a challenge. The principal issue is the static nature of VPC allocation in current ATM standards.

There are some analogies for traffic engineering between ATM VPCs and using label-stacked MPLS LSPs. The notion in using MPLS for IP traffic engineering is that of a traffic trunk [RFC 2702], which is a set of IP packets between a pair of nodes. For example, the packets offered to a traffic trunk may be completely defined by a set of destination IP addresses. An MPLS LSP could be set up as a traffic trunk from every ingress router to the egress router that handles this set of destination IP addresses. The notion of traffic trunks can also be done at one or more levels in the hierarchy. For example, in order to reduce the full mesh of LSPs to improve scalability, a set of traffic trunks formed by aggregate LSPs between core LSRs could be established, over which other LSPs could be label-stacked [Malis 2001, Hummel 2002].

REVIEW

This chapter covered the topics of traffic control, means to provide QoS control, and resource management, summarizing where these functions reside in real-world user equipment and networks. The text gave reference models for ATM and IP/MPLS nodal functions and characterized the traffic and congestion control schemes based upon the time scale over which the control operates. We then described how ATM and IP/MPLS use policing functions to check whether arriving cells or packets conform to a set of traffic parameters specified for the flow and what actions a policer can take. For ATM, we summarized the formal Generic Cell Rate Algorithm (GCRA), which precisely defines the operation of leaky bucket policing. For IP and MPLS, we described the operation of a token bucket in detail.

The chapter then covered the means by which a user ensures conformance to the traffic contract by “shaping” the cell or packet flow using one of several methods. The discussion then turned to the topic of how nodes can achieve the QoS required by different service classes. The methods described included prioritized queuing, discard thresholds, and scheduling. Next, we discussed how admission control operates at the connection-level time scale to efficiently allocate resources and meet QoS guarantees. The chapter concluded with the use of ATM- or MPLS-based traffic aggregation as a means to simplify this problem.

CHAPTER 22

Congestion Control

Webster's *New World Dictionary* defines *congest* as "to fill to excess; overcrowd[; for example,] a congested highway." Although the best solution to congestion is to simply avoid situations where and when congestion is likely to occur, this strategy isn't always possible. Unfortunately, congestion occurs in many real-world networking environments because there is always a bottleneck of some sort—a slow computer, a low-speed link, or an intermediate switch or router with low throughput. Therefore, depending upon the severity and duration of congested intervals, networks and operators can take different levels of response.

This chapter begins by relating experiences from well-known congestion phenomena in everyday life to communication networks. We then define congestion in terms specific to IP and ATM networks through use of simple examples. Depending upon the type of traffic, in conjunction with the capabilities of network elements and end-user software, congestion control takes on one of two basic flavors: open loop or closed loop. The chapter then discusses the principal performance measures of congestion control algorithms: throughput and delay.

We further categorize congestion control protocols as a framework for discussing the application of specific IP, MPLS, and ATM responses to congestion. The options for congestion indication and control range from the proactive to the cooperative and, when all else fails, the reactive. Proactive congestion management involves good long-term planning, traffic measurement, and network engineering. The next set of options involves congestion avoidance by indicating congestion, simply blocking new connection requests, and the operation of closed-loop flow control. We study the ATM Available Bit Rate (ABR) as a closed-loop flow control method of effectively avoiding congestion. Finally, we describe the last recourse: reactive congestion recovery procedures. If network congestion reaches the need for recovery, then all of these techniques discard traffic in some way. The difference between them is how they choose the traffic to discard, and what the eventual impact on higher-layer applications becomes.

CONGESTION: A FAMILIAR PHENOMENON

This section introduces the topic of congestion by drawing analogies with situations that many of us have experienced. We then apply analogies of these experiences to IP, MPLS, and ATM networks.

The Nature of Congestion

We experience congestion daily in the form of traffic jams, long checkout lines at stores, ticket lines, or just waiting for some form of service. Congestion is the condition reached when the demand for resources exceeds the available resources for an extended interval of time. Take the real-life example of a vehicular traffic jam. Congestion occurs because the number of vehicles wishing to use a road (demand) exceeds the number of vehicles that can travel on that road (available resources) during a rush hour (an extended time interval).

More specific to IP, MPLS, and ATM networks, congestion is the condition where the offered load (demand) from the user to the network approaches, or even exceeds, the network design limits for guaranteeing the Quality of Service (QoS) specified in a service level agreement (SLA). This demand may exceed the resource design limit because the network incorrectly oversubscribed resources, because of failures within the network, or because of operational errors.

Congestion is inevitable in any expanding system. As traffic increases and the resources remain static, an overload eventually must occur. This pattern repeats itself in transportation systems around the world. For example, the road systems once adequate for a quiet rural suburb of a large metropolitan area are jammed with commuters in areas with increasing population.

On the other hand, some systems seldom experience congestion, since other mechanisms or usage patterns keep demand below capacity. For example, some metropolitan centers experience little-to-no congestion during the evening hours and on weekends, since there are few residences or entertainment centers in the downtown area. Contrary to some articles in the popular press, the declining cost of transmission, memory chips, and computer processing power will not alleviate congestion. Changing cost structures frequently just move the bottleneck that causes congestion from one place to another.

Congestion is often a dynamic phenomenon caused by unpredictable events. Although the rush hour may be predictable on the freeway, an accident can cause an even more severe traffic jam at any time of day or night. Combine an unexpected event like an accident with a normal rush hour and the delays can stretch to hours.

Why should you be concerned with congestion in networks? Because, as in transportation systems, shopping malls, and other areas where many people congregate: congestion is an unpleasant experience. In IP, MPLS, and ATM networks, congestion causes excessive delay, loss, or both, which often results in unhappy customers. Delay and loss reduce the quality of the service provided to the end user application, often lowering productivity and sometimes rendering the application unusable.

Busy Seasons, Days, and Hours

Some days, traffic is heavier than others. Usually, in transportation networks, congestion occurs on a predictable basis at specific intersections or thoroughfares. Similar phenomena exist in many networks between specific communities of interest like clusters of geographic regions or departments in an enterprise. For example, Mother's Day is usually one of the busiest days in telephone networks.

A pattern typically called a *busy hour* exists in many networks much as it does during rush hour on the freeway. These observations of overall arrival rates averaged over many days during different seasons of the year differ from the random arrival model studied in the previous part. Instead of a single parameter describing the arrival process, a traffic pattern is modeled as an average arrival rate for a specific time of day, day of week, and season of the year. Similar busy hour and busy day traffic patterns also occur on Internet backbones [Thompson 97].

At other times, exceptional conditions can create overloads in unexpected places at unusual times. Loads during the busiest intervals, or during abnormal periods of higher-than-usual activity, create congestion in communication networks, much as overloads occur in transportation systems due to natural disasters or accidents. Occasionally, popular, newsworthy, or emergency events create overloads in telephone networks or on the Web.

Impact of Congestion

A number of application characteristics determine the impact of congestion, such as connection mode, retransmission policy, acknowledgment policy, responsiveness, and higher-layer flow control. In concert with the application characteristics, certain network characteristics also determine the response to congestion, such as: queuing strategy, service scheduling policy, discard strategy, route selection, propagation delay, processing delay, and connection mode.

As discussed in Chapter 22, congestion occurs on several time scales: individual cells or packets, bursts of packets, the source-destination round-trip delay, the duration of a connection, or a provisioning interval. The detection of congestion as a prelude to subsequent action is *congestion indication*, feedback, or notification. Traffic forecasts, utilization trends, buffer fill statistics, cell or packet transmission statistics, or loss counters are all indications of congestion. The reaction to indicated congestion occurs in either time or space. In *time*, reactive controls operate on either a cell-by-cell basis, on a packet (or burst) time scale, or at the call level. In *space*, the reaction can be at a single node, at the source, at the receiver, or at multiple nodes.

Examples of Congestion in a Network

In IP, MPLS, and ATM networks, the congestible resources include buffers, transmission facilities, or processors. We call the resource where demand exceeds capacity the *bottleneck*, congestion point, or constraint. Figure 22-1 illustrates an example of congestion occurring along an end-to-end route from a server connected to Node A to a user connected to Node D. Thicker lines interconnecting the nodes indicate links with higher capacity than those with thinner lines. The number of dots next to each link indicates the rate of bursts (of cells or packets) being transferred by each link. In the following examples, a thick link can transfer at a rate of three bursts, while a thin link can transfer one burst per time interval.

Two points of congestion occur in the example of Figure 22-1. First, the high-speed link between Nodes C and D is congested due to overload. As shown in the figure, the average input load to Node C is six bursts, while the output link from C to D can carry only three bursts. If this overload of input rate exceeding output rate persists long enough, Node C will drop packets or cells due to buffer overflow or depletion of other resources. The second point of congestion occurs on the low-speed link between Node D and the user at the right-hand side of the figure. Congestion occurs because the low-speed link can support a rate of only one burst per time interval, while the server on the left-hand side of the figure is sending two bursts per time interval. Therefore, if this speed mismatch persists, Node D must eventually discard packets or cells due to lack of resources.

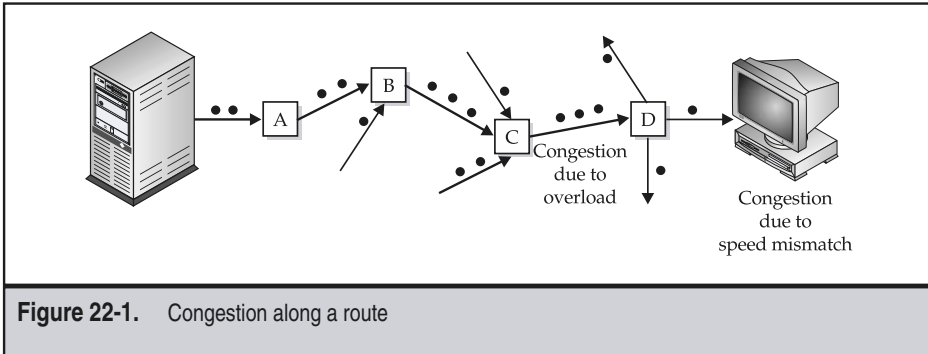


Figure 22-1. Congestion along a route

CONGESTION CONTROL: A RANGE OF SOLUTIONS

The best solution to congestion is to simply avoid situations where and when congestion is likely to occur. Unfortunately, this strategy isn't always possible. Congestion can occur in any real-world networking environment because there is always a bottleneck of some sort at some time—overload due to unanticipated demand, an intermediate switch or router with low throughput or excessive delay, or a focused overload due to a failure of one or more links or nodes. This section provides an introduction to the basic concepts of congestion control techniques and places the ones described in the remainder of the chapter into a taxonomy.

Okay, once congestion occurs, what can be done? The next two examples illustrate the basic paradigms studied in the remainder of this chapter.

Open- and Closed-Loop Congestion Control

Figure 22-2 illustrates the first case where voice communication users connected to Nodes B and D share the congested link connecting Nodes C and D. In this example, the voice bursts (shown as circles with white centers) have higher priority, and hence get through first. However, the communication link between C and D has limited capacity, hence the throughput of the server-to-user connection between Nodes A and D must decline. Thus, prioritization and ensuring that sufficient capacity exists on the selected route for high-priority traffic is one means of controlling congestion. Since this mechanism operates without any feedback, we call it *open-loop congestion control*.

Figure 22-3 illustrates the second major method of congestion control. Here, we have the same two congestion scenarios shown in Figure 22-1, but in this case, the sender obtains feedback regarding the congested state of the flow. Since the feedback comprises a closed loop from the sender to the receiver and back to the sender again, we call this technique *closed-loop congestion control*. The most commonly used type of feedback is implicit. The Transmission Control Protocol (TCP) widely used on the Internet for the Web, e-mail, and

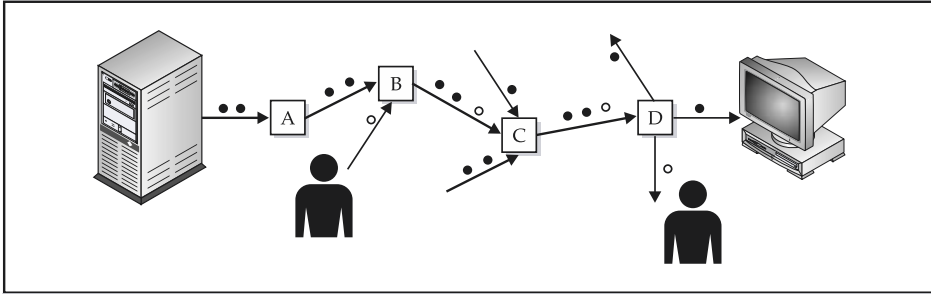


Figure 22-2. Open-loop congestion control—priority service

file transfer employs this technique. As described in Chapter 8, TCP requires that the receiver acknowledge receipt of each packet to the sender. If the sender does not receive an acknowledgment after a time-out interval, then the sender implicitly determines that the unacknowledged packet was lost, and retransmits it. Furthermore, once the sender implicitly detects congestion via a time-out, it reduces the rate at which it transmits packets, thereby reducing congestion.

The arrows at the bottom of Figure 22-3 illustrate another type of feedback generated explicitly at the point of congestion. Since Node C is aware that the link connecting C and D is congested, it could send feedback messages to the sources traversing the congested link. ATM's Available Bit Rate (ABR) closed-loop flow control technique uses this approach. Similarly, if the application running on user D's computer was the bottleneck creating congestion, it could explicitly signal the source to slow down. This technique addresses a problem that has existed since the earliest days of computer communications of interfacing a fast sender with a slow receiver. IP's ICMP source-quench protocol uses the explicit feedback technique.

Impact of Congestion on Performance

Two basic measures define the degree of congestion experienced—throughput and delay. The file transfer and voice/video applications represent extremes of application requirements for throughput and delay. *Throughput* is the data transfer rate actually achieved by the end application. For example, if a File Transfer Protocol (FTP) application loses a packet, then it must retransmit that packet and (frequently, in many implementations,) all of the packets sent after it! Useful throughput consists of only those packets actually sequentially delivered to the end application without errors. Some applications, such as FTP, accept variable throughput; while others, such as voice and video, require a specific value of throughput to work acceptably.

Delay requirements differ by application type. Real-time traffic must be delivered within a fraction of a second, while for non-real-time applications that perform retransmission, delay takes on an additional dimension. When a protocol retransmits unsuccessfully

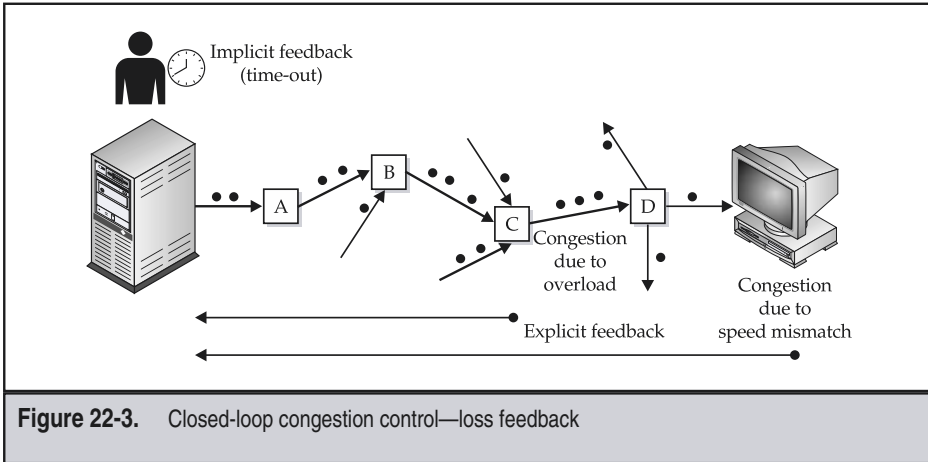


Figure 22-3. Closed-loop congestion control—loss feedback

delivered packets, the resulting delay is the time elapsed between the first unsuccessful transmission and the final successful reception of the packet at the destination.

Loss is another important consideration in congestion control. Some applications, such as video, can adapt their transmission rate and still deliver good performance if network congestion control minimizes loss. Other activities—such as Web surfing, file transfer, and e-mail—recover from loss via retransmission, usually with little user impact except for increased delay.

In general, applications that do not use retransmission should experience the same throughput, delay, and loss on the underlying IP, MPLS, or ATM network. For example, voice or video coded to operate acceptably under loss conditions is not retransmitted, and hence experiences the same throughput and delay as the underlying network. In practice, voice and video coding accept loss or delay only up to a critical value; after that point, the subjective perception of the image, or audio playback, becomes unacceptable.

Figure 22-4 plots effective throughput versus offered load. An ideal system has throughput that increases linearly until offered load reaches 100 percent of the bottleneck resource. A good congestion control protocol approximates the ideal curve. A poor congestion control scheme exhibits a phenomenon called *congestion collapse* [Jain 88]. As offered load increases toward 100 percent, throughput increases to a maximum value and then *decreases* markedly due to user application retransmissions caused by loss or excessive delay. Hence, we say that throughput collapses at the onset of congestion.

A key measure of the effectiveness of a particular congestion control scheme is how much delay or loss occurs under offered loads approaching or exceeding 100 percent of the bottleneck resource. Figure 22-5 illustrates the effective delay for the same three categories of congestion control systems described previously. An ideal congestion control system has bounded delay at all values of offered load up to 100 percent, at which point

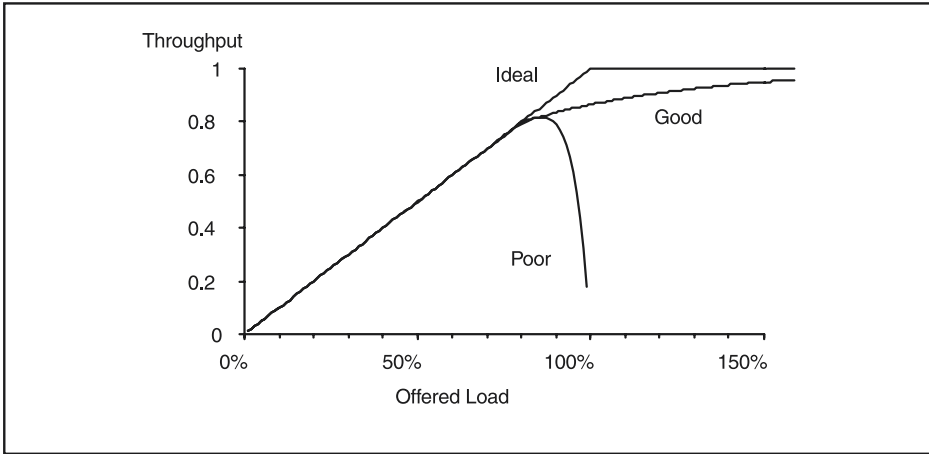


Figure 22-4. Useful throughputs for congestion control schemes

delay becomes infinite. A good congestion control protocol exhibits increased delay only as severe congestion occurs. A poor congestion control protocol has delay that increases markedly before the system reaches full utilization.

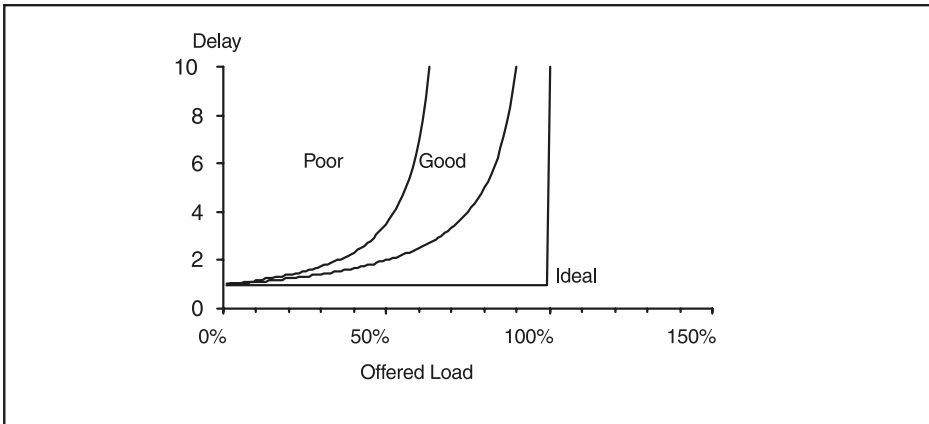


Figure 22-5. Effective delay for congestion control schemes

Clearly, achieving high throughput and low delay simultaneously is not possible. In order to optimize this trade-off, traffic engineers devised the concept of queuing power [Jain 88]. Since the objective is to balance the goals of maintaining higher throughput while simultaneously achieving lower delay, queuing power is the following ratio:

$$\text{Queuing Power} \equiv \frac{\text{Throughput}}{\text{Delay}}$$

Observe that queuing power trades off throughput against delay by comparing the ratio instead of absolute values. Figure 22-6 plots the queuing power for the ideal, good, and poor congestion control systems described previously. The ideal system exhibits power that is optimal at 100 percent load. An example of such a system is TDM. A good congestion control system for a packet-switched network has queuing power that has an optimum value at a modest utilization level. A poor congestion control system has optimum power at a low utilization level. Hence, queuing power is a useful measure for comparing the relative performance of various congestion control schemes.

CATEGORIZATION OF CONGESTION CONTROL APPROACHES

Figure 22-7 depicts a taxonomy of congestion control approaches adapted from Reference [Yang 95]. The first level of categorization is whether congestion control operates according

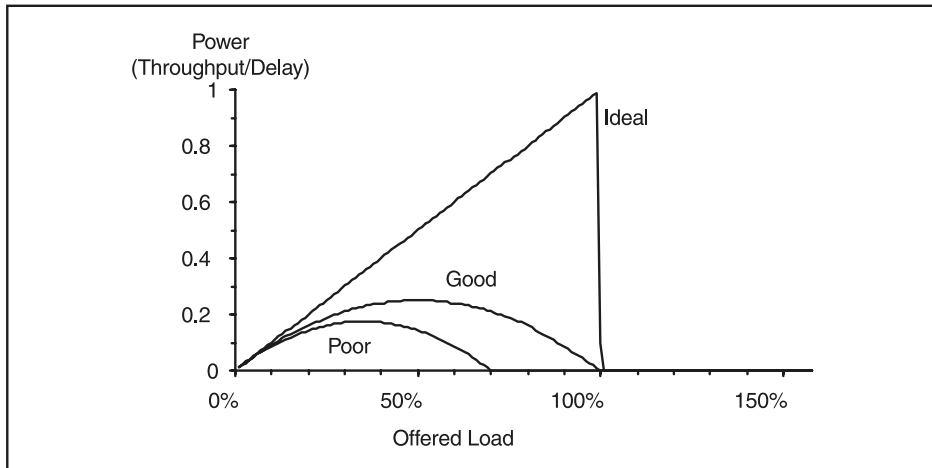
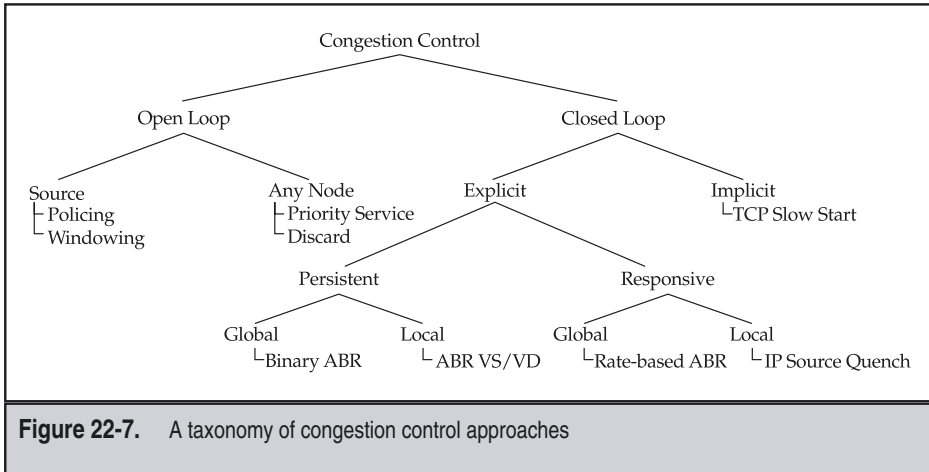


Figure 22-6. Queuing power for congestion control schemes



to either an open-loop or closed-loop control system paradigm. Open-loop protocols include policing or fixed window flow control schemes (e.g., X.25, as described in Chapter 7) invoked at the source. At any node, open-loop schemes include priority service (e.g., weighted service, priority queuing, or selective discard). Examples of open-loop control are Cell Loss Priority (CLP) discard in ATM and Random Early Detection (RED) in the Internet.

Moving to the right-hand branch of the tree, closed-loop control protocols have either explicit or implicit feedback. The premier example of a congestion control protocol employing implicit feedback is the Internet's Transmission Control Protocol (TCP), as covered in Chapter 8. Under the class of algorithms employing explicit feedback, there are two subcategories: persistent and responsive. A persistent algorithm continuously generates feedback, while a responsive algorithm generates feedback only when congestion actually occurs. Either of these classes of algorithm may operate either locally or globally. The suite of ATM's Available Bit Rate (ABR) closed-loop flow control algorithms all operate using explicit feedback; and in Chapter 13 we also looked at another ATM closed-loop feedback algorithm for signaling congestion control. Also, IP's source quench protocol is an example of an explicit feedback protocol operating locally.

The terminology for the categorization of traffic and congestion control varies, and the organization of this chapter uses another point of view, by categorizing responses to congestion as management, avoidance, and recovery. Each of these may operate at the cell or packet level, the burst or flow level, or else the connection or aggregate level, as illustrated in Table 22-1.

Congestion management has the objective of proactively controlling traffic such that congestion is unlikely to ever occur. At the cell or packet level, this involves controlling sources via shaping; while at the burst or flow level, it involves allocating access link or network core resources adequately such that all MPLS or ATM connections have adequate

Category	Cell/Packet Level	Burst/Flow Level	Connection/ Aggregate Level
Management	Shaping	Resource allocation	Network engineering, admission control
Avoidance	Policing, tagging	Window, rate, or credit flow control	Attempt blocking
Recovery	Selective discard, dynamic policing	Congestion indication	Disconnection, operations procedures

Table 22-1. Congestion Control Categories and Levels

capacity. At the connection or aggregate level, admission control usually achieves this objective. We discussed these approaches in Chapter 21, and this chapter adds some discussion on resource allocation and network engineering. The remainder of this chapter focuses on congestion avoidance and recovery.

Congestion avoidance is a set of real-time mechanisms designed to prevent entering a severely congested interval during periods of coincident peak traffic demands or transient network overloads. One example of its use is when nodes and/or links have failed. Congestion avoidance procedures operate best during periods of moderate congestion. Policing to control the traffic admitted to the network and tagging of cells or packets that do not conform to the traffic contract are methods commonly used at the cell or packet level. This includes tagging using the ATM CLP bit or the drop precedence in Diffserv assured forwarding (AF). At the burst or flow level, this involves window-, rate-, or credit-based closed-loop flow control at either the ATM layer (e.g., ABR or GFC) or higher layers (e.g., TCP). At the connection or aggregate level, blocking of new attempts is a method commonly used by admission control to avoid impending congestion.

Congestion recovery procedures operate after congestion has been detected and strive to prevent severe degradation of Quality of Service (QoS) delivered by the network. Typically, networks utilize these procedures only after loss or markedly increased delay occurs due to sustained congestion. At the cell level, congestion recovery includes selective cell discard based upon the CLP bit and dynamic setting of policing parameters. At the packet level, for ATM it includes Early or Partial Packet Discard (EPD/PPD), and for IP, Random Early Detection (RED), as described in Chapter 8. At the burst level, congestion recovery includes ATM Explicit Forward Congestion Indication (EFCI) and Explicit Congestion Notification in the IP packet header. At the connection or aggregate level, congestion recovery includes disconnection of existing connections, and/or operational procedures.

CONGESTION MANAGEMENT

Congestion management strives to ensure that the network never experiences congestion, through careful planning and conservative operation. As described in Chapter 21, when traffic is shaped to a set of parameters, a subsequent policer will likely find the arrival of packets or cells conforming, if the shaper and policer use consistent algorithms and intervening network elements do not perturb the intercell or packet spacing. As also discussed in Chapter 21, admission control allocates capacity and resources to each ATM virtual connection or MPLS LSP. In order to optimize network resources to satisfy the expected number and capacity of connection requests, service providers perform network traffic engineering. This section covers the topics of resource allocation and network engineering not already covered in the previous chapter.

Resource Allocation

Successful network designers strive to allocate resources and parameters of the following types in ATM switches and MPLS label switching routers (LSRs) to prevent congestion. Examples of these resources include physical trunk capacity and buffer space.

The manner in which a network allocates resources to meet a balance between economic implementation cost and the degree of guaranteed QoS is, of course, a network decision. For example, ATM networks may optionally allocate resources to CLP = 1 flows, although most don't. Many network providers oversubscribe ATM Variable Bit Rate (VBR) traffic to achieve economies of statistical multiplexing across multiple bursty data streams generated by a large number of users. Some ATM network providers offer best effort, or Unspecified Bit Rate (UBR) service to fill in capacity unused by higher-priority services.

The network engineer is responsible for allocating sufficient resources to meet the performance requirements for the expected traffic mix. Toward this end, admission control makes a connection-by-connection decision on whether to admit, or reject, a request in accordance with available resources and network policy. Therefore, the resource allocation must be sufficient to meet anticipated demand. Note that policy and implementation may vary. For example, the resources for all QoS classes may be in a single shared pool, or placed in separate pools in order to achieve isolation between QoS classes.

Network Engineering

One method for efficiently allocating resources is to base such decisions upon long-term, historical trending and projections. This is the method used in most large private and public MPLS and ATM networks today. In the near-to-medium term, this includes deciding on how to route each connection to meet QoS requirements, meet capacity constraints, and/or balance the load across multiple paths. In the longer term, the decisions involved also include designing an overall network, deciding on when and where to install or upgrade switches, homing users to switching devices, and installing transmission capacity. Many network operators collect various statistical measurements and actual performance data to accurately model the offered traffic. They use this as input into network

planning tools that provide specific answers to “what if” questions regarding candidate network upgrades or changes. Toward this end, Part 6 discusses commonly used traffic source models, resource models, and traffic engineering methods.

CONGESTION AVOIDANCE

Congestion avoidance attempts just that—to avoid severe congestion—while simultaneously keeping the offered load at the “knee” of the delay versus offered load curve of Figure 22-5. This is analogous to life in the fast-paced modern world where we try to travel either just before, or just after, rush hour. This section covers the following congestion avoidance methods:

- ▼ Congestion indication
- Policing and tagging
- ▲ Connection blocking

The next major section covers an important class of congestion avoidance techniques—namely, flow control.

Congestion Indication

An ATM network element in a congested state may set the Explicit Forward Congestion Indication (EFCI) payload type codepoint in the cell header for use by other network nodes or the destination equipment to avoid prolonged congestion. Typically, ATM switches set the EFCI bit when the number of cells queued in a buffer exceeds a threshold. A network element in an uncongested state should not modify EFCI, since intermediate nodes use EFCI to communicate the existence of congestion to any downstream node. As described later, the ABR service category interworks with ATM devices that set EFCI in binary mode.

As described in Chapter 7, Frame Relay (FR) has a similar congestion indication called Forward Explicit Congestion Notification (FECN). Additionally, Frame Relay also has a Backward Explicit Congestion Notification (BECN), while ATM’s ABR supports a similar concept using the Congestion Indication (CI) bit in the Resource Management (RM) cell. One reason that backward congestion notification wasn’t included in the ATM cell header (as it was in the Frame Relay header), was that experts believed the destination application protocol should communicate to the source application protocol the command to slow down transmissions when experiencing network congestion.

Intermediate equipment, such as routers, have a fundamental issue with utilizing congestion indication information. If they slow down and the source application protocol does not, then loss will occur anyway in the router. Therefore, most routers don’t do anything with Frame Relay or ATM congestion indications. However, some routers do give priority to routing messages in response to congestion indications to ensure that layer 3

routing protocols remain stable. In addition, many routers collect statistics on the number of congestion messages received, which is useful network planning information.

The congestion indications from FR (FECN and BECN) and ATM (EFCI) have not been widely utilized by end systems or higher-layer protocols such as Transmission Control Protocol (TCP). As described in Chapter 8, the IETF has defined a TCP-level standard called Explicit Congestion Notification (ECN) for this purpose. The EXP bits in the MPLS shim header could be used to carry ECN information.

Policing and Tagging

As detailed in Chapter 21, a policer enforces the traffic parameters by subjecting nonconforming traffic to either discard or tagging with a higher drop probability. In ATM, Usage Parameter Control (UPC) tags cells by changing the Cell Loss Priority (CLP) bit to indicate the nonconforming cells (CLP = 1). This action allows admission of traffic in excess of the traffic parameters to the network, which may cause congestion to occur. If a network employs UPC tagging for congestion avoidance, then it should also implement a corresponding technique, such as selective cell discard or dynamic UPC, to recover from intervals of severe congestion. In IP, only the Diffserv AF per-hop behavior (PHB) has a standard means of indicating different levels of drop precedence.

Connection Blocking

Before the network becomes severely congested, admission control can simply block any new connection requests. An example of this type of congestion avoidance is that which occurs in the telephone network—if there is blockage in the network, callers get a fast busy signal. Another example derived from telephone networks is *call gapping*. If a network detects high blockage levels for calls to the same destination number or area, the network places gaps between attempts destined for the congested number or area. In other words, the network processes only a fraction of the attempts to the congested destination, returning a busy signal or message to the gapped call attempts. These approaches help networks avoid or even recover from severe congestion for ATM SVC services, but they do little to help an ATM PVC or MPLS LSP network avoid congestion.

CLOSED-LOOP FLOW CONTROL

Many data communications applications hungrily utilize as much available bandwidth as possible, thereby creating the potential for congestion. The basic idea of congestion avoidance is to back off offered load just before any loss occurs in the network, thus maximizing usable throughput. Furthermore, the network should fairly dole out bandwidth to contending users. In other words, no one user should get all of the available bandwidth of a bottleneck resource if several users equally contend for it. Additionally, conforming users should be isolated from the effects of nonconforming or abusive users.

The generic name given to this balancing act is *closed-loop*, or *adaptive flow control*. In essence, the objective is to control traffic to achieve a throughput close to that of the maximum resource capacity, with very low loss. Such protocols require close cooperation between users and the network. For example, when the network notifies users of congestion in a timely fashion; the user's application reduces its traffic accordingly. On the other hand, when the network has available capacity; users transmit as much as they wish.

This section begins with a discussion of generic flow control methods and how they avoid congestion collapse. We then provide a summary of ATM Generic Flow Control (GFC) and detailed coverage of ATM ABR as examples of closed-loop flow control methods. At the time of writing, MPLS did not have any closed-loop flow control standards work in progress. The flow control methods in IP operate at the transport layer, specifically the TCP adaptive flow control and congestion notification, as described in Chapter 8.

Generic Closed-Loop Flow Control Methods

Figure 22-8 illustrates the generic closed-loop flow control paradigm. If congestion is detected anywhere along the route, including congestion for the outgoing link as shown in the figure, the destination sends a feedback message to the originating node. The originating

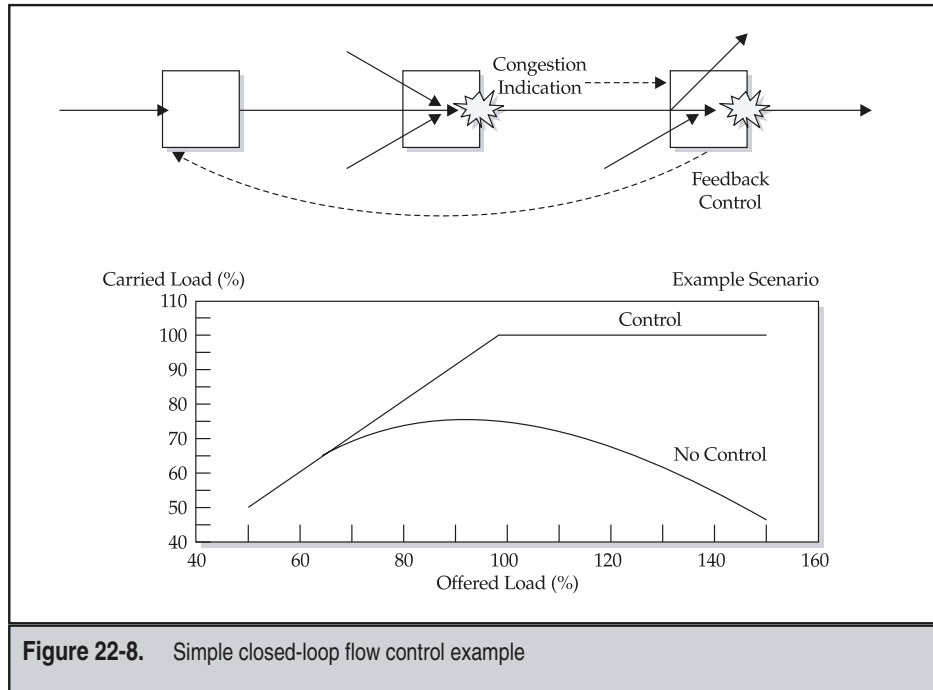


Figure 22-8. Simple closed-loop flow control example

node may service the queue for that connection more slowly, selectively discard packets or cells, or perform a combination of both. In either case, the congestion feedback effectively causes the source and/or network nodes to throttle back the traffic from source(s) responsible for congestion. This action avoids the severely congested state and congestion collapse resulting from an uncontrolled flow, as shown in the graph at the bottom of the figure.

Note that this method works only if the congestion interval is substantially greater than the round-trip delay; otherwise, congestion abates before any feedback control can act. The worst-case scenario for such a feedback scheme would be that of periodic input traffic, with a period approximately equal to the round-trip time. A realistic scenario that can result in long-term overload would be that of major trunk and/or nodal failures in a network during a busy interval. This will likely result in congestion that persists for the duration of the failure, in which case feedback control can be an effective technique for avoiding congestion and splitting the impairment fairly across different sources.

Three generic methods of flow control have been defined and used in the industry: window based, rate based, and credit based. See Reference [McDysan 98] for an in-depth comparison of these methods. The Internet's Transmission Control Protocol (TCP) employs an adaptive window-based flow control. TCP limits the amount of data a source may transmit by defining a dynamically sized transmit window based upon detected loss and time-outs. The ATM Forum's Available Bit Rate (ABR) rate-based flow control dynamically adapts the source transmit rate in response to explicit feedback from the network. A competing ATM protocol, Quantum Flow Control (QFC), uses a credit-based flow control scheme that involves receivers transmitting permission to send (called *credits*) to sources and intermediate nodes. Each of these methods strives to meet the common goal of controlling the flow from the sender to maximize throughput, yet minimize loss, hence avoiding the region of severe network congestion. Each of the methods differs in the way it detects congestion indications as well as the response taken to avoid congestion. We described the TCP window control protocol in Chapter 8, and we describe the ATM Forum ABR approach in this section.

ATM Generic Flow Control (GFC)

The concept of Generic Flow Control (GFC) has a long history in the standardization process. Initially, experts viewed the GFC as a means to implement a function similar to the Distributed Queue Dual Bus (DQDB) protocol on a shared access medium. What was standardized in 1995 in ITU-T Recommendation I.361 [ITU I.361] is a point-to-point configuration that allows a multiplexer to control contention for a shared trunk resource through use of traffic-type selective controls. In other words, GFC empowers the multiplexer to avoid congestion on the shared trunk.

The cell header allocates four bits for Generic Flow Control (GFC) at the ATM UNI. Since the GFC is part of the cell header, a multiplexer requires no additional bandwidth or VPI/VCI allocations to control terminals. The GFC bits have different meanings, depending upon the direction of cell transmission. The four bits of GFC represent almost one percent of the available ATM cell payload rate, and are therefore an important resource. Figure 22-9 illustrates the multiplexer configuration for GFC standardized in I.361 showing

the usage of the GFC bits in each direction between the multiplexer and two types of terminals. The default coding of the GFC is *null*, or all zeros, which indicates that the interface is not under GFC, in what is called an *uncontrolled* mode. ATM terminals have either one or two queues, called connection groups A and B in Figure 22-9. The protocol between the multiplexer and the terminals is asymmetric; the multiplexer controls the terminals, while the terminals only respond or convey information. The multiplexer commands terminals to stop all traffic via the HALT bit, based upon the state of its internal queues. For example, if the shared trunk resource becomes congested, the GFC multiplexer may halt certain terminals. The SET command instructs the terminal to load a credit counter with an initial “Go” value, allowing the terminal to decrement the counter for every cell transmitted. GFC-controlled terminals may send cells only if the credit counter is nonzero. Therefore, as long as the multiplexer periodically sends a SET command to the terminal, the terminal may continue sending data. The NULL meaning of the bit means that the terminal cannot reload the credit counter for that connection group. For example, the GFC multiplexer may SET the credit counter for connection group A but leave the B bit set to zero (i.e., NULL) in the GFC field. The addressed terminal can then send only connection group A traffic.

The terminal responds to commands from the multiplexer, indicating that it understands GFC, which in standards terminology means that it is a *controlled* terminal using the low-order bit as shown in Figure 22-9. The terminal indicates the traffic type for a particular VPI/VCI in the second and third bits, namely, uncontrolled (i.e., both A and B bits are zero), queue A, or queue B. Although standardized, few implementations support GFC. The ATM Forum specifications require only uncontrolled mode (i.e., no GFC) for interoperability. Also note that GFC operates only across an ATM UNI, not between ATM devices.

GFC may also be employed in a unidirectional ring using this protocol. There is also a possibility that further information could be multiplexed into the GFC field for more sophisticated controls.

Available Bit Rate

The ABR specification from the ATM Forum TM 4.1 specification [AF TM 4.1] and ITU-T Recommendations I.371 [ITU I.371] and I.371.1 [ITU I.371.1] share a common goal: to make unused bandwidth available to cooperating end users in a fair, timely manner. Therefore, ABR targets the many existing applications with the ability to reduce their information transfer rate, such as TCP. As such, this objective goes well beyond that of today’s best-effort LANs, where a single selfish, or “highly motivated,” user can paralyze a shared network. If sources conform to the rules of ABR, then the network guarantees a Minimum Cell Rate (MCR) with minimal cell loss. The network may optionally penalize nonconforming sources, or enforce a policy that users not having transmitted recently lose any rights to access bandwidth at rates greater than MCR during periods of congestion.

On the other hand, TCP congestion control implicitly assumes everyone follows the same rules. For example, policing of TCP/IP conformance is limited to a professor’s failing the graduate student who hacks the OS kernel to eliminate TCP’s adaptive flow control

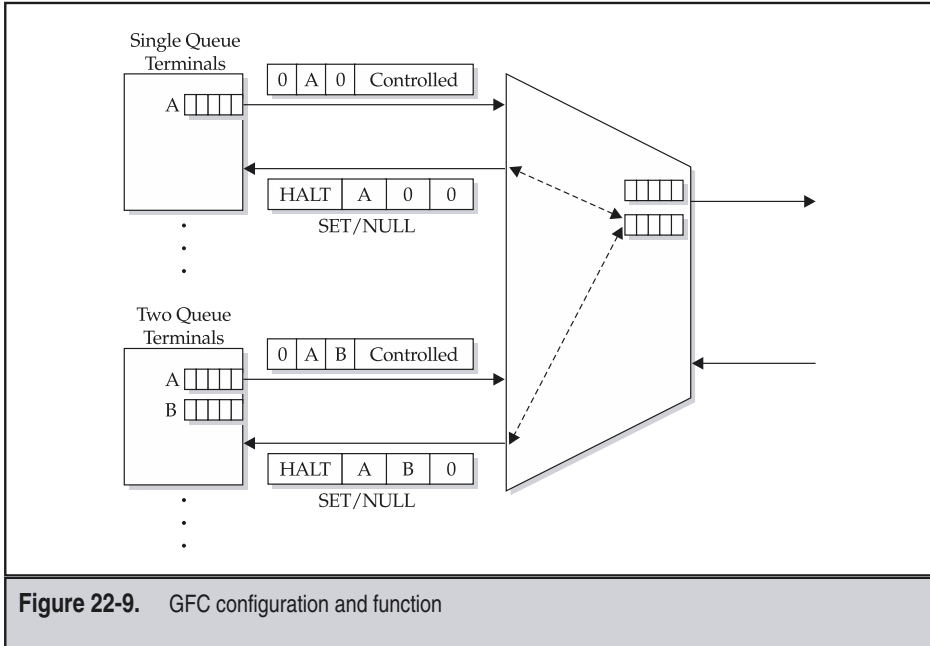


Figure 22-9. GFC configuration and function

and achieves markedly improved throughput at the expense of other TCP users sharing the same IP network. ABR's congestion control overcomes this problem by providing the means for an ATM network switch to police users that do not conform to the mandated source behaviors. Hence, a rogue user who sends at a greater rate than that determined by the ABR protocol receives poorer performance than those ABR users that follow the rules. In the previous TCP scenario, the single rogue user achieves good performance while all the other honest users suffer.

Although TCP sounds more like our daily lives and ABR seems to strive for a lofty, utopian goal, we now describe how ABR achieves fairly arbitrated congestion control. Let's begin by considering some simple analogies from everyday life that involve adaptive closed-loop flow control.

ABR's binary mode is like the green/red lights at the entrance to congested freeways. The light turns green and another car (cell) enters the freeway (interface). Downstream sensors detect congestion and meter cars entering the freeway so that reasonable progress occurs. A police officer tickets vehicles that do not conform to this rule. By analogy, ATM switches indicate congestion using the EFCI bit in the ATM cell header in the forward direction. The destination end system employs a Congestion Indication (CI) field in a Resource Management (RM) cell to communicate the presence of congestion back to the source, which makes green-light/red-light-type decisions regarding the source's current

transmission rate. The recipient of the CI bit does not completely stop sending, but instead reduces its transmission rate by a fraction of the currently available cell rate.

The Explicit Rate (ER) mode of ABR operation adds another degree of complexity. Similar to the manner in which air traffic controllers control the speed of multiple airplanes (cells) converging on a crowded airport, ATM switches along the path of the end-to-end connection communicate an explicit rate for each source. In this way, controller (RM cells) throttle back fast planes (user cells) during periods of congestion to yield a regular arrival pattern at the congested airport (interface).

The Virtual Source/Virtual Destination (VS/VD) mode is analogous to air traffic controllers and airline carriers coordinating the speed and route of aircraft (cells) as they approach airports (interfaces), making connections that maximize the number of seats filled—as well as customer satisfaction. Each airport-to-airport (virtual source destination pair) route makes decisions with an awareness of congestion at other airports (other VS/VD pairs).

Of course, the details of ABR are more complex than these simple analogies; however, keeping these insights in mind may help the reader grasp the essence of ABR as the following treatment proceeds into more detail.

The Great Rate Versus Credit Debate

During the initial stages of the definition of ABR at the ATM Forum in late 1993, two candidate algorithms emerged: a rate-based scheme and a credit-based scheme. The rate-based scheme [Bonomi 95] touted simplicity at the price of reduced efficiency in some cases. Initially, the credit-based scheme [Kung 95] was more complex and required larger buffers, but claimed theoretically ideal efficiency. The debate about which scheme the ATM Forum should follow raged until the fall of 1994, making the popular communications press on a regular basis, until the Forum's technical committee voted in the rate-based scheme by a wide margin. Leading up to the final Traffic Management 4.0 (TM 4.0) specification in the spring of 1996 [AF TM4.0], the rate-based scheme gained considerable complexity in response to credit-based skeptics, but addressed the majority of the efficiency drawbacks of earlier, simpler, rate-based proposals. Some members from the credit-based camp went away and started the Quantum Flow Control (QFC) consortium, which no longer exists; but version 2.0 was archived at the Cell Relay Retreat at the University of Indiana at the time of this writing.

In general, the credit flow control algorithm dedicates buffer space proportional to the link delay and maximum VC bandwidth product, in which case throughput approaching the theoretical maximum is readily achieved. Furthermore, credit control isolates all virtual connections from each other and causes congestion to back up through the network, on a per-virtual-connection basis. Since basic credit-based flow control dedicates buffers to each VC, other VCs don't experience any impact from congestion. Disadvantages of this method are the complexity in the switch and traffic source for implementing the credit control logic, the relatively large amount of storage required for many VCs with long round-trip delays, and the usage of approximately 10 percent of the link bandwidth for per-VC credit messages.

A key criticism of the credit-based approach was the requirement to dedicate buffer capacity to each connection proportional to the round-trip delay. Hence, for a LAN environment, many agreed that the credit-based approach was optimal, and simpler than the rate-based approach. However, in the quest for a single approach that scaled across both LAN and WAN environments, the ATM Forum chose a rate-based scheme where the source generated Resource Management (RM) cells periodically, which the destination, in turn, looped back. The destination, or switches in the return path, either indicated detected congestion or explicitly signaled the maximum source rate they could currently support in these looped-back RM cells. Closing the flow control loop, the ATM Forum TM 4.0 and TM 4.1 specifications detailed the behaviors conforming sources must follow. Despite earlier proposals to specify network enforcement, the final specification left network policing of ABR sources as an implementation option. Informative Appendix III of TM 4.1 describes a Dynamic GCRA as a possible conformance-checking rule for use at the network edge to police ABR sources. As a consequence of these decisions, the VS/VD became the de facto means for networks to dynamically shape user traffic to ensure conformance to the specified traffic contract.

Three Flavors of ABR

ABR specifies a rate-based, closed-loop, flow control mechanism. A key objective of ABR is to fairly distribute the unused, or available, bandwidth to subscribing users while simultaneously achieving a low cell loss rate for all conforming ABR connections. All user data cells on ABR connections must have the CLP bit set to zero. The bandwidth allocated by the network to an ABR connection ranges between the Minimum Cell Rate (MCR) negotiated at connection establishment time and the Peak Cell Rate (PCR) depending upon network congestion and policy.

Users must conform to feedback provided via RM cells according to rules detailed in the ATM Forum's TM 4.1 specification [AF TM 4.1]. ABR flow control occurs between a sending end system, called a *source*, and a receiving end system, called the *destination*, connected via a bidirectional, point-to-point connection. Each of the terminals is both a source and a destination for each direction of an ABR connection. ABR specifies an information flow from the source to the destination composed of two RM flows, one in the forward direction and one in the backward direction, that make up a closed flow control loop. The forward direction is the flow from the source to the destination, and the backward direction is the flow from the destination to the source. In the following sections, we describe the information flow from the source to the destination and its associated RM flows for a single direction of an ABR connection; the procedures in the opposite directional are symmetrical, but may use different parameter values.

Binary Mode ABR

The binary mode, shown in Figure 22-10, involves ATM switching nodes setting EFCI in the forward direction so that the destination end station can set the CI field in a returned RM cell to control the flow of the sending end station. The binary mode ensures interoperability with older ATM switches that can set the EFCI bit in the forward direction

only in response to congestion. This is the simplest mode; however, it experiences higher loss rates in certain situations, such as those where congestion occurs at multiple points in the network. This occurs because EFCI can only indicate the presence or absence of congestion, while other modes can communicate a level of congestion. A flow control reaction that is too slow causes increased loss, while one that is too rapid causes decreased throughput. Furthermore, unless the network elements perform per-connection queuing, unfairness may result when sharing a single buffer.

The complexity in the end system rate control procedures compensates for the simple operation in the network in binary mode. An end system must tune over a dozen parameters to achieve good performance in this mode. The ATM Forum specification also defines a relative rate marking scheme using the Congestion Indication (CI) and No Increase (NI) bits in the RM cell to provide more granular adjustments of the source system's transmission rate.

Explicit Rate (ER) ABR

Figure 22-11 illustrates the ER mode where each Network Element (NE) explicitly sets the maximum allowed rate in RM cells looped back by the destination as they progress backward along the path to the source. The ATM Forum TM 4.1 specification gives examples of how switches may set the explicit rate in the feedback path in Informative Appendix I, leaving the implementation as a vendor-specific decision. Therefore, a key issue that a network ABR switch must address is how it to set the explicit rates to control the sources. The goal is that each user receives a fair allocation of available bandwidth and buffer resources in proportion to their traffic contracts in a responsive manner. Simultaneously, the ABR service should operate at high utilization with negligible loss. This mode requires tuning of far fewer parameters than the binary mode, since an explicit indication of the allowable rate is returned, and hence it is the preferred method in networks that are capable of supporting explicit rate ABR.

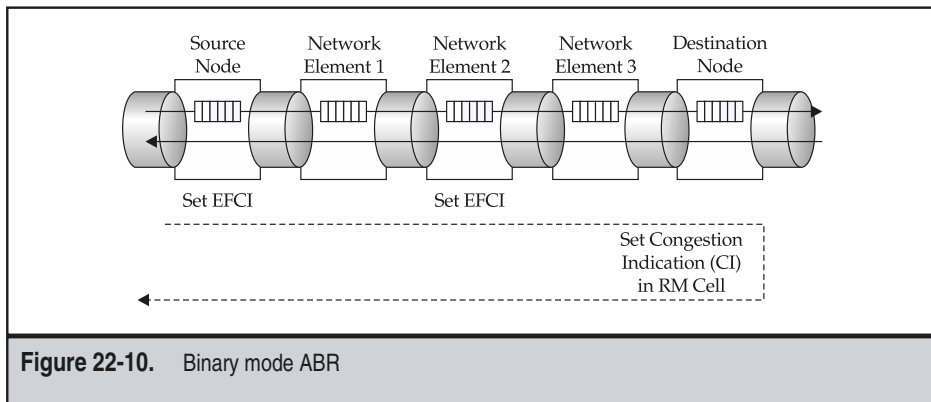
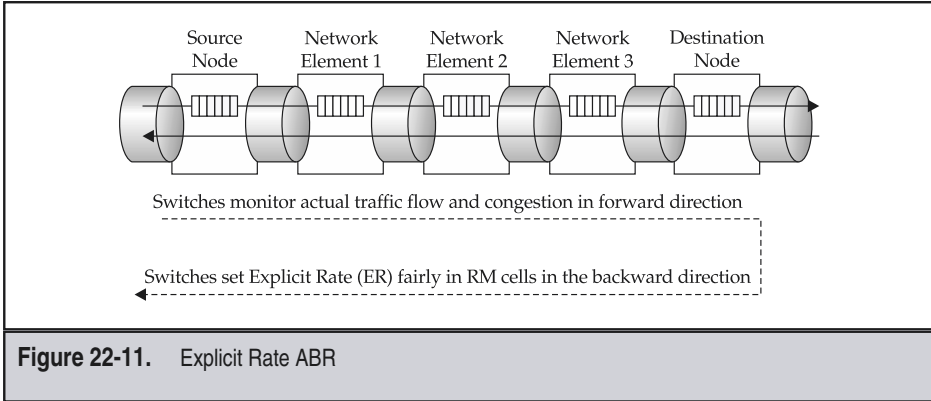


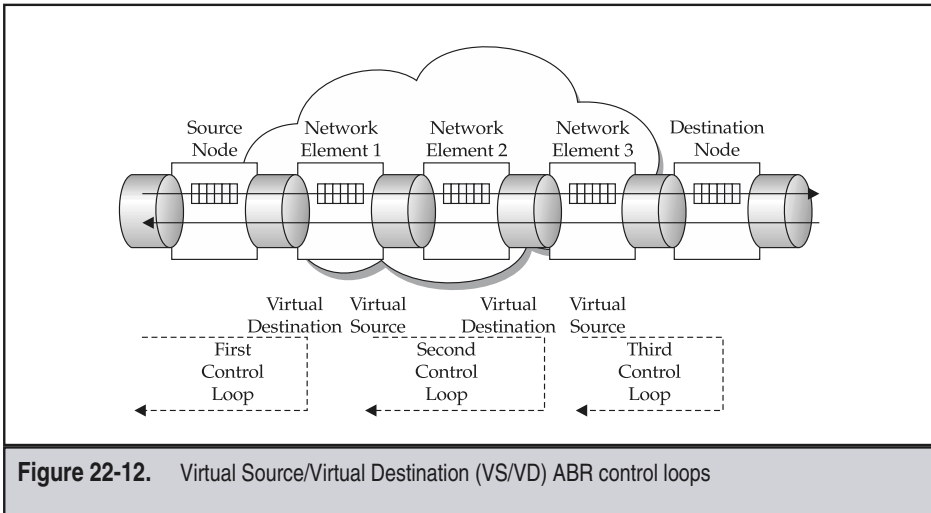
Figure 22-10. Binary mode ABR



Virtual Source/Virtual Destination (VS/VD) ABR

Figure 22-12 illustrates an ABR virtual connection that incorporates segmentation using the concept of mated pairs of virtual sources and destinations. These sources and destinations form closed flow control loops across the sequence of network elements involved in an ATM ABR connection, as shown in the figure.

A configuration where every switch on the path is a virtual source/destination is also called *hop-by-hop* flow control. The example in the figure shows the network elements using VS/VD mode to isolate the source and destination node control loops. The VS/VD



scheme also provides a means for one network to isolate itself from nonconforming ABR behavior occurring in another network. On any network element, the virtual destination terminates the ABR control network for its corresponding source. The same device then originates the traffic on a virtual source for the next control loop that conforms to the end-to-end ABR traffic contract.

ABR Parameters and Resource Management Cells

Upon connection establishment, ABR sources request and/or negotiate the operating parameters shown in Table 22-2. Where applicable, the table also gives the parameter's default value. The user negotiates these parameters with the network via information elements in signaling messages for SVCs, or network management interfaces for PVCs. The following narrative describes each of these parameters and the context in which ABR uses them.

Acronym	Meaning	Default Value
PCR	Source's Peak Cell Rate policed by the network	-
MCR	Minimum Cell Rate guaranteed by the network	0
ACR	Currently Allowed Cell Rate by the network	-
ICR	Initial Cell Rate used by source prior to feedback or after an idle period	PCR
TCR	Tagged Cell Rate for "out-of-rate" RM cells	10 cps
Nrm	Number of cells between forward RM cells	32
Mrm	Control on number of RM cells between forward and backward directions	2
Trm	Upper bound on interval between forward RM cells	100 ms
RIF	Rate Increase Factor used in binary mode	1
RDF	Rate Decrease Factor used in binary mode	1/32,768
ADTF	ACR Decrease Time Factor	0.5 s
TBE	Transient Buffer Exposure	16,777,215
CRM	Count of Missing RM cells	TBE/Nrm
CDF	Cutoff Decrease Factor	1/16
FRTT	Fixed Round Trip Time	-

Table 22-2. ABR Service Parameters

The Allowed Cell Rate (ACR) varies between the minimum and peak rates (i.e., MCR and PCR) negotiated for the connection. The Initial Cell Rate (ICR) applies either to the case where a source transmits for the very first time, or to the case where the source begins transmitting after a long idle period. The ABR specification defines a means to calculate ICR based upon the TBE, Nrm, and FRTT parameters. The other parameters control the source, destination, and network behavior in conjunction with the RM cell contents, as described in the text that follows.

In order for the ABR algorithms to operate responsively, feedback must occur. Figure 22-13 shows how RM cells sent periodically by the source probe the forward path, while the destination assumes the responsibility for turning around these RM cells by changing the DIRECTION bit in the RM cell and sending the RM cell in the opposite direction.

The ABR specification allows the destination to return fewer RM cells than it receives. This design handles the case where the MCR of the backward connection cannot sustain feedback at a rate corresponding to one RM cell for each of Nrm cells received from the source in the forward direction. This occurs in practice in ATM over ADSL applications where the bandwidth differs by an order of magnitude in the forward and backward directions. *In-rate* RM cells count toward the Allowed Cell Rate (ACR) of an ABR connection in each direction. A source must send an RM cell at least once every Trm milliseconds. If the source does not receive a backward RM cell within CRM cell times, then it reduces ACR by the CDF factor, unless this would cause ACR to become less than MCR. End systems may optionally send *out-of-rate* RM cells at a lower priority by setting CLP = 1 at a rate limited by the TCR parameter to make the ABR algorithm more responsive. However, since these cells have a lower priority, the network may discard them during periods of congestion.

Table 22-3 depicts the content of the ATM Forum 4.1 ABR Resource Management (RM) cell. The fields are aligned with the ITU-T ABR specification in ITU-T Recommendation I.371. We briefly summarize how the various ABR operating modes specify source and network behaviors using these fields in RM cells. See Reference [Jain 96] for more information on source behaviors.

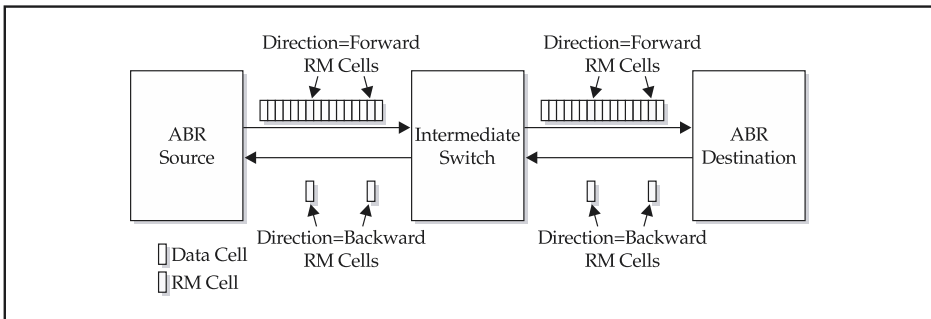


Figure 22-13. ABR RM cells insertion and feedback

Field	Description	Size
Header	ATM RM Cell: VPC VCI = 6, PTI = 110, VCC: PTI = 110	5 bytes
ID	Protocol Identifier = '0000001' Message Type Field	1 byte
DIR	Direction: Forward = 0 and Backward = 1	1 bit
BECN	Backward Explicit Congestion Notification	1 bit
CI	Congestion Indication	1 bit
NI	No Increase	1 bit
RA*	Request/Acknowledge	1 bit
Reserved	Reserved for future use	3 bytes
ER	Explicit Rate	2 bytes
CCR	Current Cell Rate	2 bytes
MCR	Minimum Cell Rate	2 bytes
QL*	Queue Length	4 bytes
SN*	Sequence Number	4 bytes
Reserved	Reserved for future use	246 bits
CRC-10	Cyclic Redundancy Check	10 bits

* Fields defined in ITU-T I.371, but not used in ATM Forum TM 4.1

Table 22-3. ATM Forum Resource Management (RM) Cell Contents

The destination or the network may optionally set the Backward Explicit Congestion Notification (BECN) field in RM cells in the backward direction in order to command the source to reduce its rate immediately. The source always sets CI = 0 in forward RM cells. Either the destination or an intermediate switch sets the Congestion Indication (CI = 1) bit to cause the source to decrease its ACR. When operating in binary mode, sources utilize the Rate Increase and Decrease Factors (RIF and RDF) to calculate a new ACR.

The destination or an intermediate switch sets the No Increase (NI) bit to prevent a source from increasing its ACR. The intent of the NI bit is to address detection of impending congestion in networks with long delay paths. Sources set NI = 0 in the forward direction, and switches cannot change NI from 1 to 0 in the backward direction.

TM 4.1 encodes all rates (i.e., ACR, CCR, and MCR) using a 16-bit floating point format that represents values of over 4 trillion cells per second. A practical limitation is the 24-bit representation of rate values utilized in the signaling messages, which limits the maximum rates to approximately 16 million cells per second, or approximately 7 Gbps.

In the ER and VS/VD modes, the source sets the Current Cell Rate (CCR) field to its current ACR in forward RM cells. The destination and intermediate switches may use the CCR field to calculate ER in the backward direction, but they cannot modify it. The destination and/or any intermediate switch uses the Explicit Rate (ER) field to set the source's Allowed Cell Rate (ACR) to a specific value. Typically, the destination initially sets ACR to the maximum possible rate (i.e., PCR). Subsequently, any network element in the backward path may reduce the ER to communicate the supportable fair share for the connection. According to the rules of ABR, no network element in the backward path can increase ER.

The source sets the Minimum Cell Rate to the negotiated value for the connection. The destination and intermediate switches may use the MCR field to calculate ER. If the MCR value in the RM cell differs from the one signaled, then a switch may correct it.

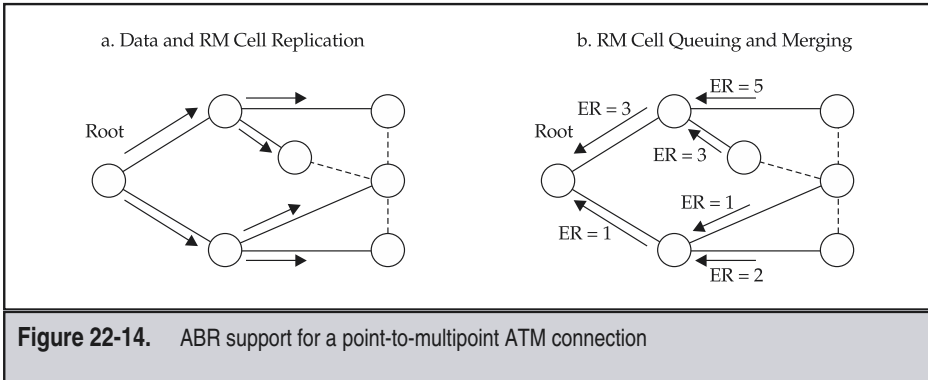
ABR Conformance Checking: Dynamic GCRA

Informative Appendix III of the ATM Forum's TM 4.1 specification defines a technique called *dynamic* GCRA that meets the requirements for checking conformance of an ATM cell flow against the ABR traffic contract in terms of a sequence of Explicit Rate (ER) fields transmitted by the network within RM cells in the backward direction. The principal difference between the dynamic GCRA and the static GCRA is that the Increment (I) may change over time. A network may use dynamic GCRA to police ABR users to ensure fairness and conformance to the ABR traffic contract.

Point-to-Multipoint ABR

As described in Chapters 18 and 19, ATM makes extensive use of point-to-multipoint connections in LAN Emulation (LANE), as well as the protocol supporting IP multicast over ATM defined in RFC 2022. Therefore, the ATM Forum described a framework in support of an optional capability to provide ABR service over point-to-multipoint connections. Recall that an ATM point-to-multipoint connection emanates from a root node and branches at a number of intermediate nodes to each of the leaves, as illustrated in Figure 22-14.

As required in point-to-multipoint connections, the network must return cells from each of the leaves back to the root. The branch nodes implementing ABR in a point-to-multipoint connection perform two important functions as illustrated in the figure: replicating data and RM cells in the direction from the root to the leaves, and merging RM cells in the direction from the leaves back to the root. Figure 22-14a illustrates the replication of both data and RM cells from a root node out through two branching points to leaves in a small network. In the backward direction (i.e., from the leaves to the root), the branching nodes may operate in different modes. In one mode of operation, a branching node queues RM cells and consolidates the feedback—for example, taking the minimum Explicit Rate (ER) value and returning it back toward the root as shown in Figure 22-14b. Note that this mode of operation reduces the throughput of the entire point-to-multipoint connection to that of the slowest receiver, or bottleneck at a branching point. Although this may seem bad, it isn't if the goal is for effectively lossless broadcast communication to all parties in the point-to-multipoint connection. As stated in the TM 4.1 specification, branching nodes may also implement the virtual source and destination closed loop-protocol between themselves.



CONGESTION RECOVERY

ATM and MPLS network devices initiate congestion recovery procedures after entering the severely congested region when all else has failed. These are drastic measures and may impact some traffic more than others. This section covers the following methods of congestion recovery:

- ▼ Selective discard
- ATM Early/Partial Packet Discard (EPD/PPD)
- Dynamic UPC
- Disconnection and/or rerouting
- ▲ Operations procedures

Selective Discard

As discussed in Chapter 21, selective discard of cells or packets is an effective means to provide a higher loss rate to packets or cells at a lower loss priority. ATM standards define selective cell discard as the mechanism where the network may discard CLP = 1 cell flow while meeting Quality of Service (QoS) on both the CLP = 0 and CLP = 1 flows. Recall that the Cell Loss Priority (CLP) bit in the ATM cell header indicates whether a cell is of high priority (CLP = 0) or low priority (CLP = 1). Selective discard of cells gives preferential treatment to CLP = 0 cells over CLP = 1 cells during periods of congestion. In an analogous manner, the IP Diffserv Assured Forwarding (AF) standard defines three levels of drop precedence.

Selective cell discard is an important standardized ATM function for recovering from severe congestion. An ATM network may use selective discard to ensure that the CLP = 0 cell flow receives a guaranteed QoS. If the network is not congested, then the application may achieve higher throughput by also transferring CLP = 1 cells, but never less than the

requested amount for the CLP = 0 flow. Of course, the ATM network policer must implement tagging using the CLP bit, or the user may also tag cells as CLP = 1 if it considers them to be of a lower priority. However, user-tagged cells create an ambiguity because intermediate network nodes have no way to discern whether the user set the CLP bit, or the network's UPC set it as a result of tagging nonconforming cells. If the user sets the CLP bit, and the network does tagging, then it may not be possible to guarantee a cell-loss ratio for the CLP = 1 cell flow. In practice, this isn't much of a problem because few applications utilize the CLP bit.

Selective packet discard in Diffserv AF based upon drop preference is an analogous function in IP and MPLS networks. The application may set the drop preference field, or a policer may set the drop precedence bits based upon conformance to traffic parameters, as described in Chapter 20.

Figure 22-15 gives an example of how hierarchical video coding makes use of selective discard congestion recovery. Hierarchical video coding (for example, MPEG-2) encodes the critical information required to construct the major parts of the video image sequence as the higher priority CLP = 0 marked cells, while encoding the remaining, detailed minor change information as a separate stream of lower-priority CLP = 1 marked cells. Thus, when there is a scene change, the video coder generates CLP = 0 cells at an increased rate for a period of time, as indicated by the slowly varying solid line in the figure. The video application sends the detail and minor adjustments within the scene as CLP = 1 cells as shown by the jagged line in the figure. When a switch multiplexes several such video sources together and utilizes selective cell discard, as shown in Figure 22-15, this congestion recovery scheme ensures that the critical scene change information gets through, even if some of the detail is momentarily lost during transient intervals of congestion.

Early/Partial Packet Discard (EPD/PPD)

The ATM Forum TM 4.1 specification also specifies an intelligent frame discard function as an optional congestion recovery procedure. For AAL5, a number of studies and tests show that a more-effective reaction to congestion is to discard at the frame level rather than at the cell level. The situation where intelligent frame discard helps the most occurs when many sources congest a particular resource, such as an output queue on an ATM switch serving a heavily utilized link. Note that a network element that discards at the cell level may discard portions of many packets. Usually, the objective of intelligent frame discard is to maximize the number of complete packets transferred. However, other objectives, like maximizing the number of bytes transferred (in complete large frames), are also possible. Intelligent frame-level discard helps networks recover from the phenomenon of congestion collapse during periods of severe overload.

An ATM device may treat user data as frames only if the user indicates so by using the broadband ATM traffic descriptor IE in an SVC message, or by setting the value in the ILMI MIB for a PVC at subscription time. Once the user negotiates frame-level discard service with the network using these means, the ATM switches use the Payload Type Indicator (PTI) in the ATM cells to detect the last cell of an AAL5 PDU, as detailed in Chapter 12. The commonly used industry terminology for the intelligent frame discard capability is the following two actions:

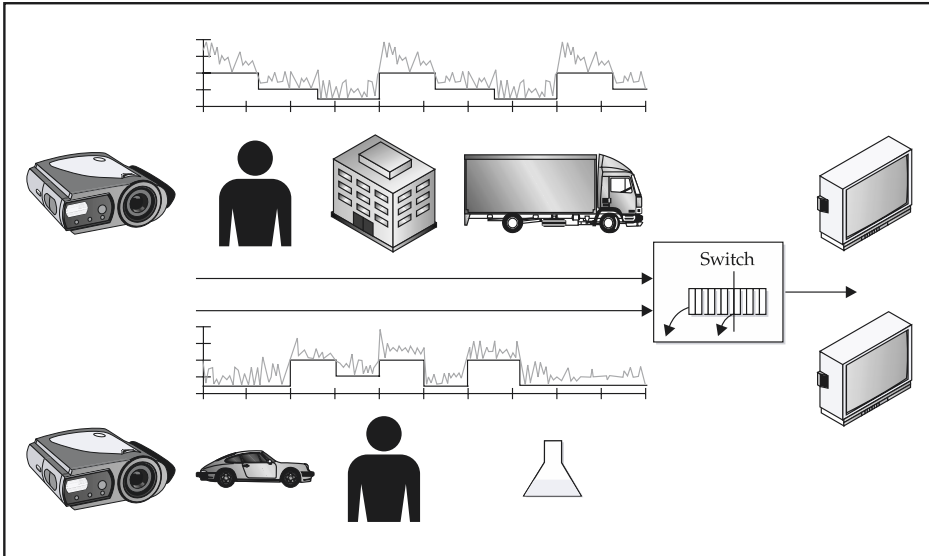
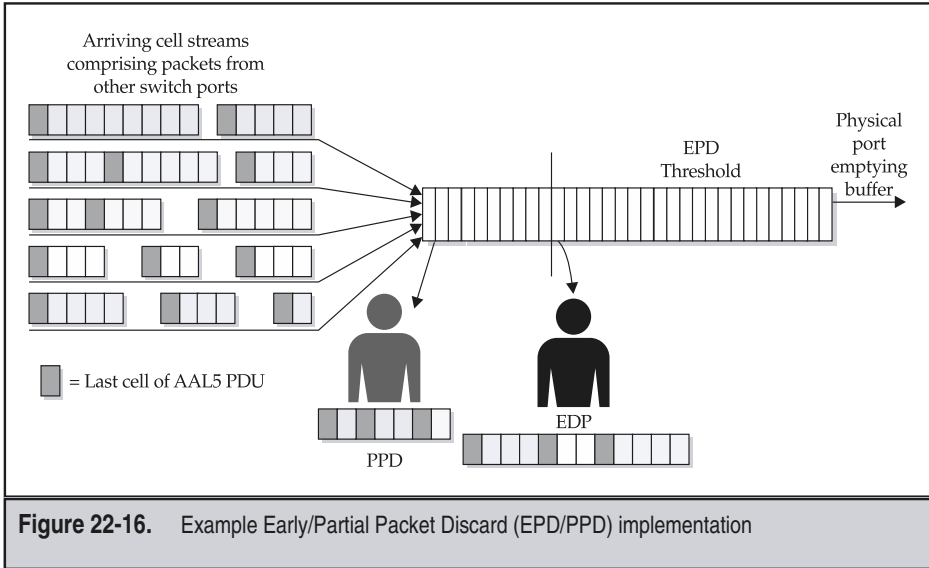


Figure 22-15. Hierarchical video coding

Early Packet Discard (EPD) occurs when a device in a congested state discards every cell from an AAL5 PDU. EPD prevents cells from entering the buffer, reserving remaining buffer capacity for the cells from packets already admitted to the buffer.

Partial Packet Discard (PPD) occurs when a device discards all remaining cells, when it discards a cell in the middle (i.e., not the first or last cell) of an AAL5 packet. PPD acts when some cells from a packet have already been admitted to the buffer. Some EPD/PPD implementations don't discard the last cell to ensure a SAR failure and maximize the chances of reassembling the next frame on the affected Virtual Channel Connection (VCC)

Figure 22-16 shows a representative implementation of EPD/PPD using a single shared buffer. Cells from multiple frame-level sources arrive at the switch buffer from other ports on the switch, as indicated on the left-hand side of the figure. Once the buffer level exceeds the EPD threshold, the EPD gremlin selectively discards cells from entire frames. Once the buffer reaches capacity—or some other discard action occurs, such as selective cell discard dropping a $CLP = 1$ cell within a frame, the PPD gremlin takes over, discarding the remaining cells of the frame. PPD improves overall throughput because, if the network loses even one cell from a frame, the end user loses the entire frame. For example, a maximum size Ethernet data frame has 30 cells. When applied in the context where many sources contend for a common resource, EPD/PPD can improve usable throughput significantly.



Dynamic Usage Parameter Control (UPC)

Another way to recover from congestion is to dynamically reconfigure the UPC parameters. This could be done by renegotiation with the user, or unilaterally by the network for certain types of connections. This technique is related to the dynamic GCRA defined for ABR conformance checking.

Disconnection and/or Rerouting

Another rather drastic response that provides recovery from congestion is to disconnect some connections if and when severe congestion persists. For example, some connections may be preemptible. The U.S. government levies such a requirement upon carriers to support national defense traffic or local community emergency services at the highest priority, such that all other traffic is subject to disconnection if these priority connections require capacity. Another important practical case is disconnection followed by rerouting. This can provide for recovery if existing connections are suboptimally routed, since a disconnection followed by another connection attempt achieves rerouting and may restore failed connections. There are many creative ideas in this area; another standard example is the ATM signaling congestion control [AF SCC 1.0] that attempts to prevent increasing congestion at nodes that are approaching a congested condition (in terms of call processor load), by redirecting the signaling messages requesting new connections, or attempting to reestablish ones, around such nodes.

Operational Procedures

If all of the automatic methods fail, then human operators can intervene and manually disconnect certain connections, reroute traffic, or patch in additional resources. Network management actions for controlling the automated reroutes are not standardized, and are therefore a proprietary network implementation. These procedures must be carefully coordinated, especially if multiple networks are involved.

REVIEW

This chapter defined congestion as demand in excess of resource capacity. The degree of congestion impacts contention for resources, which can reduce throughput and increase delay, as occurs in vehicular traffic jams. Congestion occurs at multiple levels in time and space. In time, congestion occurs at the cell level, the burst (or packet) level, or the connection level. In space, congestion occurs at a single node, multiple nodes, or across networks. This chapter categorized congestion control schemes in terms of time scale and their general philosophy. First, in the network planning time scale of weeks to months, congestion management attempts to ensure that congestion never occurs, which may be done at the expense of reduced efficiency. Next, acting in the region in the time scale of bursts, congestion avoidance schemes attempt to operate on the verge of mild congestion to achieve higher utilization at nearly optimal performance. An important class of techniques here is that of closed-loop flow control. As an example of this approach, we described the ATM ABR service. Finally, the chapter described congestion recovery techniques that move the network out of a severely congested state in the event that the previous two philosophies fail, sometimes using rather drastic measures such as selective discard, or even disconnection of some users.



PART VI



Communications Engineering, Traffic Engineering, and Design Considerations

This part provides you with an application-oriented view of the communications engineering, traffic engineering, and design considerations applied to ATM. First, Chapter 23 defines and applies communications and information theory terminology. This includes the concepts of signal structure, frequency passband, noisy communications channels, bit errors, channel capacity, and error-correcting codes. The

text applies the theory to determine the undetected error rates of ATM HEC and AAL5. Next, Chapter 24 covers the important topic of traffic engineering. The treatment begins by introducing random processes and basic models from queuing theory. The analysis applies these results to ATM switch design trade-offs, performance of CBR and VBR traffic types, equivalent capacity, statistical multiplexing, and priority queuing. Finally, Chapter 25 discusses additional design considerations involved in ATM networks. These include a look at the impacts of delay, loss, and delay variation on applications. The text also describes the performance of TCP over various types of ATM services, focusing on the impacts of buffer size, packet discard techniques, and congestion scenarios. This chapter also analyzes the statistical multiplex gain achieved by packetized voice and the savings achieved by integrating voice and data. Throughout this part, we apply theoretical approaches to the real-world business problems defined in the previous sections. A spreadsheet can implement most formulas presented in this part.

CHAPTER 23

Basic Communications Engineering

This chapter deals with some useful approximations for modeling and estimating ATM network performance. The text introduces the concepts of probability theory, followed by an overview of digital signals and their frequency spectra. Next, we analyze the effects of errors on the achievable capacity over a given communications channel. The chapter concludes with an overview of error-correcting codes and data compression. In particular, the text gives a simple method for evaluating the probability of undetected error when using the ubiquitous cyclical redundancy check (CRC) technique. All formulas used in these examples are simple enough for spreadsheet computation so that the reader can readily evaluate a specific network situation. The chapter cites references to treatments that are more detailed and extend these concepts for the interested reader.

PHILOSOPHY

This section discusses several dimensions of communications engineering philosophy. First, we cover the basic notion of a communications channel and how factors such as noise, frequency pass band, and errors impact reliable communications. Next, the text touches on the different roles for deterministic and random models in communication links. Finally, the section concludes with some practical approaches to basic communications engineering.

Communications Channel Model

Figure 23-1 illustrates the fundamental elements of a typical communications system [Gallagher68], [Proakis83]. Starting from the left-hand side, a source generates analog or digital information. Next, a source encoder transforms this information into a series of binary digits. A source encoder may also perform data compression to decrease the required number of binary digits. The source encoder may also add other overhead at

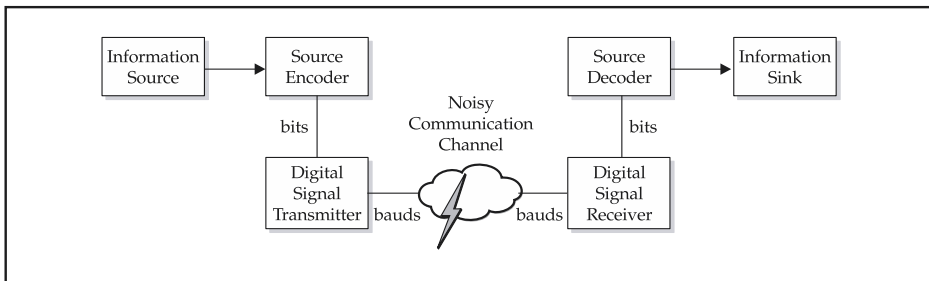


Figure 23-1. Typical communications channel model

protocol layers below the application layer. The digital signal transmitter may add error detection and/or error correction prior to sending modulated digital waveforms onto a noisy communications channel. The transmitter may encode multiple bits as a single baud, or channel symbol. A digital signal receiver takes waveforms received over the channel, which now contain the original signal contaminated by noise, and attempts to re-create the same sequence of binary digits sent by the transmitter, for example, using demodulation and error correction coding. The source decoder takes this received stream of binary digits from the receiver and applies the higher-layer protocols prior to passing the information on to the information sink on the far right-hand side. For example, the source decoder may perform data decompression.

Deterministic Versus Random Modeling

The discipline of communications engineering utilizes both deterministic and random models depending upon the situation. Deterministic models describe signal structures and their frequency spectra. Random models approximate impairments on real-world communications channels, as well as information source behavior, as covered in the next chapter. Although the actual phenomena that cause errors on real channels are very complex, often a simplifying random model allows an engineer to design applications for the noisy channel. Of course, the available frequency spectrum and signal-to-noise ratio place a fundamental limit on the rate of communication, known as Shannon's channel capacity, as described later in this chapter.

PROBABILITY THEORY

This chapter introduces some basic probability theory, with parameters chosen to model some reasonable communications channels encountered in ATM and MPLS networking.

Randomness in Communications Networks

Probability theory is used in communications engineering in two principal areas: modeling source behaviors and modeling the effects of noisy communications channels. Central to the theory is the notion of sets, or groupings of experimental outcomes. A familiar experiment with two equally likely outcomes is a fair coin toss: it comes up either heads or tails. A particular type of binary communications channel examined in more detail later exhibits a similar dual outcome: either the received bit is received correctly, or it is in error. The term *bit error rate (BER)* refers to the probability that the bit is received in error. The same random model also applies to a source generating random data.

Often, however, the outcome of one trial is not independent of preceding trials. Such is the case for communications channels that experience bursts of errors, for example, twisted copper pairs, protection switching in fiber optic systems, and many radio channels. Information sources also tend to have bursty behavior, a subject that the discussion on equivalent bandwidth addresses in the next chapter.

Random Trials and Bernoulli Processes

A Bernoulli process is the result of N independent “coin flips” in an experiment where the probabilities of heads and tails are unequal: p being the probability that “heads” occurs and $(1 - p)$ being the probability that “tails” occurs as the result of each coin flip. The probability that k heads occur, and hence $(N - k)$ tails also occur, as a result of N repeated Bernoulli trials (“coin flips”) is called the *binomial distribution* as given by

$$\Pr[k \text{ "heads" in } N \text{ "flips"}] = b(N, k, p) = \binom{N}{k} p^k (1 - p)^{N - k}$$

$$\text{where } \binom{N}{k} \equiv \frac{N!}{(N - k)!k!}$$

Microsoft Excel implements this formula for $b(N, k, p)$ in the BINOMDIST(k, N, p, FALSE) function. The mean, or average, of the binomial distribution is Np , and the variance is $Np(1 - p)$.

The Normal/Gaussian Distribution

A consequence of the deMoivre-Laplace and Central Limit theorems states that the Gaussian, or Normal, distribution is a good approximation to the binomial distribution when Np is a large number in the $Np(1 - p)$ region about the mean [Papoulis 91]. Figure 23-2 compares the binomial and Gaussian distributions for an example where $N = 100$ and $p = 0.1$ to illustrate this point. The distributions have basically the same shape; and for large values

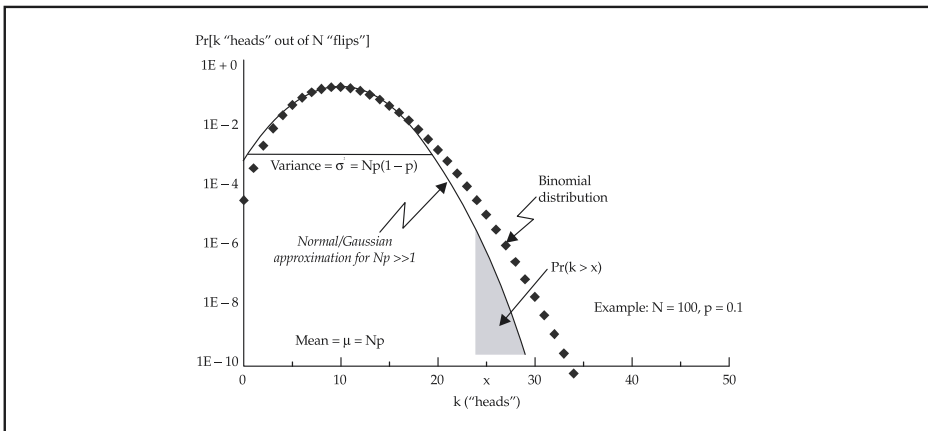


Figure 23-2. Normal approximation to binomial distribution

of Np , in the $Np(1 - p)$ region about Np , the Gaussian distribution is a good approximation to the binomial distribution.

This is helpful in analyzing relative performance in that the probability area under the tail of the Gaussian, or Normal, distribution is widely tabulated and implemented in many spreadsheets and mathematical programming systems. The cumulative distribution of the normal density, defined as $Q(\alpha)$ in the following formulae, is a good approximation to the sum of the tail of the binomial distribution.

$$\text{Prob}[k > x] \approx Q\left(\frac{x - \mu}{\sigma}\right) = Q(\alpha) \approx \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2} dx$$

$$\text{where } Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2} dx$$

This book uses this approximation in several contexts to estimate loss probability, statistical multiplex gain, and delay variation in subsequent chapters. Microsoft Excel implements the commonly used function $Q((x - \mu) / \sigma)$ as $1 - \text{NORMDIST}(x, \mu, \sigma, \text{TRUE})$ for a normal random variable with mean μ and standard deviation σ . Note that from the preceding approximation for $Q(\alpha)$, a closed-form expression for $\epsilon \approx \text{Pr}[k > x]$ is

$$\alpha \approx \sqrt{-2 \ln(\epsilon) - 2 \ln(2\pi)}$$

Note that Microsoft Excel implements the function to determine α more precisely as $\alpha = -\text{NORMINV}(\epsilon, 0, 1)$. The preceding approximation overestimates the required capacity; therefore, use the accurate formula if your software provides it.

As discussed in the next section, the normal distribution is also an excellent model for noise generated on electrical and optical communications channels. The same probability density and distribution functions just defined are widely used by communications engineers to estimate the error rate of the physical layer connecting switching nodes. The normal distribution is widely used in other branches of engineering, for example, in approximating buffer overflow and statistical multiplexing gain in queuing theory, as studied in the next chapter.

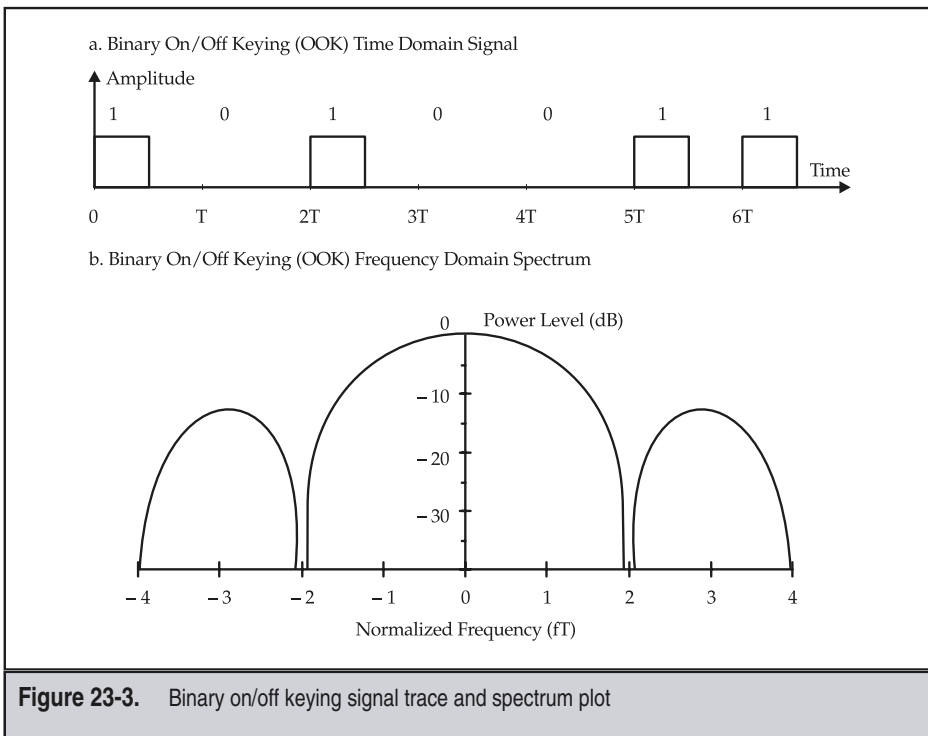
COMMON DIGITAL SIGNALS AND THEIR SPECTRA

This section summarizes key concepts of digital modulation to achieve efficient utilization of the available frequency range for a bandwidth-limited channel. This notion of restricted bandwidth is key to the maximum information rate achievable over a noisy channel. Since frequency limitations and noise are key impairments in many real communications channels, a background in these areas is important in designing physical links in ATM-based networks. Electrical and optical digital communications systems use different types of modulation schemes to physically transfer information from a transmitter to a receiver. The text begins with the simplest scheme—basically, a synchronous telegraph signal—and finishes with the scheme that is the basis of modulation techniques employed by modern high-speed modems.

The Telegraph Pulse: Binary On/Off Keying

The simplest signal is a pulse, similar in nature to the telegraph. The transmitter emits a pulse of electromagnetic energy (e.g., an optical pulse, a flow of current, a change in an electric field, or a radio frequency wave) into the physical medium (e.g., an optical fiber, a pair of wires, or the atmosphere), which the receiver detects. Figure 23-3a shows the time domain representation of a random telegraph, or binary on/off keying (OOK) signal mapping the input sequence of ones and zeros into either a transmitted pulse or the absence of a pulse, respectively, once every T seconds. The information rate of this system is $R = 1/T$ bits per second. The baud rate of this system is also R .

The spectrum is the Fourier transform of the time domain signal, which results in a frequency domain representation of the signal. Communications engineers define frequencies in a measure defined in terms of the number of complete sinusoidal cycles of an electromagnetic wave in a 1-s interval. The unit of Hertz (abbreviated Hz) corresponds to one sinusoidal cycle per second. Common alternating current (AC) power systems in the United States operate at 60 Hz. Although this book frequently uses the term “bandwidth” to refer to the digital bit rate of a transmission link, please note that many commu-



communications engineers use the term “bandwidth” to refer to the *frequency passband* of a particular signal. For random signals, an analogous concept to the frequency spectrum is that of the power spectral density, which is the Fourier transform of the autocorrelation function of the random time domain signal [Papoulis 91].

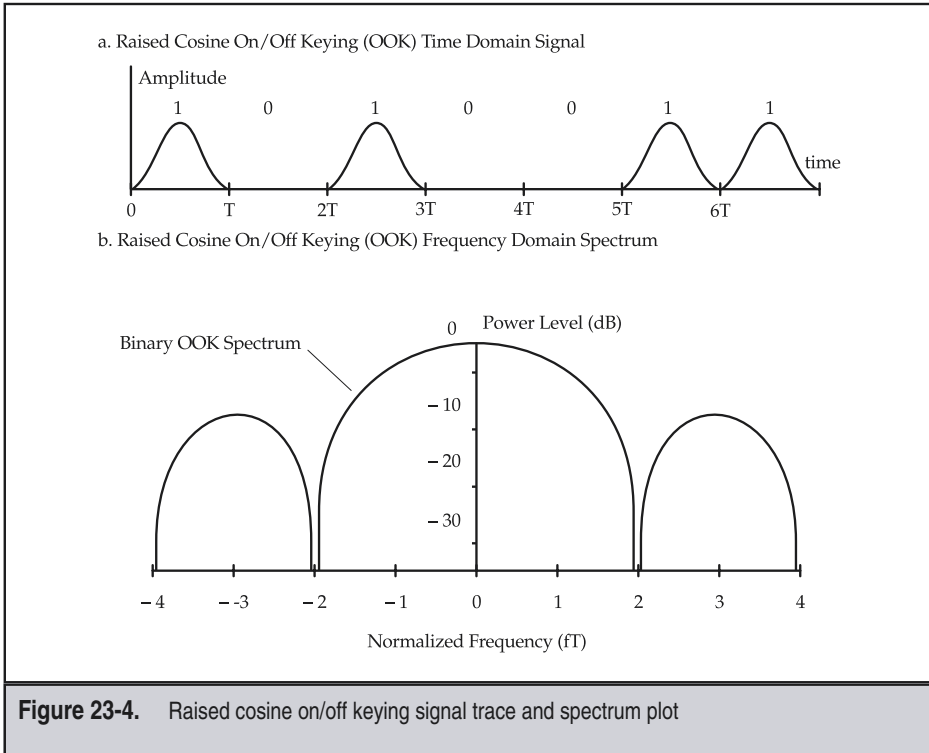
Figure 23-3b illustrates the frequency spectrum for the binary on/off signal, also called a return to zero signal. Binary on/off keying does not make efficient use of the available frequency passband because the main lobe of the spectrum is widely spread and significant power is in the side lobes. For example, at three times the source bit rate, the power is still over 5 percent (−13 dB) below the peak power level. For a particular absolute signal-to-noise ratio, x , the SNR in decibels is computed as $\text{SNR} = 10 \log(x)$. For example, $x = 2$ corresponds to roughly 3 dB, $x = 4$ corresponds to approximately 6 dB, and $x = 10$ corresponds to 10 dB. The frequency *side lobes* from one signal’s spectrum can interfere with signals in adjacent frequency bands. Hence, shrewd designers use more spectrally contained signals in band-limited circumstances. On optical fiber systems, however, available spectrum is not a concern in some networks, since each fiber has over one trillion Hertz of available bandwidth. Furthermore, on/off keying is an inexpensive means to pulse a laser or light-emitting diode (LED) in an optical communications system. Another commonly used modulation is Non Return to Zero (NRZ), whose time domain signal and spectrum are given in the cited references.

A Better Way: Pulse Shaping

As seen in the previous section, an unfortunate result of the telegraph pulse is that it does not utilize the frequency passband very efficiently. As communications engineers strove to develop better ways to transfer information at increasingly higher rates, they invented more spectrally efficient waveforms. As you might expect, making the pulse smoother could improve the spectrum characteristics, and this is indeed what communications engineers discovered. One of the “smoothest” pulses is the raised cosine shape illustrated in Figure 23-4a. This smoother pulse makes a significant difference in the frequency spectrum as shown in Figure 23-4b, shown plotted alongside the binary OOK for comparison purposes. The plot normalizes the power for these two signaling methods. The raised cosine pulse must have approximately a 20 percent higher amplitude than the rectangular pulse in order to convey the same energy. The side lobes are down over 30 dB (i.e., a level of only 0.1 percent) from the main spectral lobe. Furthermore, the side lobe occurs much closer to the main lobe than it does in binary OOK. This means that more channels fit within a band-limited channel without interfering with each other. Electrical and radio systems can readily implement this technique, as can some optical systems.

Pushing the Envelope: Quadrature Amplitude Modulation

If you’re a communications engineer at heart, your thoughts should be racing now, wondering how much better the frequency passband can be utilized. This quest rapidly becomes a very complex and specialized subject. However, we briefly introduce one more example of signal modulation techniques—in particular, the one used in the popular standard V.32 modem signaling at 9.6 Kbps over most telephone networks in the world. To start



with, real electromagnetic physical media actually support two carrier frequencies, called the in-phase and quadrature channels. This fact is due to the mathematics of the complex sinusoids. Specifically, when correlated over a full period (i.e., the integral of their product), the cosine and sine functions cancel each other out. Communications engineers say that these signals are orthogonal, or that they are in quadrature because the sine and cosine functions are 90 degrees out of phase. Additionally, the amplitude of the pulse can also be modulated. Hence, combining the notions of sinusoidal phase and amplitude, the signal space makes up a two-dimensional constellation diagram for quadrature amplitude modulation from the V.32 modem standard for nonredundant 9.6 Kbps coding, as shown in Figure 23-5. Since the constellation uses four values of amplitude and phase independently, the sixteen resulting amplitude and phase combinations represent all possible four-bit patterns, as shown in the figure.

The time domain representation of QAM is straightforward, yet somewhat too involved for our coverage here. See References [Proakis83], [Bellamy82], or [Held 95] for

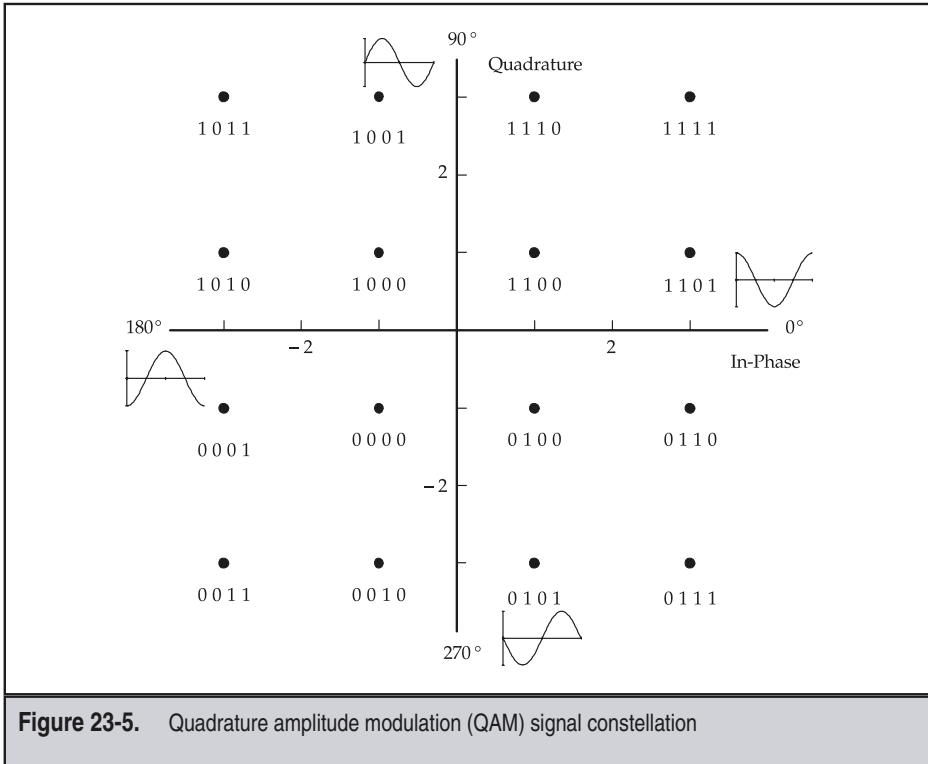


Figure 23-5. Quadrature amplitude modulation (QAM) signal constellation

more details. Figure 23-6 illustrates an example of the time domain characteristics of QAM signals. The top of the figure shows the input binary sequence, which modulates the in-phase and quadrature channels corresponding to the x and y axes on the signal constellation plot of Figure 23-5. Below the in-phase and quadrature input levels is the resulting waveform modulated by a cosine and sine wave, respectively. The trace at the bottom of the figure is the composite sum of the modulated in-phase and quadrature channels. In this case, the information rate is $4/T$ bits per second. Communications engineers call the signal transmitted every T seconds a *baud*, or they refer to the channel symbol rate in units of *bauds per second*. For QAM, the information bit rate is four times the baud rate.

What did this additional processing gain? It improves the spectral efficiency markedly, as shown in Figure 23-7. The figure shows the spectrum for on/off keying for comparison purposes. Note that the spectral width for the main lobe of QAM is half that of OOK. Furthermore, the system transmits four times as much information per channel symbol time (T); that is, the information rate of the channel is $R = 4/T$ bits per second.

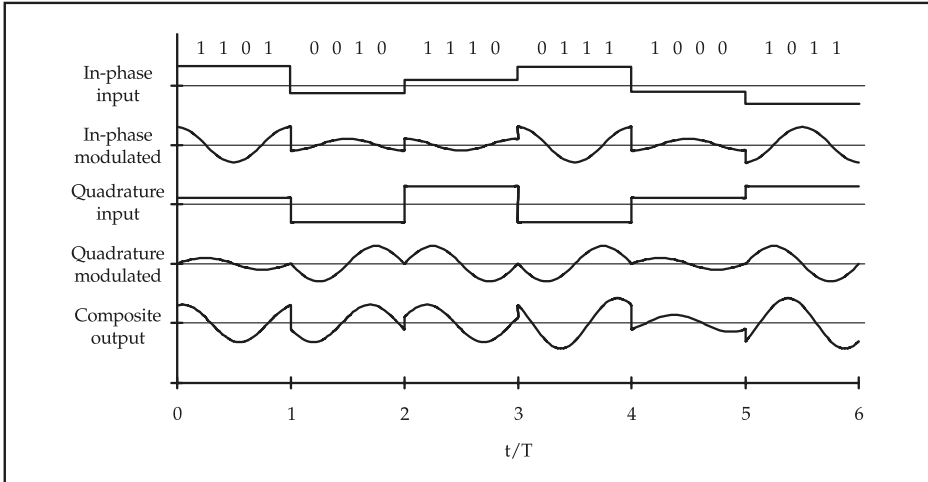


Figure 23-6. QAM in-phase, quadrature, and composite time domain signals

Furthermore, the side lobes are significantly lower, reducing interference seen by adjacent channels. Input level pulse shaping, similar to the raised cosine shape discussed earlier, drives these side lobes down to very small levels [Bellamy 82, Korn 85]. The shaped QAM signal uses spectrum very efficiently. As we shall see, these types of techniques

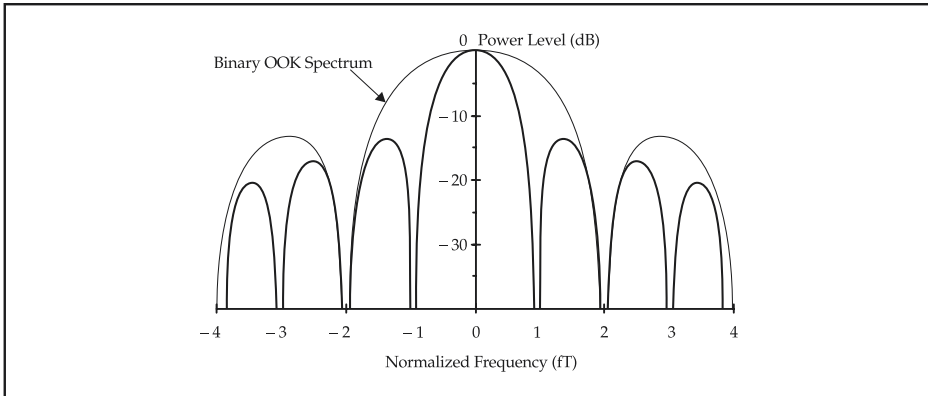


Figure 23-7. QAM spectrum compared with OOK spectrum

form the basis for higher-speed modems and xDSL transmission systems to deliver higher information rates over real-world, band-limited noisy physical layer channels.

ERROR MODELS AND CHANNEL CAPACITY

This section describes how basic physics, machines, and human intervention create errors on modern digital communication systems. But, despite the inevitable errors, communications engineers approach the ideal limits of performance for band-limited, noisy communications channels.

Typical Communications Channel Error Models

What causes errors on communications links? Why can't we just increase the transmit power? What causes noise anyway? This section answers these questions for a number of commonly utilized digital transmission systems. Over the years, communications engineers grouped channels into various types. The text covers three common cases, explaining how various channel characteristics manifest themselves to channel modems.

Additive White Gaussian Noise

You'll often see it abbreviated as AWGN in textbooks, and it has the distribution described earlier. This model covers electromagnetic channels (i.e., radio, satellite, and cellular), as well as thermal noise occurring in the electronics of all communications transmission systems, including electro-optical receivers in many fiber optic-based systems. The term "white" refers to the fact that the power spectrum of AWGN is flat across the signal passband. A corollary of this flat spectrum is the fact that the noise is uncorrelated from instant to instant. An example of AWGN is the sound heard when the TV is receiving no input, or the noise heard on an analog telephone line when no one is speaking.

Binary Symmetric Channel

The *binary symmetric channel (BSC)* is a good model for the low-level errors on fiber optic communications links. Typically, the random residual error rate on fiber optic links is on the order of 10^{-12} or less. This simple channel has a certain probability of bit error, p , for each bit independent of all other, preceding bits. As shown in Figure 23-8, the probability of receiving a bit correctly is $1 - p$. The channel error probability is also independent of whether the input was a one or a zero. The study of error-correcting codes later in this chapter uses this simple model.

Burst Error Channel Model

The basic burst error channel model is where a string of bits received over a communications channel has an error in its starting and ending positions, with an arbitrary number of other bit errors in between. Often, analysis assumes that every bit in the string between

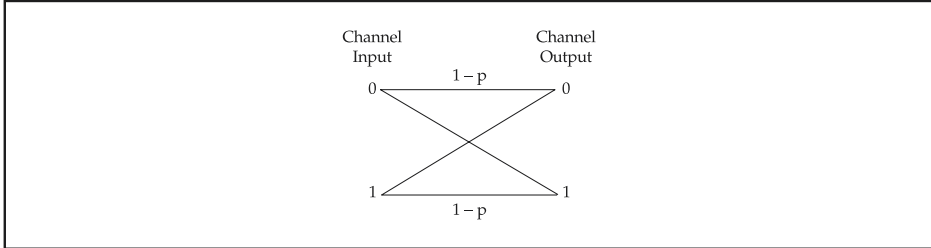


Figure 23-8. Binary symmetric channel error model

the first and last errors is randomly in error. Intermittent periods of high noise energy cause these types of errors on many atmospheric communications channels. Unshielded electrical transmission systems, like the twisted pair of copper wires leading to most residences, also pick up noise from lightning strikes, faulty vehicle ignitions, and power tools. Although less prone to bursts of errors, fiber optic transmission systems also experience bursts of errors due to protection switching in the equipment, maintenance activities that flex the fibers, and noise within the receiver electronics. Another commonly used model for bursts of errors is the Poisson arrival process defined in the next chapter.

Shannon's Channel Capacity

In 1948, Claude Shannon derived an upper bound on the maximum error-free information transfer rate that any communication system could achieve over a noisy communications channel with a particular frequency passband of W Hz (see [Gallagher 68], [Held 95], or [Shannon 48]). The disciplines of information and communication theory show that the performance of communications systems with appropriate source encoding, channel encoding, and signal selection approach this bound. The simple formula for this maximum rate, C , called *channel capacity*, is

$$C = W \log_2 (1 + \text{SNR})$$

where SNR is the signal-to-noise ratio of the communications channel. The model used by Shannon assumed additive white Gaussian noise.

Applying this theory to a familiar situation, a typical telephone line has a frequency passband of approximately 3000 Hz and a SNR of approximately 30 dB (i.e., the signal power is one thousand times greater than the noise power), note that channel capacity is approximately 30 Kbps from Shannon's formula. This is the maximum speed of the highest-speed telephone modems. Note that channel capacity refers to the information-carrying bit rate, and that many commercial telephone-grade modems also employ data compression of the source information to achieve even higher effective user data transfer rates. Data compression squeezes out redundancy in data using sophisticated coding schemes as summarized at the end of this chapter. This is how modems operate at 56 Kbps.

A basic lesson from Shannon's theorem is that, in order to increase the information transfer rate, a communications engineer must increase the frequency range (i.e., bandwidth) and/or the signal-to-noise ratio. Figure 23-9 plots this trade-off by depicting the normalized channel capacity, C/W , in units of bps per hertz versus the signal-to-noise ratio (SNR). Another way of looking at this result is to note that for a line transmission rate of $R = 1/T$ bits per second, each symbol conveyed over the link must carry $N = R/(2W)$ bits of information in order to approach channel capacity. In other words, the line bit rate R must be N times the Nyquist sampled signal rate $2W$ to achieve channel capacity.

We've indicated several popular modem standards on the chart: V.32 (9.6 Kbps), V.32bis (14.4 Kbps), and V.34 (28.8 Kbps). This chart also illustrates the reason for modems automatically negotiating down to a lower speed if the telephone connection is too noisy. Higher-speed modems over telephone lines also use several other techniques besides sending multiple bits per channel symbol. One that xDSL also employs is that of echo cancellation, where the transmitter is able to effectively cancel out its own signal. This means that both directions of communication can utilize the frequency passband simultaneously.

Shannon's theorem gave an important clue to modulator-demodulator designers to cram as many bits into the transmission of each discrete digital signal over the channel as possible. The evidence of their craft is the ever-increasing higher performance of modems for use over the telephone network to access the Web, high-performance radio and satellite communications, and the promise of ubiquitous high-speed digital subscriber line (xDSL) modems operating at over ten times the speeds of the fastest telephone-grade modems or ISDN lines. As described in Chapter 11, xDSL achieves this tremendous gain by using a much larger frequency passband.

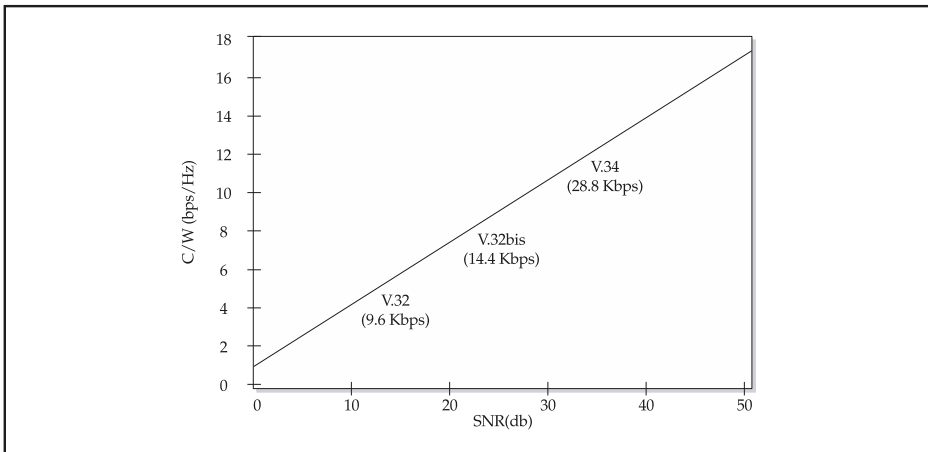


Figure 23-9. Normalized channel capacity (C/W) versus signal-to-noise ratio (SNR)

Error Performance of Common Modulation Methods

Figure 23-10 plots the bit error rate (BER) performance of the modulation schemes studied in the previous section versus the signal-to-noise ratio per bit (i.e., SNR/N). Note that the total signal power is N times that plotted in the figure, where N is the number of bits per channel symbol (i.e., $N = 1$ for OOK and $N = 4$ for QAM). Although the QAM makes slightly more efficient use of signal power than the simple OOK return to zero on/off pulse scheme, remember that the QAM system is sending four bits for every one conveyed by the OOK system in approximately the same amount of bandwidth.

Furthermore, as we saw in the last section, the QAM system also makes more efficient use of the available frequency spectrum. The increasing sophistication of digital electronics has enabled designers to apply these more sophisticated techniques on channels where frequency spectrum is scarce, such as telephone lines, terrestrial radio, and satellite transmission systems. Where frequency spectrum is not scarce—for example, on fiber optic cables—a simpler and cheaper scheme such as on/off keying works fine.

All right, the communication depicted in Figure 23-9 isn't error free as Shannon's theorem promises; and for lower signal-to-noise ratios, the performance is marginal. For example, for QAM a 30 dB channel, SNR equates to a signal-to-noise ratio per bit of approximately 8 dB, meaning that a significant number of bits in every hundred would be in error. How do communications engineers achieve a nearly lossless channel? What if

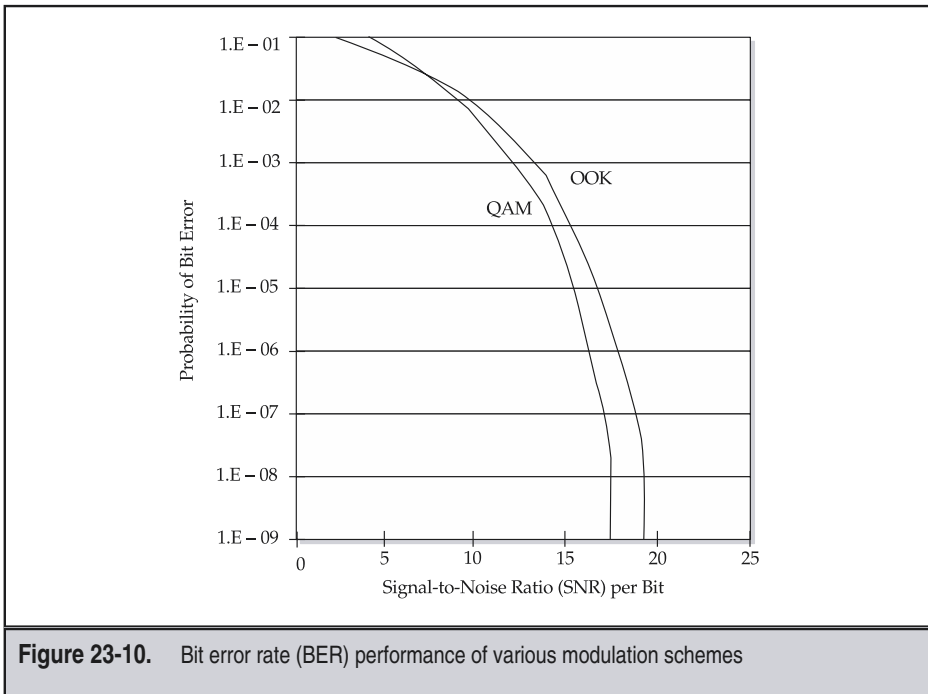


Figure 23-10. Bit error rate (BER) performance of various modulation schemes

my application can't tolerate the delay required to retransmit errored information? Communications engineers use three techniques to further improve the error rate. One is through the use of trellis coding that makes symbol transitions to adjacent places in the symbol constellation impossible. This improves the "distance" between signal points in the constellation, and hence markedly reduces the error rate. (See References [Proakis 83] and [Bellamy 82] for more information on trellis coding.) The second technique is through the use of error-correcting codes, as covered in the next section. A third method is to interleave the transmitted bits so that a burst of errors does not fall outside the error-correction capability of the receiver.

ERROR-DETECTING AND -CORRECTING CODES

Now that we've learned that signals can be transmitted over a channel frequency passband within a certain error rate, how do communications engineers achieve the effectively lossless communication essential to applications? The answer is through error-detecting and -correcting codes. This section reviews the performance of the two basic schemes in use today: the simple parity check typically done in software and the cyclical redundancy check frequently performed in hardware. The text then evaluates the performance of IP running over HDLC and ATM in terms of undetected error rate.

Simple Parity Check Schemes

The concept of parity checking originated in the era of digital computing in the 1950s. Since electronic circuits used for storage were quite unreliable, computer engineers designed a simple scheme in computer logic to compute a parity check. The required circuit was a modulo 2 adder (i.e., exclusive or logical function), implemented with a total of three electronic logic gates. The direct result of this calculation is called *even parity*, which means that the parity bit is zero if there are an even number of ones. A key concept defined during this period is the *Hamming distance* of a particular code. The distance of a code is the number of bit positions in which two codewords differ. A single parity bit added to a bit string results in a code with a distance of two—that is, at least two bit errors must occur to change one valid codeword into another. The parity check bit on 7-bit ASCII characters is an example of a simple parity scheme.

A means to do bitwise parity checking easily in software involves a bitwise exclusive or of subsequent segments of the message. Parity checking is commonly called a *checksum*. Both TCP and IP use this technique. The following simple example illustrates such a bitwise checksum performed across three bytes of binary data.

	0 0 1 1 1 0 1 1
	0 1 1 0 1 1 0 1
	1 1 0 1 0 1 1 1

Bit-wise checksum	1 0 0 0 0 0 0 1

A checksum encoder generates the checksum at the transmitter and sends it as part of the message to the receiver, which computes the checksum again on the received string of bits. This simple parity checksum detects any single bit error or combination of an odd number of bit errors in the message. If two bit errors (as well as all other combinations of even-numbered multiple bit errors) occur in the same position of the checksum columns, then the checksum will not detect the errors. For channels where the bit errors are random, and hence the likelihood of a single error is by far the largest, then a simple parity check works reasonably well—for example, within computer systems and over some local area network connections. However, for channels subject to bursts of errors, or nonnegligible error rates, then engineers include more powerful error-detection schemes in data communications networks. Since these noisy channels constitute the majority of the physical layer in real-world networks, we are fortunate indeed that communications engineers developed efficient and effective coding techniques that integrated circuits implement so inexpensively.

Cyclical Redundancy Check (CRC) Codes

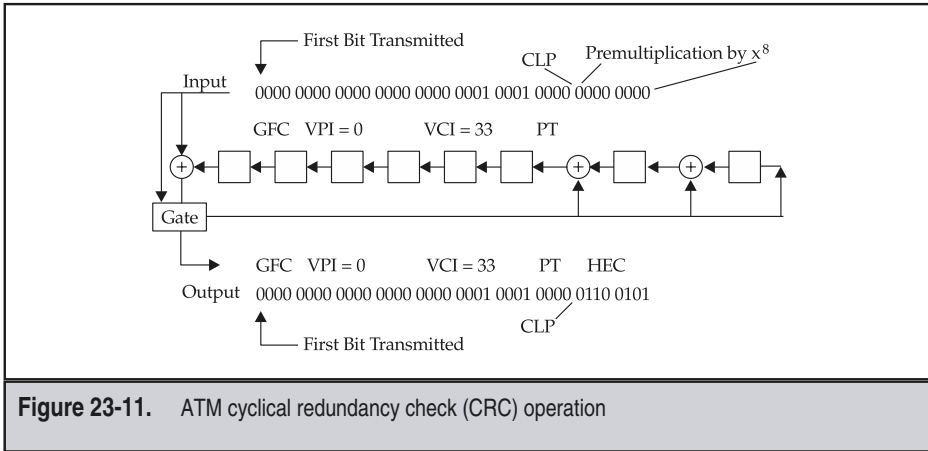
The magic that enabled reliable error detection over noisy communications channels for the early packet-switched networks has a rather complex mathematical basis, yet a relatively simple hardware implementation. The text takes a brief look at the mathematics followed by a description and example of the hardware implementation.

Coding theory represents strings of bits as polynomials with binary coefficients; for example, the bit string 1 0 0 1 would be represented as $x^3 + x^0$. Coding theorists say that a bit string of length n is represented by a polynomial of degree $n - 1$. Standards define specific CRC codes via a generator polynomial, $G(x)$, of degree k . An encoder adds k zeros to the end of a message polynomial $M(x)$ to yield a new polynomial $x^k M(x)$ and divides it modulo 2 by $G(x)$ to yield a remainder $R(x)$. The encoder then computes $T(x) = x^k M(x) - R(x)$. The receiver demodulates the bit pattern $T(x) + E(x)$, where $E(x)$ is a polynomial with ones representing the positions of bit errors. By the manner in which $T(x)$ was computed, $T(x)/G(x) = 0$. Hence, the receiver examines $E(x)$ to see if it contains any ones, which indicate errors on the channel. This mathematical treatment is highly simplified; see References [Tannenbaum 96] and [Peterson 72] for further details.

The generator polynomial for the 8-bit CRC used in the ATM Header Error Check (HEC) octet of every ATM cell [ITU I.432] is

$$G(x) = x^8 + x^2 + x + 1$$

Fortunately, the hardware implementation is easier to understand, as well as implement (it's also straightforward to implement on a spreadsheet or in a software lookup table). Figure 23-11 depicts an example of the ATM CRC code with generator polynomial of degree 8 protecting an ATM cell header for a user cell on VPI = 0, VCI = 33 without any congestion or AAL indications. Starting at the top of the figure, the premultiplication by x^8 means that the header is appended with eight zeros. During the first 20 bit positions corresponding to the ATM cell header GFC, VPI, VCI, PT, and CLP, the gate feeds



back the modulo 2 sum of the input and the leftmost shift register stage to the rightmost stages. The circled plus symbols indicate modulo 2 addition. For the last 8 bit positions, the gate breaks the feedback loop and shifts out the contents of the shift register, which are the HEC sequence 0110 0101.

The I.432 standard also recommends that ATM transmitters perform a bitwise exclusive or with the pattern 0101 0101 in order to improve cell delineation in the event of bit-slips. This “exclusive or” setting also ensures that if the entire header is all zeros, then the input to the physical link scrambler is nonzero. An ATM receiver first subtracts this pattern from the HEC octet prior to shifting the received bits into the shift register just described. After shifting all of the received bits into the register, the pattern remaining is called the *syndrome*. If no errors occurred, then the syndrome is identical to the received HEC field. If errors occur, then the syndrome indicates the single bit error correction required, or the presence of an uncorrectable double bit error.

There are several different ways to use cyclic codes to avoid all zero bit patterns. Some standards—for example, HDLC and AAL5—initialize the shift register to all ones so that an all-zero information packet results in a nonzero CRC. The basic operation is then the same as that described previously using a shift register with taps corresponding to the nonzero coefficients of the code’s generator polynomial.

Performance of ATM’s HEC

This section gives some handy, practical tips for selecting and evaluating the performance of particular codes. When comparing the performance of different CRC standards for your application, note that the error-detection distance (i.e., the number of bit positions in which packets differ) is no more than the number of nonzero coefficients in the generator polynomial [Peterson 72]. For example, the number of nonzero coefficients in

the ATM HEC generator polynomial is four, and, in fact, the minimum error-detection distance for this code is also four.

CRC codes also have excellent error burst detection properties. Specifically, a CRC code with degree n (i.e., the highest numbered coefficient in the generator polynomial) detects all bursts of errors up to length n bits, assuming that there are no other errors in the packet. A burst of errors is defined as errors in the first and last positions with an arbitrary number of other bit errors in between. A CRC code can detect more errors in a burst than those randomly distributed over a packet because all of the information is contained within the shift register in a burst. Thus, CRC codes work well on burst error channels, but they also provide good error detection on random error channels.

The standards give two options for using the HEC: error correction and error detection. In the presence of random errors, the probability of x errors in a 40-bit cell header is given by the binomial distribution $b(40,x)$ described earlier with p set equal to the probability of a random bit error. With header error detection, the probability that an errored cell header is falsely mistaken as a valid header is determined by the probability of randomly matching a valid codeword ($1/256$) if three or more bit errors occur. Therefore, the probability of falsely passing an errored header given that the receiver performs HEC error detection is

$$P[\text{False | Detection}] = \frac{(1 - b(40,0) - b(40,1) - b(40,2))}{256}$$

When using HEC detection, the receiver discards the cell if the decoder detects any errors. Of course, the receiver cannot detect a falsely matched valid cell header—hence, the probability of discard given header detection is

$$\text{Pr}[\text{Discard | Detection}] = (1 - b(40,0)) (1 - P[\text{False | Detection}])$$

If an implementation employs header error correction, then a false match occurs if three or more bit errors cause an exact codeword match—or if the errors result in a codeword that appears to have only one bit error and is then inadvertently corrected into a valid codeword. Actual testing uncovered this unexpected result. In fact, the probability of invalid cells is approximately 40 times greater than when the receiver employs HEC detection as given by the following formula:

$$P[\text{False | Correction}] = \frac{41(1 - b(40,0) - b(40,1) - b(40,2))}{256}$$

When using HEC correction, the receiver corrects all single bit errors and detects all other bit errors that don't result in a falsely matched valid cell header. Therefore, the probability of discard given that header correction is performed is

$$\text{Pr}[\text{Discard | Correction}] = (1 - b(40,0) - b(40,1))(1 - P[\text{False | Correction}])$$

Figure 23-12 plots the results of these calculations. The values of $\text{Pr}[\text{Discard | Correction}]$ and $P[\text{False | Correction}]$ from the preceding formulas correspond closely to those of Figure A.1/I.432 of Reference ITU-T I.432. The figure also plots $\text{Pr}[\text{Discard | Detection}]$

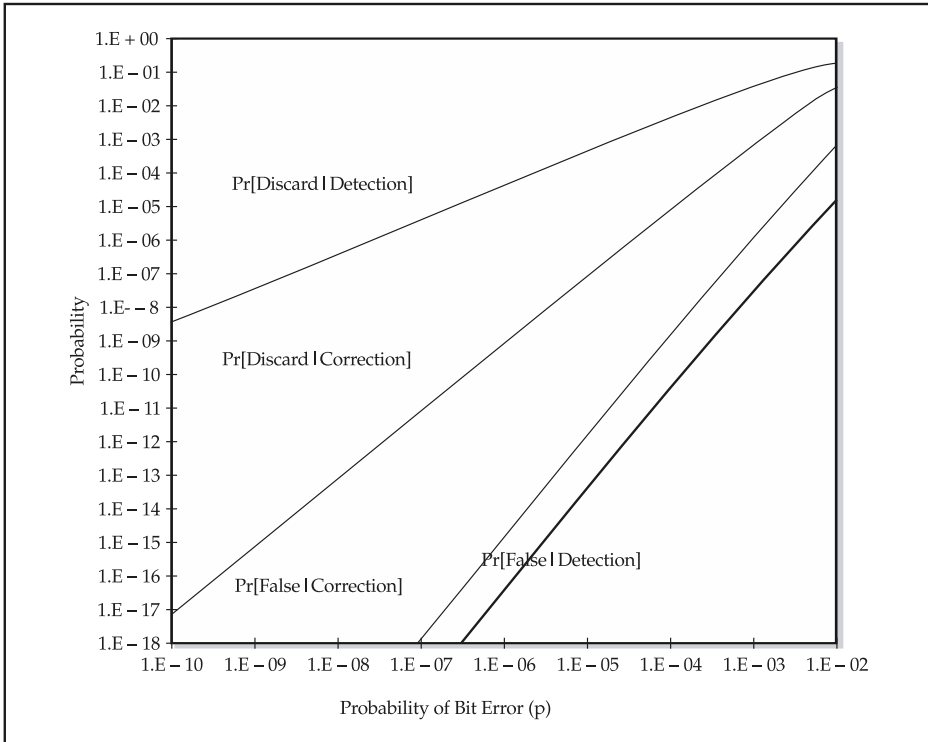


Figure 23-12. Undetected cell header error for HEC correction and detection

and $\text{Pr}[\text{False} \mid \text{Detection}]$ for comparison purposes. Note that the use of error detection results in a false cell matching probability somewhat less than when using correction; in fact, they differ by a factor of 40, as was just shown. The probability of discarding a cell due to detected errors, however, is much greater than when using HEC detection when compared with the probability of cell discard when HEC correction is used.

If your application can tolerate a few false cells via further error checking using higher-layer protocols, then use HEC correction. If your application doesn't have a higher-layer protocol—for example, video over ATM—you may want to use error detection instead.

A useful computational trick when computing small numbers (such as the ones in this example) in a spreadsheet is to raise 10 to the power of the sum of the base 10 logarithms of product terms. If you don't do this, then you'll find that you're getting a result of one or zero instead.

Undetected Error Performance of HDLC and AAL5

Despite the good error performance of CRC coding, the bit stuffing mechanism of HDLC is susceptible to random and burst errors [Selga 81]. The bit stuffing mechanism used to eliminate the occurrence of a flag sequence within an HDLC frame as described in Chapter 7 causes this flaw. One or two bit errors in the wrong place in a received frame creates a valid flag field, which truncates the frame. Also, the trailing flag field may be obliterated by bit errors. The resulting probability of undetected error for HDLC is well approximated by the following formula:

$$\Pr[\text{Undetected Error} \mid \text{HDLC}] \approx \left(1.36 kp + \left(\frac{m}{k}\right) p^4 \right) 2^{-16}$$

where the variable k represents the number of bytes in the HDLC frame, p is the random bit error probability, and m is the average HDLC frame length (in bits) after bit stuffing given by the following formula:

$$m = 8 \left(\frac{64}{63} k + 2 \right)$$

For comparison purposes, the undetected AAL5 PDU performance does not have the bit stuffing problem. Since the AAL5 32-bit CRC generator polynomial has 15 non-zero coefficients, its Hamming distance is also 15. Hence, a particular pattern of exactly 15 random bit errors must occur and still result in a valid CRC field, which we assume occurs randomly with probability 2^{-32} . Hence, the undetected frame-level error performance of AAL5 for an information field of k octets is

$$\Pr[\text{Undetected Error} \mid \text{AAL5}] \approx b\left(8 \left\lceil \frac{k+8}{48} \right\rceil, 15, p\right) 2^{-32}$$

where $b(n,x,p)$ is the value from the binomial distribution defined above for bit error probability p with $\lceil x \rceil$ denoting the smallest integer greater than x , commonly known as the “ceiling” function, implemented in Microsoft Excel as `CEILING(x,1)`.

Figure 23-13 depicts the undetected frame error rate for HDLC and ATM’s AAL5 versus random bit error probability p . Note that the undetected error performance approaches 1 in 10,000 for high-error rate channels for HDLC. This is one reason why higher-level protocols like TCP/IP add a length field and additional parity checks. On the other hand, the undetected error probability for AAL5 is negligible, even at high channel error rates. The practical implication of this analysis is that ATM AAL5 is much better suited for noisy, error-prone channels than HDLC is when the objective is reliable error-free operation.

DATA COMPRESSION

Most real-world sources generate data sequences that contain redundancy. For example, in ordinary text, some letters are much more likely to occur than others. For example, the

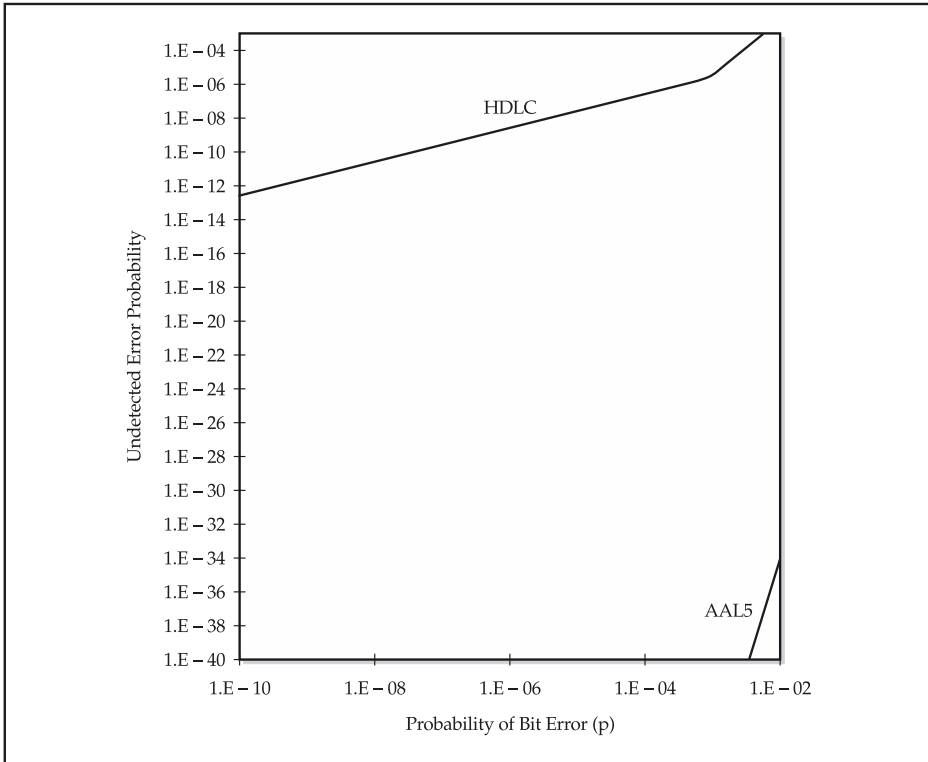


Figure 23-13. Undetected error rate for HDLC and AAL5

letter “e” is hundreds to thousands of times more likely to occur in common text than the letter “z.” The basic idea of data compression exploits this difference in likelihood of particular inputs by coding frequently occurring source symbols with fewer bits and coding rarely occurring source symbols with longer sequences of bits. The net result is a reduction in the average number of bits required to transmit a long message. Another simple data compression method used in facsimile transmission is called *run-length coding*. Since most facsimile input consists of characters or lines, a *run-length coder* scans the input pages one scan line at a time from left to right. If the input page is dark in the scanned dot, then the source encodes a one; otherwise, it encodes a zero. If the source detects a long run of ones or zeros, then it inserts a special source symbol indicating that a specified number of subsequent scan positions have the same value. The receiver reverses the process by printing out a series of dots for a long run of ones, or leaving the fax output blank if a long run of zeros is received. Data compression requires additional soft-

ware processing, or even custom hardware to operate at higher speeds. If bandwidth is scarce, then investigate the use of data compression to see if the additional processing costs less than the incremental cost required to increase the available bandwidth.

REVIEW

This chapter discussed several key aspects of communications engineering philosophy: probability theory, performance measurement, digital communications, error rates, channel capacity, error-detecting and -correcting codes, and data compression. The communications system model includes source coding (data compression), channel coding, and modulation on the transmitter side, along with corresponding functions at a receiver, in order to achieve reliable digital transmission. We then covered some basics of probability theory using the discrete binomial distribution and the continuous normal (or Gaussian) distribution as background. The text then summarized some concepts of modulated signals and their frequency spectra. We then introduced the seminal concept of channel capacity, and we reviewed the bit error performance of some common modulation schemes. The chapter then introduced the techniques of parity checksums and cyclic redundancy check codes used to detect channel errors. The text applied this theory in a comparison of undetected error rates for ATM HEC, HDLC, and AAL5. Finally, the chapter concluded with a brief overview of data compression techniques.

CHAPTER 24

Traffic Engineering

This chapter deals with the modeling of traffic sources and switch performance. It covers many useful approximations for estimating performance. The text begins with a discussion of traffic modeling philosophies as an introduction to an application-oriented approach. We introduce several simple models for source traffic to illustrate key concepts. The treatment then introduces the traditional telephone system call arrival, blocking, and queuing models, and applies them to ATM and MPLS. The text analyzes the performance of commonly used buffering schemes. Next, the analysis continues with an evaluation of the performance resulting from multiplexing many constant-rate and variable-rate traffic sources—an important consideration in the economics of integrated voice, video, and data networks. The coverage then moves on to define the concept of equivalent bandwidth. The text defines statistical multiplexing gain and the traffic characteristics for which it is attractive. Finally, the chapter concludes with an analysis of priority queuing traffic control performance. All formulas used in these examples are simple enough for spreadsheet computation so that you can use them to evaluate a particular switching machine or network configuration. Throughout the chapter, the text gives references to more sophisticated models for the interested reader.

PHILOSOPHY

This section discusses several dimensions of traffic engineering philosophy: source model traffic parameter characterization, performance specification and measurement, and modeling accuracy.

Source Model Traffic Parameter Characteristics

There are two basic philosophies for characterizing source traffic parameters: deterministic and random. These approaches often embody a trade-off between accuracy and simplicity.

Deterministic parameters use the notion of the traffic contract outlined in Chapters 20 and 21, with conformance verifiable on either a per-cell or per-packet basis using the leaky or token bucket algorithm. The IP token bucket and ATM leaky bucket precisely define the worst-case arrival pattern of a sequence of packets or cells, respectively. Thus, the deterministic traffic model clearly bounds the source characteristics in a measurable, repeatable, and unambiguous way. Unfortunately, accurately characterizing a source and formulating these parameters can be rather complicated. On the other hand, this traffic model clearly defines the source characteristics as understood by the user and the network.

The other philosophy for modeling source behavior utilizes random (also called probabilistic or stochastic) models for traffic parameters. Usually, random-model parameters correspond to measurable long-term averages, which describe the short-term behavior in a statistical manner. However, unlike the deterministic model, random models have behavior that is potentially unbounded and with results that are often difficult to repeat except in a statistical sense. Since the method and interval for averaging differ, conformance testing should define the details of the measurement method. With these

additional assumptions, the user and network can agree on performance for a certain level of traffic throughput. While these statistical methods are not standardized, they are useful approximations to the deterministic traffic contract behavior. These methods are useful in analysis when employing a simple statistical model, as shown in this chapter.

Modeling Accuracy

A key aspect of traffic engineering philosophy relates to the required accuracy of the model. As expected, the more complicated the model, the more difficult the results are to understand and calculate. A good guideline is to make the accuracy of the switch and network model comparable to the accuracy of the source model traffic. If you know only approximate, high-level information about the source, then an approximate, simple switch and network model is appropriate. If you know a great deal of accurate information about the source traffic, then an investment in an accurate switch and network model, such as a detailed simulation, is appropriate.

While theoretically optimal, detailed source modeling can be very complex, and usually requires computer-based simulation. Often this level of detail is not available for the source traffic. Using source traffic details and an accurate switch and network model will result in the most realistic results, but may not always be appropriate.

When either traffic or switch and network details are not available, approximations are the only avenue that remains. Approximate modeling is usually simpler and can often be done using only analytical methods. One advantage of the analytical method is that insight into relative behavior and trade-offs is much clearer. The analytical method is the approach used in this book. One word of caution remains, however: these simplified models may yield overly optimistic or pessimistic results, depending upon the relationship of the simplifying assumptions to a specific real-world network. Modeling should be an ongoing process. As you obtain more information about the source characteristics, device performance, and quality expectations, feed these back into the modeling effort. For this reason, modeling has a close relationship to the performance measurement aspects of network management, as discussed in Part 7.

OVERVIEW OF QUEUING THEORY

The use of a particular source model, and its accurate representation of real traffic, is a hotly debated topic, and the subject of intense ongoing research and publication. This section defines some basic probability and queuing theory concepts utilized within this chapter to model commonly encountered traffic engineering problems.

General Source Model Parameters

This section defines some general source model parameters used throughout the remainder of this chapter, and in many papers and publications listed in the References section at the end of this book.

A source may send at a peak rate, limited by transmission speed of the physical link or other mechanisms, such as traffic shaping or processor speed. Typically, we express the rate in bits/second for physical transmission links, packets/second for frame-based protocols like IP, or cells/second for ATM networks. Usually, the peak rate is equivalent to a short-term measurement corresponding to a small number of packets or cells. A longer-term average corresponds to the average rate.

Figure 24-1 shows an example of a voice conversation that gives an illustration of how peak and average rates capture the essence of actual communications network traffic. Often, when talking on the telephone, one party speaks while the other listens. There are, however, instances when we interrupt each other or make brief comments while the other person speaks, as a part of normal interactive conversation. The two traces in the figure depict the transmission of speech in each direction at a peak rate when one person is talking and the other is listening over the duration of a short telephone call. Typically, the peak rate on TDM-based digital telephone systems is 64 Kbps, as described in Chapter 7. The figure shows the average transmission rate over the duration of the call as a horizontal line. In this example, the person in the upper part of the figure talked approximately twice as much as the one in the lower part of the figure.

Burstiness is another measure of source traffic. Traffic engineers call a source that sends traffic at markedly different rates at different times *bursty*, while a source that always sends at the same rate is called *nonbursty*. A simple measure of burstiness is the ratio of peak-to-average rates of traffic over a specific sampling period, as specified by the following equation:

$$\text{Burstiness} = \frac{\text{Peak Rate}}{\text{Average Rate}}$$

This simple measure captures only a portion of the source's traffic characteristics. For example, the traffic from two sources could have the same peak-to-average ratio, but have quite different characteristics in terms of the time scale of the traffic bursts. We will

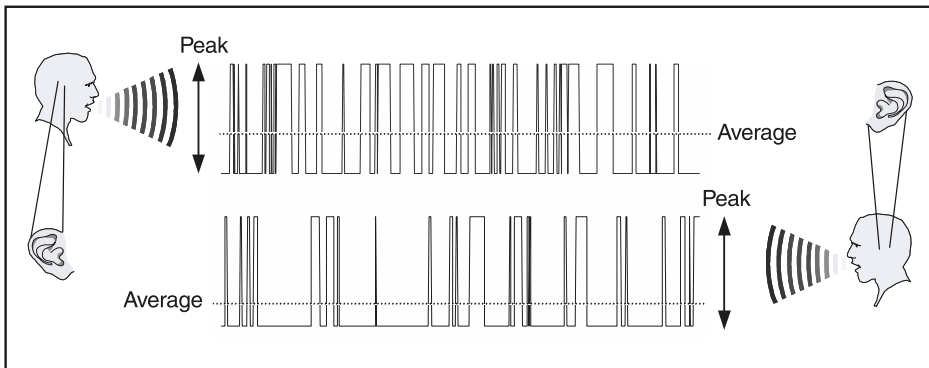


Figure 24-1. Example of peak and average rates resulting from a voice conversation

analyze other definitions of burstiness later in this book, beginning with the burst duration as described in the next section.

The *source activity probability* p is a measure of how frequently the source sends, defined by the following formula:

$$p = \frac{\text{Avg}[\text{Active}]}{\text{Avg}[\text{Inactive}] + \text{Avg}[\text{Active}]} = \frac{\text{Average rate}}{\text{Peak rate}} = \frac{1}{b}$$

As shown in the formula, the source activity probability is inversely proportional to the burstiness b . For example, if a source has a peak rate of 10 Mbps and an average rate of 500 Kbps, then the traffic source has a burstiness factor of 20. Alternatively, the source activity probability is only 5 percent. Typically, local area network (LAN) users exhibit burstiness on the order of 10 to 100, or more. In other words, they use the LAN at the peak rate less than 10 percent of the time. Usually, burstiness approaches unity if the source aggregates the outputs from many individual users. Such is often the case in wide area network (WAN) backbone networks. For example, a router connecting a medium-to-large enterprise to the Internet aggregates the traffic flowing to and from many individual workstations. In these situations, the burstiness often approaches 1 during the busiest hours of the workday. On the other hand, the burstiness of a dial-up connection for an individual Internet user can vary widely depending upon the application. If the user is downloading a large file, then the burstiness in the network-to-user direction approaches 1; that is, the activity is continuous. Note that a casual Web surfer may generate traffic that has a burstiness factor of 10 or more!

Generally, the data stream generated by many sources has a statistically predictable on-off pattern. The intervals of time that a source can generate data are a result of many factors. For example, a person can speak for only so long before he or she must take a breath. Many computer communication protocols have a maximum window, which limits the amount of data that can be sent without acknowledgment. Sometimes, the network may limit the maximum duration of a burst to ensure that it can deliver the desired service quality.

Continuing the preceding example, note that the specification of a source with a peak rate of 10 Mbps and an average rate of 500 Kbps is incomplete. For example, a source that sends a 500,000-bit burst once a second has the same average rate as a source that sends 500-bit bursts 1000 times a second. Thus, a complete traffic specification must also quantify the burst duration in addition to the peak and average rates.

Traffic engineers call the action of combining the outputs from many variable-rate sources into one data stream statistical multiplexing. Figure 24-2 illustrates example plots of the activity of a number of sources: bulk data, interactive data, video, and voice. Each has peak and average rates and burst durations as defined earlier in this section. The trace at the bottom of the figure shows the resulting aggregate bit rate resulting from adding the output of the various sources. Note how the statistically multiplexed data stream reaches the peak rate only a small fraction of the time. This is the basis for networks providing statistical guarantees on QoS. Namely, when aggregating enough traffic, the probability of arriving traffic exceeding allocated resources can be statistically predicted with high confidence. We cover this effect and the topic of statistical multiplexing in detail later in this chapter.

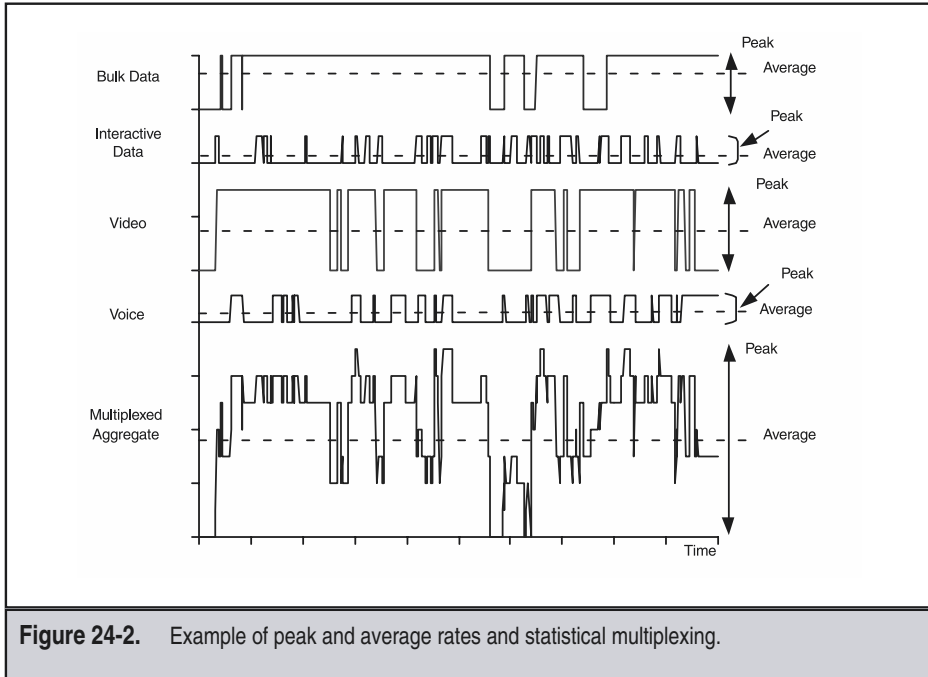


Figure 24-2. Example of peak and average rates and statistical multiplexing.

Poisson Arrivals and Markov Processes

This section describes the Poisson (or Markov) random arrival processes with reference to Figure 24-3. Interestingly, Poisson developed these statistical models on the basis of his experience in Napoleon's army regarding the likelihood of soldiers dying after being kicked in the head by their horses. Like many random events (most not quite so morbid), Poisson arrivals occur such that for each increment of time (T), no matter how large or small, the probability of arrival is independent of any previous history. These events may be either individual cells or packets, a burst of cells or packets, cell or packet service completions, or other arbitrary events.

The probability that the interarrival time between events t , as shown in Figure 24-3, has a certain value is called the *interarrival time probability density*. The following formula gives the resulting probability that the interarrival time t is equal to some value x when the average arrival rate is λ events per second:

$$\text{Prob}(t = x) = \lambda e^{-\lambda x}$$

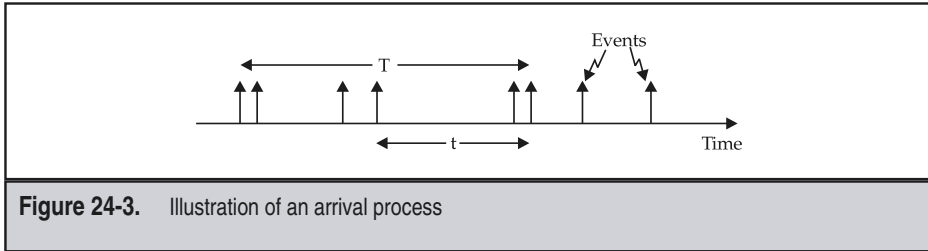


Figure 24-3. Illustration of an arrival process

Queuing theorists call Poisson arrivals a *memoryless process*, because the probability that the interarrival time will be x seconds is independent of the *memory* of how much time has already expired. This fact greatly simplifies the analysis of random processes, since no past history, or memory, affects the computation regarding the statistics of the next arrival time. These types of processes are commonly known as *Markov processes*, named after the Russian mathematician of the nineteenth century.

The probability that n independent arrivals occur in T seconds is given by the familiar *Poisson distribution*:

$$\text{Prob}(n, T) = \frac{(\lambda T)^n}{n!} e^{-\lambda T}$$

We combine these two thoughts in a commonly used model called the Markov Modulated Poisson Process (MMPP). There are two basic types of this process: the *discrete* type that corresponds to ATM cells, and the *continuous* type that corresponds to higher-layer protocol data units (PDUs) that generate bursts of cells. The next two figures give an example for the discrete and continuous Markov process models.

The labels on the arrows of Figure 24-4 show the probability that the source transitions between active and inactive bursting states, or else remains in the same state for each cell time. In other words, during each cell time, the source makes a state transition, either to the other state or back to itself, with the probability for either action indicated by the arrows in the diagram.

The burstiness, or peak-to-average ratio, of the *discrete* source model is:

$$b = \frac{\alpha + \beta}{\alpha}$$

where α is the probability that a burst begins and β is the probability that a burst completes within a small time interval.

Often we think in terms of β^{-1} , which has units of the average number of seconds per burst. We define D as the cell quantization time, having units of seconds per cell. Therefore, αD defines the probability that a burst begins in a particular cell time, and βD defines the probability that a burst ends in a particular cell time. The mean of the standard geometric series determines the average burst duration (or active period) as $(\beta D)^{-1}$ and the average inactive period as $(\alpha D)^{-1}$.

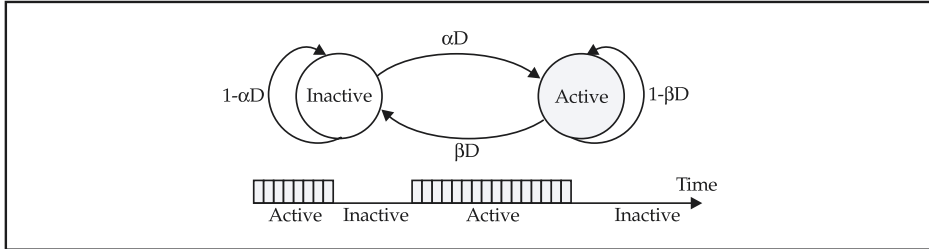


Figure 24-4. Discrete-time Markov process model

Figure 24-5 illustrates the *continuous-time* Markov process, which models the statistics for the time duration of bursts instead of modeling the individual cells as the discrete model does. Since this model eliminates the effects of quantization inherent in segmentation and reassembly, it is inherently less accurate. However, its analysis is somewhat simpler and it also models packets; therefore, this book relies on it. Queuing theorists call the diagram depicted in Figure 24-5 a *state transition rate diagram* [Kleinrock 75], since the variables associated with the arrows refer to the rate exponent in the negative exponential distribution introduced earlier in this chapter. Specifically, α is the arrival rate of bursts and β is the rate at which bursts complete service. Both the discrete and continuous Markov models yield equivalent results except for the cell quantization factor D .

The corresponding burstiness b for the continuous process is

$$b = \frac{\alpha + \beta}{\alpha}$$

and the average burst duration (in seconds) is

$$d = \frac{1}{\beta}$$

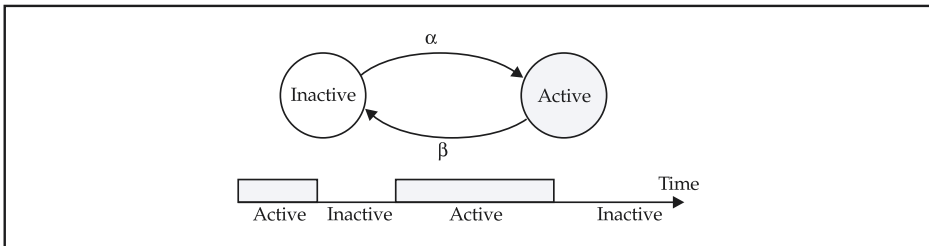


Figure 24-5. Continuous-time Markov process model

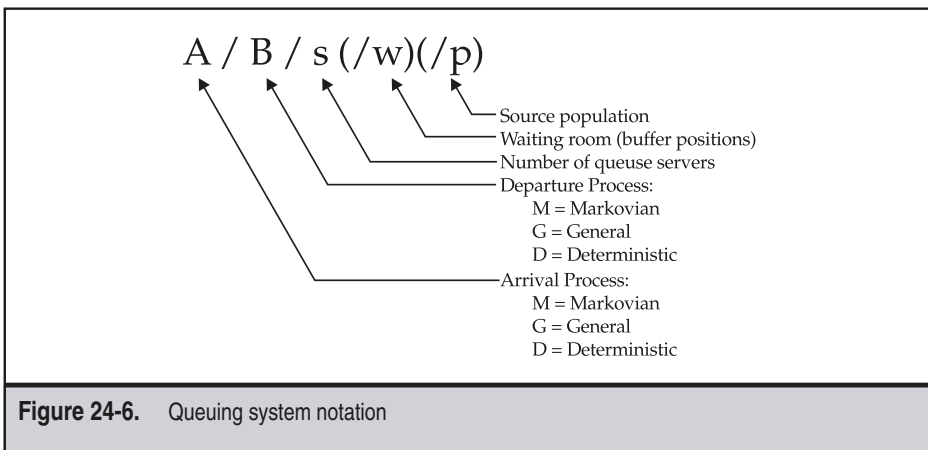
Note how these formulas are identical to the discrete case except for the absence of the discrete cell time D in the denominator of the equation for the average burst duration in the continuous model.

Another distribution sometimes used to model extremely bursty traffic is that of the hyperexponential, which is effectively the weighted sum of a number of negative exponential arrivals. This turns out to be a more pessimistic model than Poisson traffic because bursts and burst arrivals are more closely clumped together. For further information on this distribution, see [Kleinrock 75].

Recent work based upon actual LAN traffic measurements indicates that these traditional traffic models may be overly optimistic. For further information on this work, see [Fowler 91], [Leland 93], [Stallings 98], and [McDysan 00]. These results show that LAN and Internet traffic is *self-similar*, which means that the traffic has similar properties across a broad range of time scales over which it is observed. This is in sharp contrast to the Poisson and Markovian models, where the traffic tends to become smoother and more predictable when considering longer and longer time averages. Through a simple example, this chapter illustrates the impact of these different traffic models on the required buffer capacity in a switch or router when designing to meet a specific loss objective.

Queuing System Models

Figure 24-6 depicts a widely used notation employed to categorize queuing systems. This chapter makes use of this shorthand terminology, and it is widely used in the technical literature. Hence, readers doing further research should become familiar with this notation, originally attributed to Kendall. The notation designates arrival processes (denoted as A and B) as M for Markovian, as described previously; G for General; or D for Deterministic. The required parameter s defines the number of servers in the queuing system; for

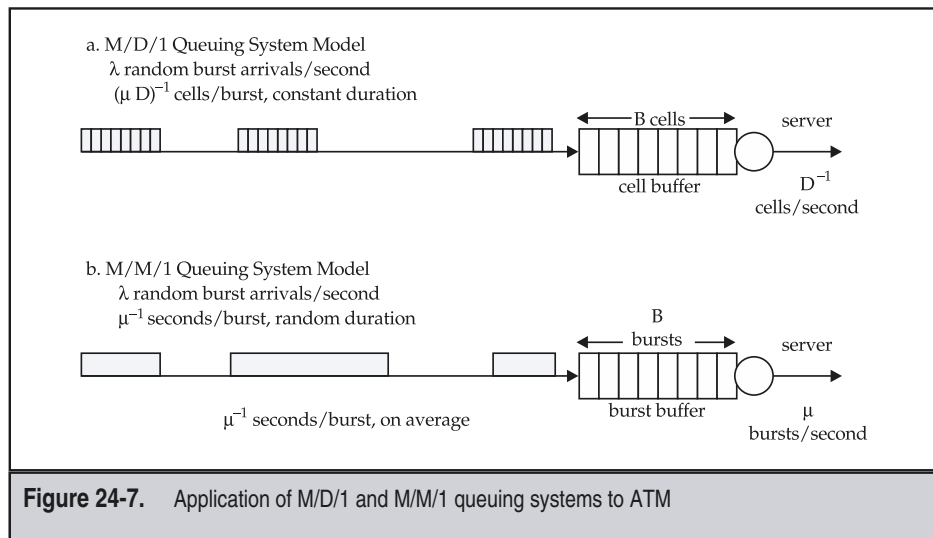


example, in the case of communications networks, this would be the transmission path and a single server. The optional w and p parameters specify the waiting room for unserved arrivals and the source population, respectively.

Figure 24-7 illustrates two particular examples of queuing systems, namely, the $M/D/1$ and $M/M/1$ systems. Each of these queuing systems has Markovian arrivals at a rate of λ bursts per second. The $M/D/1$ system has constant-length bursts, as shown in Figure 24-7a, while the $M/M/1$ system has random-length bursts with a negative exponential distribution (Markov), as shown in Figure 24-7b. As we will see, $M/D/1$ is a good model for applications like voice over ATM, while $M/M/1$ is a good model for packet traffic over ATM or MPLS.

The parameter μ^{-1} defines how many seconds the transmission link requires to send each burst. For the $M/D/1$ system, this is the constant or fixed length of every burst; while in the $M/M/1$ system, this is an exponentially distributed random number with average length μ^{-1} . Both queuing models also have a single server (i.e., physical transmission link), and an infinite population (number of potential bursts) and theoretically infinite waiting room (buffer space). The units of the buffer in the $M/D/1$ model are cells, while in the $M/M/1$ case, the units of the buffer are bursts. Modeling a real-world finite buffer case is more complex; however, some simple approximations are quite useful.

This is a good example of the trade-offs encountered in modeling. The $M/D/1$ system accurately represents the fact that the buffers in the switch are in units of cells; however, the model also assumes that bursts are all of the same fixed length. While this is a good model for voice, video, and some data sources, it is a bad model for many packetized data applications such as file transfers or Web surfing. On the other hand, the $M/M/1$ system does not model the switch buffers accurately, because it expresses waiting room in units



of bursts and not cells or variable-length packets; however, the modeling of random burst lengths is more appropriate to many traffic sources. Furthermore, the M/M/1 model is the simplest queuing model to analyze. Therefore, this text uses the M/M/1 model extensively to illustrate general characteristic of ATM and MPLS packet-switching systems. In general, if the traffic is more deterministic than the M/M/1 model (for example, more like the M/D/1 model), then the M/M/1 model is pessimistic (there will actually be less queuing and less delay in the modeled network). If the traffic is more bursty than the M/M/1 model, then the M/M/1 results will be optimistic (there will actually be more queuing and more delay in the modeled network). In this case, the self-similarity-type models are more appropriate.

In many of the following results, the system delay and loss performance will use the concept of offered load ρ as defined by the following formula:

$$\rho = \frac{\lambda}{\mu}$$

Recalling that λ is the average number of arriving bursts per second, and that μ^{-1} is the average number of seconds per burst required by the transmission link, observe that the offered load ρ is unitless. Thus, the offered load has the interpretation of the average fraction of the resource capacity the source would use if no loss occurred.

The service rate μ is computed as follows for a burst of B bytes at a line rate of R bits per second:

$$\mu = \frac{R}{8B} \left(\frac{\text{Bursts}}{\text{Second}} \right)$$

The probability that n bursts are in the M/M/1 queue awaiting service is

$$\text{Prob}[n \text{ bursts in M/M/1 queue}] = \rho^n (1 - \rho)$$

The average queuing delay (i.e., waiting time) expressed in seconds for the M/M/1 system is

$$\text{Avg}[M/M1 \text{ queuing delay}] = \frac{\rho/\mu}{1-\rho}$$

The total delay that a cell or packet experiences is the sum of the waiting time and the service time (e.g., μ^{-1} for an M/M/1 queue). Since the service time is a constant, we focus on the effect on waiting time. M/D/1 queuing predicts better performance than M/M/1. Indeed, the average total delay of M/D/1 queuing is exactly one-half of the M/M/1 delay. The probability for the number of cells in the M/D/1 queue is much more complicated, which is one reason the text uses the M/M/1 model in many of the following examples. The CBR model described later in this chapter gives an approximation to the M/D/1 distribution.

CALL ATTEMPT RATES, BLOCKING, AND QUEUING

This section looks at traditional traffic modeling derived from over a century of experience with telephone networks, largely attributed to work published by the Danish mathematician A.K. Erlang in 1917. Since ATM and MPLS both use the connection-oriented paradigm, and are in fact similar in nature to telephone calls, these older call models still apply. One significant difference, however, is that unlike phone calls, which are always the same speed, ATM SVCs and MPLS LSPs may have vastly different bandwidths. Furthermore, ATM SVCs may request different service categories and QoS parameters.

Statistical Model for Call Attempts

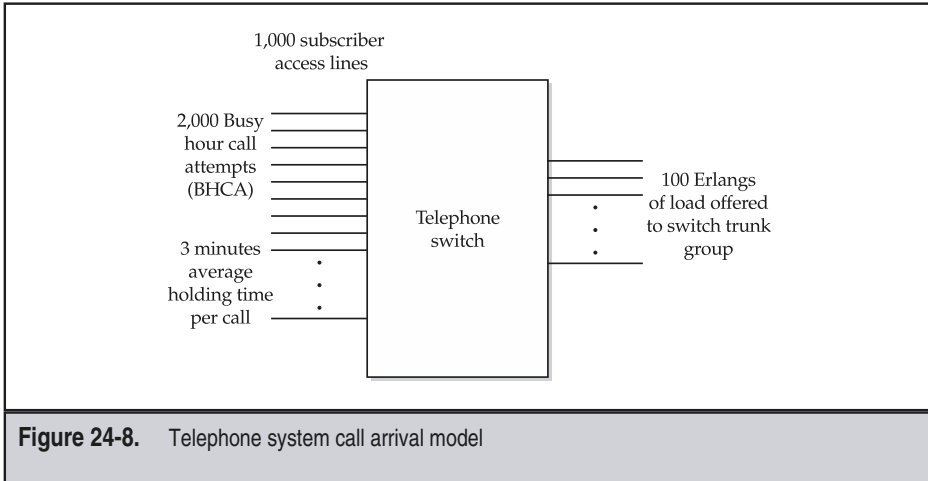
Through extensive measurements, traffic engineering experts know that the Markov process is a good model for telephone call attempts. The primary parameters are the call arrival rate λ , usually expressed in terms of busy hour call attempts (BHCA), and the average call holding time (i.e., call duration) $T = \mu^{-1}$. Without any blocking, the average number of calls in progress during the busy hour in such a system is λT . Traditionally, telephony engineers assign the units of offered *Erlangs* to this quantity, although, in fact, the measure is unitless. In the following sections we refer to the average offered traffic load (expressed in units of Erlangs) as

$$a = \lambda T \text{ average offered load (in units of Erlangs)}$$

In a real telephone network, blocking occurs with a certain probability B . Therefore, telephone engineers say that the system carries $(1 - B)a$ Erlangs. Erlang also modeled systems that queue calls instead of blocking them. For example, a call answering system that places calling users on hold listening to music if all the operators are occupied effectively queues calls instead of blocking them via responding with a busy signal. We study these two types of systems—blocked calls cleared and blocked calls held—in the next sections.

Implicit in the Erlang model is the assumption that the switching system under consideration supports a large number of users. Furthermore, the model assumes that only a fraction of these users are active during the busy hour. When these assumptions are not valid, use a more complicated model developed by one of Erlang's contemporaries, Engset, instead [Kleinrock 75].

Let's look at a simple numerical example with reference to Figure 24-8. A group of 1000 subscriber lines originates an average of $\lambda = 2000$ call attempts during the busy hour. Each completed call lasts for 3 minutes on average, or $T = 0.05$ hours. Thus, the offered load to the trunk on the telephone switch is $a = \lambda T = 2000 \times 0.05 = 100$ Erlangs. The interpretation of this model is that without any call blocking, on average only 100 of the subscribers are actually on the phone at any point in time. Note that this type of model also applies to calls placed to a pool of dial-up modems via which subscribers attempt to access the Internet. In the next sections, we examine the performance when the switching system blocks or queues calls that exceed the trunk capacity.



Another note on historical terminology for telephone call attempt rates that you may encounter is that of the call century seconds (CCS). The name CCS derived from the operation where a camera took a picture of the call peg counters on electromechanical switches once every 100 seconds for billing purposes. If the counter advanced between photographs for a particular subscriber line, then the carrier assumed that the call was active for one hundred (a century) seconds, which explains the name CCS. Since an Erlang corresponds to a call with holding time equal to 3600 seconds, we have the relation that 1 Erlang is equivalent to 36 CCS. Prior to extended dial-up sessions on the Web, a typical residential phone line carried 3 to 6 CCS on average; that is, its average load was between 8 percent and 16 percent. Now, many residential lines carry 9 to 12 CCS.

Erlang's Blocked Calls Cleared Formula

A *blocked calls cleared* switching system has Markovian call arrivals, Markovian call durations and n servers (trunks), and n waiting positions. Therefore, an $M/M/n/n$ queuing system model defines the blocking performance. For an average offered load of Erlangs and n trunks, the following formula gives the probability that the system blocks a call attempt:

$$B(n,a) = \frac{a^n/n!}{\sum_{k=0}^n a^k/k!}$$

Typically, texts call this formula the *Erlang-B* formula [Kleinrock 75, Cooper 81]. Many older books contain tables for the values of the Erlang-B (lost calls cleared) probability.

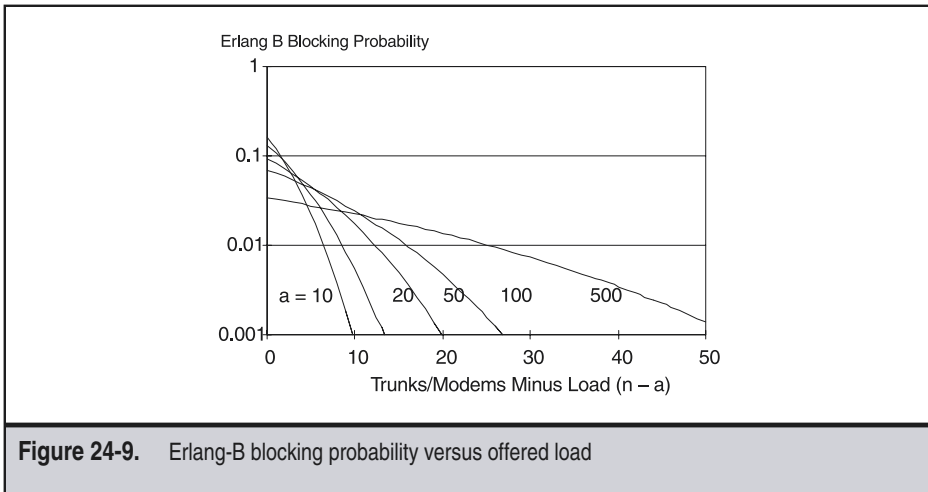
However, you can easily compute your own using the following simple recursion [Cooper 81] in a spreadsheet:

$$B(n+1, a) = \frac{aB(n, a)}{n+1 + aB(n, a)}$$

where $B(0, a) = 1$

This is a useful result to solve the commonly encountered problem of determining the number of trunks (n), given an offered load (a) to meet an objective blocking probability $B(n, a)$. You can either write a spreadsheet macro or define two columns of cells in the spreadsheet. The first column contains the values of n , starting at zero. The second column contains the value 1 in the first row corresponding to $B(0, a)$. All subsequent rows contain the preceding formula for $B(n+1, a)$, coded to use the preceding row's result, $B(n, a)$, and the average offered load a .

Figure 24-9 illustrates the result of using this recursive method of calculation for some values of the Erlang-B blocking probability for various values of offered load (a) versus the difference between the number of trunks and the offered load ($n - a$). Note how the blocking probability decreases more rapidly when adding trunks to serve smaller loads than it does when adding trunks to serve larger loads. Intuitively, we expect this result, since larger systems achieve an economy of scale.



One way of expressing the economy of scale is to compare the percentage of additional trunking required to achieve a fixed blocking probability across a range of offered loads. We define this measure as the *overtrunking ratio*, which is simply the required trunks divided by the average offered load, namely, n/a . For example, an overtrunking ratio of 100 percent means that the number of trunks exactly equals the average offered load yielding a particular blocking probability. For typical blocking probabilities in the range of 0.1 percent to 1 percent, the overtrunking ratio ranges between 110 percent and 120 percent for large values of offered load. Figure 24-10 illustrates the overtrunking ratio versus the average offered load for a set of representative values of blocking probabilities. Many commercial voice networks are engineered for a blocking probability in the range of 0.1 to 1 percent. Also shown in Figure 24-10 is the square root approximation, $1 + 1/\sqrt{a}$, which approximates the overtrunking ratio reasonably well for blocking probabilities on the order of a few percent.

Erlang's Blocked Calls Held Formula

A *blocked calls held* switching system has Markovian call arrivals, Markovian call duration and n servers (e.g., operators), and an infinite number of waiting positions. Therefore, an $M/M/n$ queuing system model defines the performance. For an average offered load of a

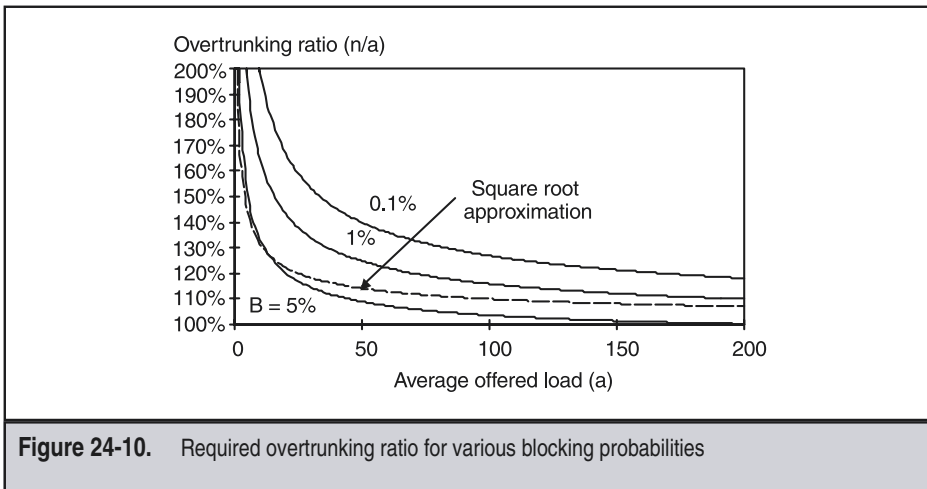


Figure 24-10. Required overtrunking ratio for various blocking probabilities

Erlangs and n servers, the following formula gives the probability that the system queues a call attempt.

$$C(n,a) = \frac{\omega}{\sum_{k=0}^{n-1} \frac{(na)^k}{k!} + \omega}$$

$$\text{where } \omega = \left(\frac{(na)^n}{n!} \right) \left(\frac{1}{1-a} \right)$$

Typically, texts call this formula the *Erlang-C* formula [Kleinrock 75, Cooper 81]. Many older books contain tables and plots for the values of the Erlang-C (lost calls held) probability. However, you can easily compute your own using the following simple formula [Cooper 81] utilizing the Erlang-B recursion defined in the previous section.

$$C(n,a) = \frac{nB(n,a)}{n - a[1 - B(n,a)]}$$

Figure 24-11 illustrates some values of the Erlang-C formula for server group sizes in excess of the average load ($n - a$) as a function of offered load (a). A similar trend exists to that of blocking probability when comparing the queuing probability of a system with a smaller offered load to a system with a larger offered load. The fact that the queuing probability decreases much more slowly with each additional server for larger systems than it does for smaller ones results in an economy of scale similar to that of blocking systems.

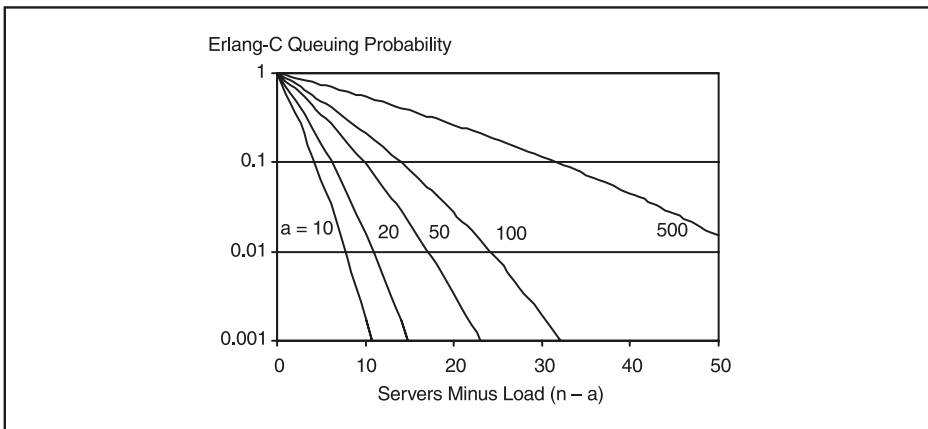


Figure 24-11. Erlang-C queuing probability versus offered load

PERFORMANCE OF BUFFERING METHODS

This section analyzes several simple models of switch delay and loss performance determined by the switch buffer architecture. For simplicity, the text assumes Poisson arrivals and negative exponential service times as the traffic model.

Input Versus Output Queuing Performance

Recall from Chapter 16 the basic types of switch buffering: input, internal, and output. Output queuing delay performance behaves as a classical M/M/1 system. However, input queuing incurs a problem known as head of line (HOL) blocking. HOL blocking occurs when the cell at the head of the input queue cannot enter the switch matrix because the cell at the head of another queue is traversing the matrix.

For uniformly distributed traffic with random message lengths, the maximum supportable offered load for input queuing is limited to 50 percent [McDysan 89], while fixed message lengths increase the supportable offered load to only about 58 percent [Hui 87], as shown in Figure 24-12. On the other hand, output queuing is *not* limited by utilization as in input queuing.

Figure 24-12 illustrates this result by plotting average delay versus offered load for input and output queuing. For a more detailed analysis of input versus output queuing, see [Hluchyj 88], which shows that these simple types of models are valid for switches with a large number of ports.

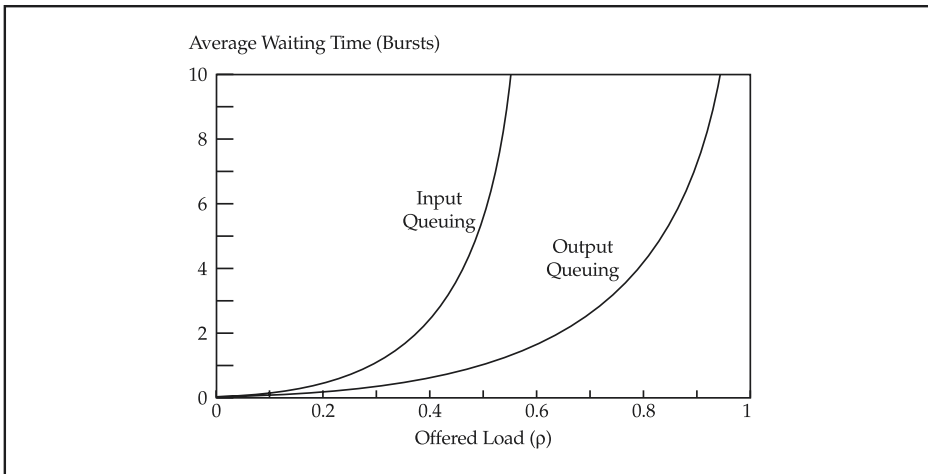


Figure 24-12. Delay versus load performance for input and output queuing

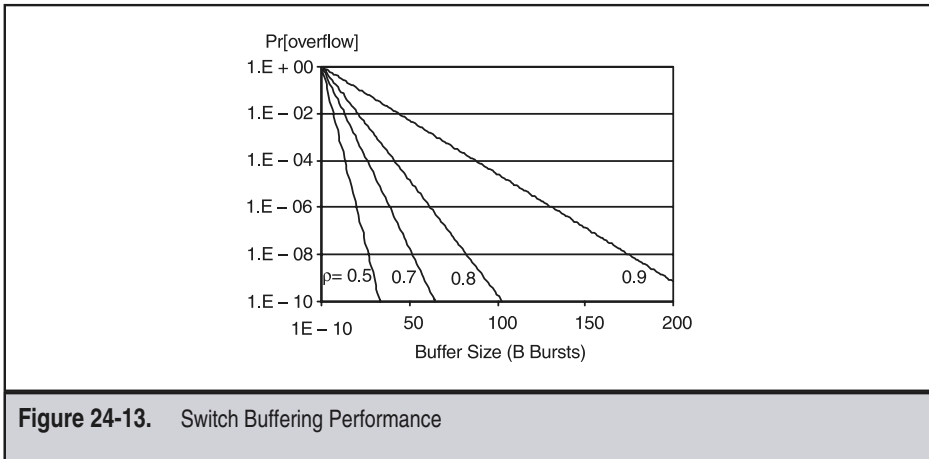
The consequence of this result is that almost all ATM and MPLS switches have some form of output buffering. Modern switches use input buffering in conjunction with some means to address head of line (HOL) blocking. Examples of methods to address HOL blocking involve schemes where other cells or packets pass an HOL-blocked cell or packet. For example, a switch may have one input queue for each destination port, one for each destination port and service category, or even one queue per virtual connection.

Output Buffer Overflow Probability

This section gives a simple approximation for the probability that traffic overflows a buffer. The analysis models the router or switch buffer as an M/M/1/B queuing system that has a finite buffer capable of holding B bursts of negative exponentially distributed length. For simplicity, the analysis assumes an M/M/1 queuing system, which has an infinite buffer, instead of a M/M/1/B system, which has a finite buffer of B cell positions. The overflow probability for a buffer of size B cells with input bursts containing P cells on average is approximately the probability B / P bursts exist in the queuing system with infinite capacity. In modern switches and routers, buffers with effectively infinite capacity are widely available, and therefore this is a valid practical model. Comparison of simulation results and exact analysis has shown this to be a reasonable approximation [Schwartz 77]. The exact and approximate buffer overflow probabilities using this method are

$$\text{Pr}[\text{overflow}] \equiv \varepsilon = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}} \approx \rho^B$$

The approximation is valid when $\rho^B \ll 1$. Figure 24-13 plots this approximate buffer overflow probability versus buffer size for various levels of offered load ρ . Of course, if



the buffer has dimensions of cells, then you should multiply the x-axis by the average number of cells contained in a burst.

Typically, a network design requires a specific overflow probability objective. A curve of overflow probability versus buffer size varies as a function of the offered load, as shown in Figure 24-14. Note that for a given buffer size B , the overflow probability increases as the offered load ρ increases. This means that there is a maximum offered load that the queuing system operates at while still meeting a specified overflow probability ρ .

We can solve the preceding equation to yield a simple guideline for the required buffer capacity B on a switch or router port serving Markovian traffic to yield a desired overflow probability ϵ . The result is the following:

$$B \approx \frac{\ln(\epsilon)}{\ln(\rho)} \equiv \frac{-\ln(\epsilon)}{1-\rho}$$

The right-hand asymptotic result occurs as ρ approaches 100 percent load as derived from the series expansion for the natural logarithm. We can compare the results for overflow probability ρ and required buffer capacity B for self-similar traffic input using an approximation for Fractional Brownian Motion traffic, as described in [Norros 95]. The probability of overflowing a buffer of size B_{ss} with self-similar traffic input characterized by an offered load ρ , Hurst parameter H is approximately:

$$\epsilon \approx \exp\left[-\frac{1}{2}\left(\frac{1-\rho}{\rho H}\right)^{2H} \left(\frac{B_{ss}}{1-H}\right)^{2-2H}\right]$$

Note that we omitted several factors from the more precise expression reported in [Norros 95] for the sake of simplicity. Solving the preceding equation for the required buffer capacity for self-similar traffic input yields the following result

$$B_{ss} \approx \frac{[-2\ln(\epsilon)]^{2-2H}}{1-H} \left[\frac{\rho H}{1-\rho}\right]^{H/(1-H)}$$

When the Hurst parameter $H = 1/2$, the correlated self-similar process reduces to uncorrelated Brownian motion, which has only short-range dependence, as contrasted with the long-range dependence of a self-similar process. Substituting $H = 1/2$ into the preceding equation yields the following expression for the required buffer capacity:

$$B_{ss} \equiv \frac{-2\ln(\epsilon)\rho}{1-\rho} \text{ as } H \rightarrow \frac{1}{2}$$

Comparing the preceding asymptotic result to the buffer dimensioning formula for a Markovian queuing system as ρ approaches unity, we see that the expressions have the same functional form. Drawing on the preceding analysis, we are now in a position to compare the buffer requirements necessary to meet a particular overflow objective for a Markovian queuing system against that of a queuing system driven by self-similar input traffic characterized by Hurst parameter H . Figure 24-14 plots the buffer capacity required

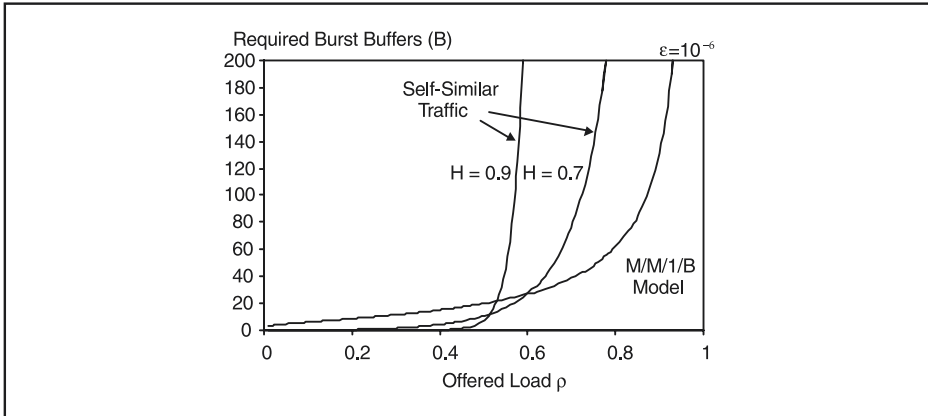


Figure 24-14. Required buffers to achieve a specific overflow probability

to achieve an objective overflow probability for the Markovian system and systems driven by self-similar traffic for $H = 0.7$ and $H = 0.9$ for an overflow probability objective $\epsilon = 10^{-6}$.

This result teaches us several important lessons in traffic engineering. First, characterization of traffic is extremely important for buffer dimensioning. If you are designing an IP or ATM network for traffic best characterized by the self-similar model, the Markovian model will drastically underestimate the required buffer capacity. See [McDysan 00] and [Stallings 98] for practical methods to estimate the self-similar nature on the basis of actual traffic measurements. However, if the traffic in your network employs flow or congestion control, other factors influence the selection of an optimal buffer size, as discussed in the next chapter.

Shared Buffer Performance

The shared buffer scheme exhibits a marked improvement on buffer overflow performance. Since it is unlikely that all ports are congested at the same time, sharing a single, larger buffer between multiple ports is more efficient than statically allocating a portion of the buffer to each port.

The exact analysis of shared buffer performance is somewhat complicated [Hluchyj 88]; therefore, we present a simple approximation based on the normal distribution. In the shared buffer architecture, N switch ports share the common buffer, each having the $M/M/1$ probability distribution requirement on buffer space as defined earlier. The sum of the individual port demands determines the shared buffer probability distribution. The normal distribution approximates a sum of such random variables for larger values of N . The mean and variance of the normal approximation are then given by the following:

$$\text{Mean} = \frac{N\rho}{1-\rho}, \text{ Variance} = \frac{N\rho}{(1-\rho)^2}$$

Figure 24-15 shows a plot of the overflow probability versus the equivalent buffer size per port for shared buffers on switches of increasing port size (N), along with the dedicated output buffer performance for large N from Figure 24-12, for comparison purposes. The offered load is $\rho = 0.9$, or 90 percent average occupancy. The total buffer capacity on a shared buffer switch is N times the buffer capacity on the x-axis. Note that as N increases, the capacity required per port approaches a constant value. This illustrates the theoretical efficiency of shared buffering.

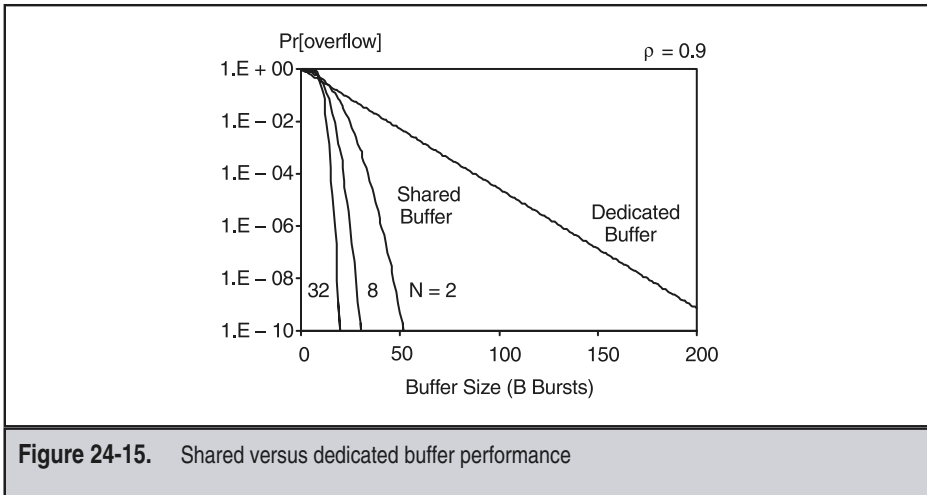


Figure 24-15. Shared versus dedicated buffer performance

DETERMINISTIC CONSTANT RATE PERFORMANCE

The accurate loss performance measure of another very important traffic type, constant bit rate (CBR), turns out to be relatively easy to calculate. Figure 24-16 illustrates the basic traffic source model. N identical sources emit a cell once every T seconds, each beginning transmission at some random phase in the interval $(0, T)$. Thus, even if the network allocates capacity according to the peak rate, the random phasing of the constant rate sources can still create overflow. This is most likely to occur in an IP router or ATM/MPLS switch that has a large number of interfaces that can have simultaneous arrivals. A good approximation for the cell loss rate for such a randomly phased CBR traffic input is [Dron 91]:

$$\varepsilon \equiv \text{Prob}[\text{Loss}] \approx \exp[-2B^2/n - 2B(1-\rho)]$$

where n is the number of constant rate connections,

B is the buffer capacity (in units of fixed length bursts),

and $\rho = nT$ is the offered load.

A closed-form solution for the number of buffers required to achieve a specified loss probability results by solving the preceding formula for the minimum buffer size as follows [Wernik 92]:

$$B_{\text{req}} \approx \frac{\sqrt{[n(1-\rho)]^2 - 2n \ln(\varepsilon)} - n(1-\rho)}{2}$$

Figure 24-17 illustrates the results of this calculation by plotting the required buffers B_{req} versus the number of constant rate connections n for various levels of overall throughput, ρ . If the switch or router implements priority queuing, this performance

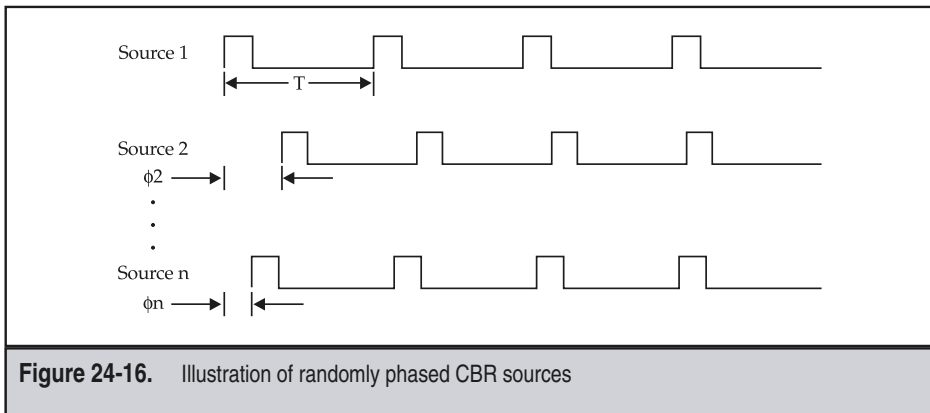


Figure 24-16. Illustration of randomly phased CBR sources

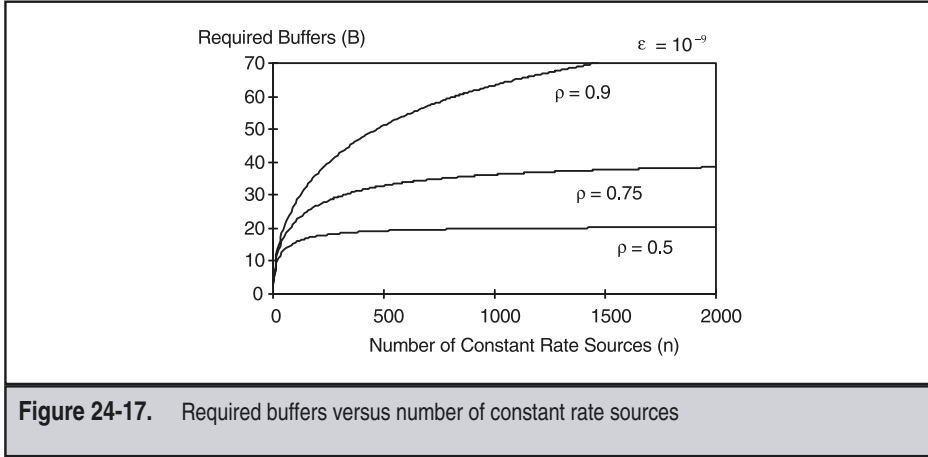


Figure 24-17. Required buffers versus number of constant rate sources

measure can be applied independent of other performance measures. This means that constant rate traffic uses only the fraction ρ of the overall link capacity. The remaining capacity is available for other traffic classes.

The actual delay variation Δ , measured in absolute time, delivered by the switch or router depends upon the packet size β and the link rate R . In general, delay variation decreases in inverse proportion to decreasing link rate. The worst-case delay variation occurs between the extremes of the buffer being entirely empty and completely full. That is, the worst-case bound on delay variation is

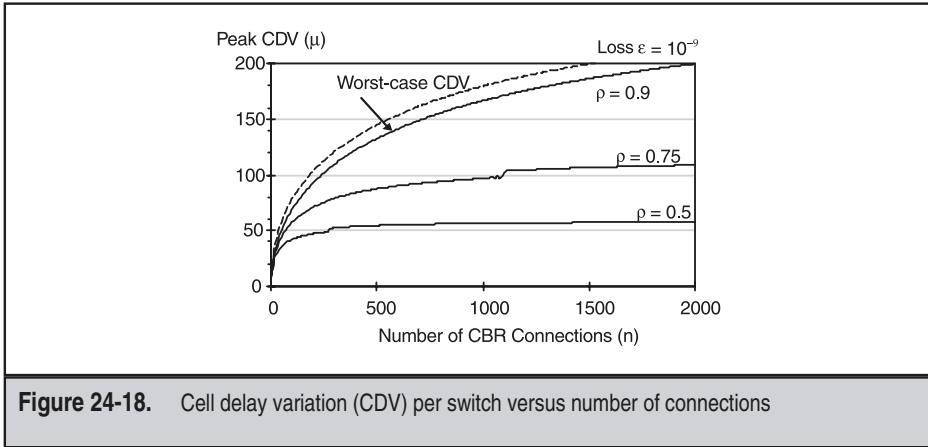
$$\Delta \leq \frac{\beta B_{\text{req}}}{R}$$

A more accurate approximation for delay variation is the difference between the buffer capacity B_{req} and the average number of bursts in the buffer [Dron 91] divided by the link rate, namely,

$$\Delta \approx \frac{\beta}{R} \left\{ B_{\text{req}} - \frac{\sqrt{n}}{2} \frac{1 - \Phi[(1 - \rho)\sqrt{n}]}{\phi[(1 - \rho)\sqrt{n}]} \right\}$$

where ϕ and Φ are the standard normal density and distribution functions, respectively. The NORMDIST function in Microsoft Excel implements the ϕ and Φ functions in the preceding formula.

Given the preceding solution for the required number of buffers B_{req} , we can compute the delay variation Δ introduced by each node. For example, Figure 24-18 plots the resulting cell delay variation (CDV) on a 149.76 Mbps OC3/STM-1 link versus the number of constant bit rate (CBR) ATM connections n . The loss objective is $\epsilon = 10^{-9}$, and the overall link utilization



ρ is the parameter. For large values of utilization ρ , the approximation and the worst-case bound yield comparable results. However, for lower values of utilization, the approximation predicts a much smaller CDV than the worst-case bound does. The next chapter provides a simpler estimate of end-to-end CDV across an ATM network.

EQUIVALENT CAPACITY

Equivalent capacity is a relatively simple analytical technique that models capacity requirements for variable-rate traffic sources [Guerin 91, Schwartz 96]. This model approximates the exact solution by combining two separate models, applicable for different regions of operation. For systems with a small number of sources, a fluid flow model considers each source in isolation, allocating a fixed amount of capacity to each source. Unfortunately, for larger systems, the fluid flow model overestimates the required capacity because it fails to account for statistical multiplexing gain. In the case of larger systems, a model based upon the normal distribution accurately models the statistical benefits of capacity shared by a large number of sources. The equivalent capacity model is then the minimum of the capacity determined by the fluid flow model and the statistical multiplexing gain model. This section first covers the fluid flow model, followed by the normal distribution-based statistical multiplexing gain model, and concludes with the equivalent capacity model.

This section uses a model with N identical traffic sources characterized by the following parameters:

- ▼ Peak rate P (bps)
- Average rate A (bps)
- ▲ Average burst size β (bits)

The model is readily extended to a mix of different source types using the techniques described in References [Guerin 91] and [Ahmad 95].

Fluid Flow Approximation

The fluid flow model treats each source independently, reserving bandwidth and buffer capacity to meet a specified loss probability. Conceptually, the model treats each source as a continuous-time two-state Markov process, each being in either an active (on) or an idle (off) state. In the active state, the source generates a burst of cells at a rate of P bps, with each burst containing an average of β bits. The model assumes that the switch or router allocates B bits in a buffer for each source. Figure 24-19 illustrates the concept of the fluid flow model.

The following formula for the fluid flow equivalent capacity depends largely upon the utilization of the source, $\rho = A / P$, and the ratio of the average burst size to the buffer capacity:

$$C_f = P \frac{z - 1 + \sqrt{(z - 1)^2 + 4\rho z}}{2z}$$

$$\text{where } z = -\ln(\epsilon) \frac{\beta}{B}$$

Figure 24-20 plots the normalized fluid flow capacity (i.e., C_f divided by the peak rate P) versus the ratio of average burst size to buffer capacity, β/B , with the average source utilization $\rho = A/P$ as the parameter. This normalized fluid flow equivalent capacity is the fraction of the actual source peak rate P required to achieve the required loss probability ϵ . Note that when the burst is very small with respect to the buffer (i.e., $\beta/B = 0.001$), then the required equivalent capacity approaches the average source utilization ρ . However,

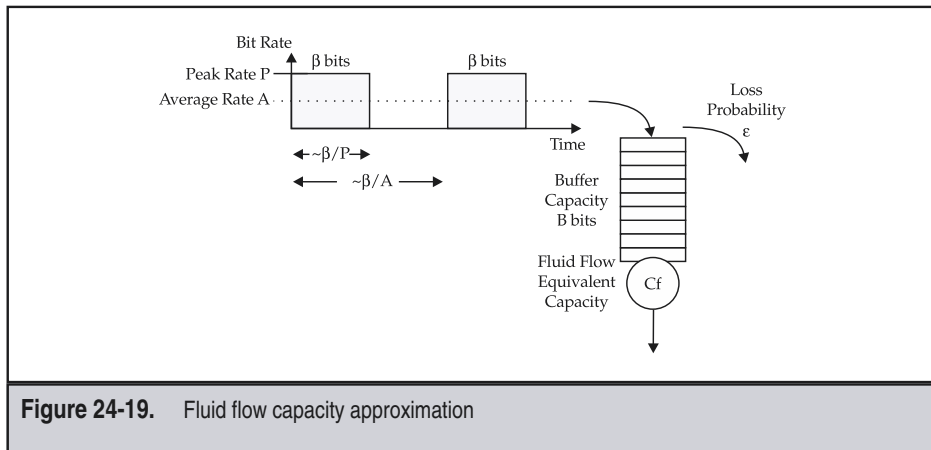
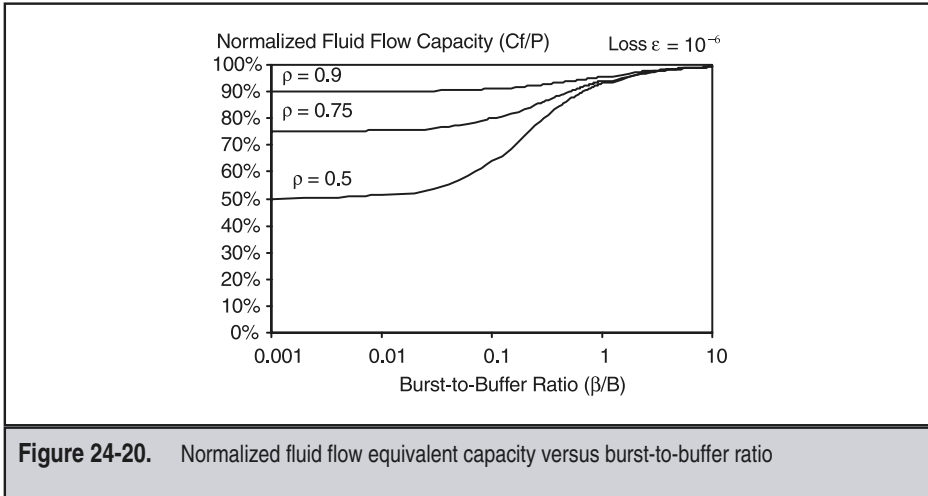


Figure 24-19. Fluid flow capacity approximation



as the size of the burst increases with respect to the buffer capacity (i.e., $\beta / B > 1$) then the required equivalent capacity approaches 100 percent of the peak rate. Thus, if your application can tolerate wide variations in delay caused by large buffers, then you can run a network quite efficiently. However, if you're integrating voice and video along with this data, your network needs to do more than buffering.

Statistical Multiplex Gain Model

Packet and cell switching enables statistical multiplexing, which exploits the on-off, bursty nature of many source types, as illustrated in Figure 24-21. In the left-hand side of the figure, $N = 4$ sources generate bursts of data characterized by a peak rate P and an average rate A . A statistical multiplexer combines these inputs, resulting in the multiplexed output stream shown in the right-hand side of the figure. In this simple example, the aggregate traffic normally requires only two channels at any point in time. The statistical multiplexer either discards or buffers bursts in excess of the link rate L , as shown in the figure. As the multiplexer combines more sources together, the statistics of this composite sum become increasingly more predictable. The statistical multiplexing gain G is the ratio of channels supported to the number of required channels, as indicated in the following formula:

$$G = \frac{\text{Number of Sources Supported}}{\text{Required Number of Channels}}$$

Note that the gain is never less than one. Furthermore, G is significantly greater than one only if the number of sources supported greatly exceeds the number of channels required by individual sources. The probability density for N identical sources is given by

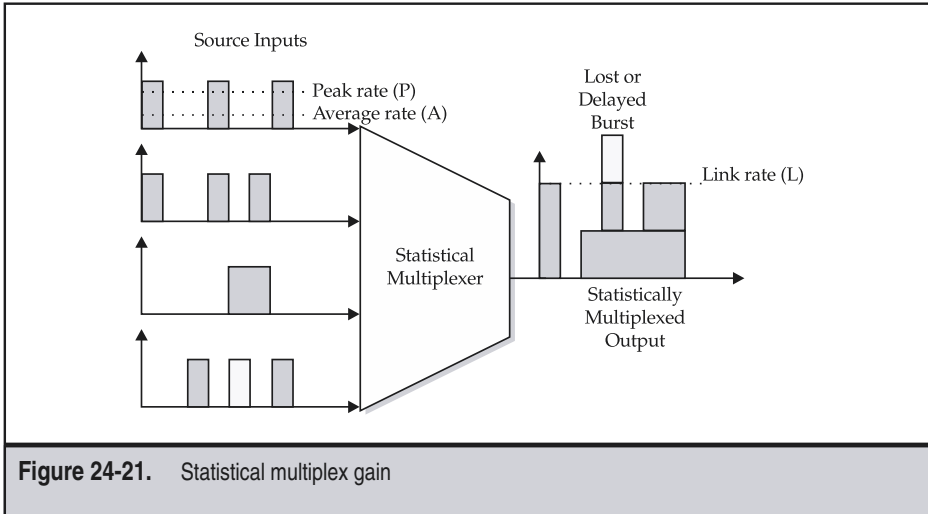


Figure 24-21. Statistical multiplex gain

the binomial distribution and approximated by the normal distribution with the following parameters:

$$\text{Mean} = N\rho \quad \text{variance} = N\rho(1 - \rho)$$

The required number of channels for this Gaussian approximation, expressed in units of the number of peak rate sources, to achieve an objective loss probability $\varepsilon = Q(\alpha)$ is

$$C_g = N\rho + \alpha\sqrt{N\rho(1 - \rho)}$$

where $Q(\alpha)$ is the probability that a zero-mean normal random variable takes on a value greater than α .

Therefore, the statistical multiplexing gain G is the ratio of the capacity required for the number of sources supported N , divided by the required channels C_g . Equivalently, G is the sum of the peak rate for all sources NP divided by the link rate. Equating these expressions, we have

$$G = \frac{N}{C_g} = \frac{NP}{L} = N\eta$$

where $\eta \equiv P/L$ is the peak-to-link-rate ratio.

Setting C_g in the preceding equation equal to the link capacity per source $L/N = 1/\eta$ yields a solution for N using the quadratic formula. Multiplying N by the parameter η results in the following formula for statistical multiplexing gain:

$$G \approx \frac{\eta \left(\sqrt{\alpha^2 (1 - \rho) + 4/\eta} - \alpha\sqrt{1 - \rho} \right)^2}{4\rho}$$

Figure 24-22 plots the achievable statistical multiplexing gain G versus the peak-to-link ratio η with burstiness $b = P / A = 1 / \rho$ as a parameter for a loss of $\epsilon = 10^{-6}$. This figure illustrates the classical wisdom of statistical multiplexing. The ratio of any individual source with respect to the link rate η should be low, and the burstiness of the sources b must be high (or, equivalently, the source activity ρ must be low) in order to achieve a high statistical multiplexing gain G .

Note that the statistical multiplexing gain G never exceeds the source burstiness b , since $b = P / A$ is the maximum gain if only the average rate A were required for each source. Another way of looking at the performance is to consider the fraction of the link utilized on average. The link utilization U is the average rate A generated by N sources divided by the link rate L . We can rewrite U using the definition of G and recalling that the burstiness $b = P / A = 1 / \rho$ as follows:

$$U = \frac{NA}{L} = \frac{G}{b} = G\rho$$

Figure 24-23 plots the utilization U achieved for the same parameters as in Figure 24-22 with burstiness $b = 1/\rho$ as a parameter. Note that the system achieves high link utilization only when the peak-to-link ratio, $\eta = P/L$, is very small. The reduction in statistical multiplexing gain results in a correspondingly lower utilization as the peak-to-link ratio increases, as expected.

Implicit in the preceding expressions for statistical multiplexing gain and the associated utilization is that a certain number of sources N are multiplexed together. The required number of sources N from the preceding equations is simply G / η . Figure 24-24 illustrates the number of sources N that must be multiplexed together to achieve the statistical multiplexing gain and utilization predicted in the previous charts. This plot con-

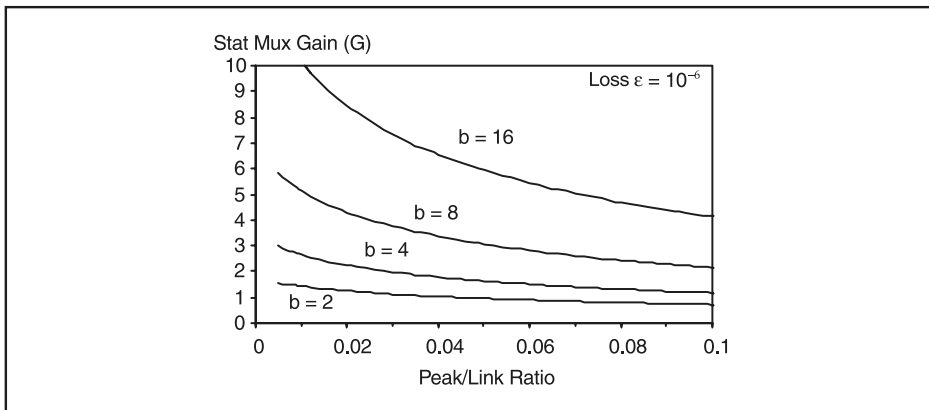


Figure 24-22. Achievable statistical multiplex gain

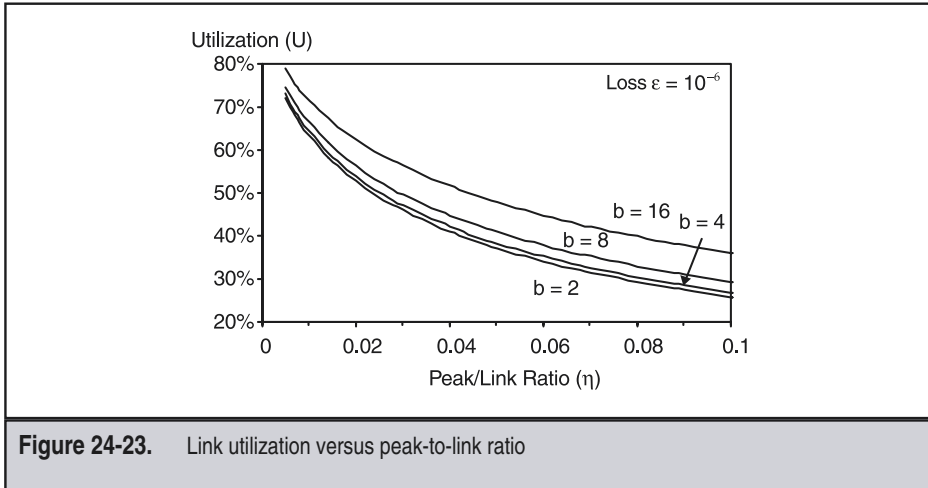


Figure 24-23. Link utilization versus peak-to-link ratio

firmly the applicability of the following statistical multiplexing gain assumption. A large number of sources N with high burstiness b (or, equivalently, low source utilization ρ), along with a modest peak-to-link rate ratio η , must be multiplexed together in order to achieve a significant amount of statistical multiplexing gain G .

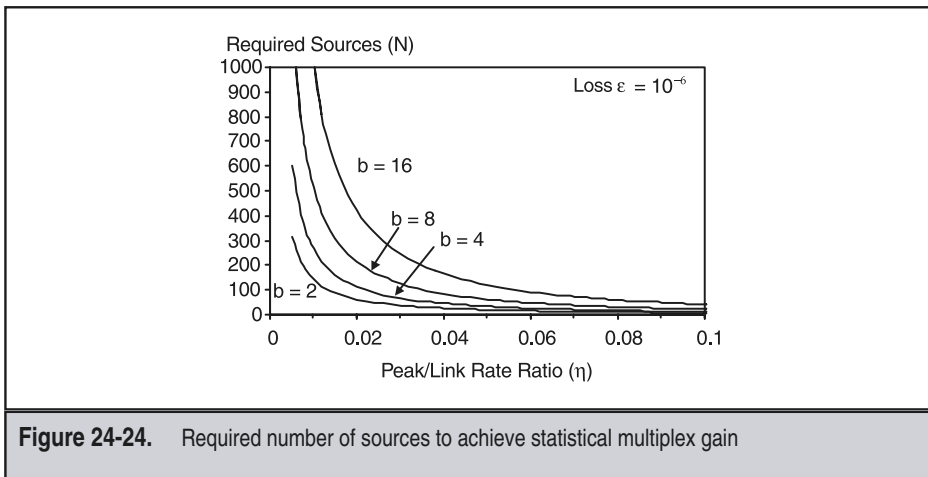


Figure 24-24. Required number of sources to achieve statistical multiplex gain

Equivalent Capacity Approximation

The preceding sections gave approximations that made different predictions. The fluid flow approximation predicts efficient operation when the burst size is much less than the buffer size. On the other hand, the statistical multiplexing model predicts higher efficiency when the source peak rate is much smaller than the link rate for a large number of sources. The equivalent capacity model combines these two models to yield the required equivalent capacity C_e according to the following formula [Guerin 91]:

$$C_e = \text{Min}\{N C_f, C_g P\}$$

where C_f = fluid flow equivalent capacity

C_g = statistical multiplex gain capacity

Although the equivalent capacity approximation overestimates the required capacity, its computational simplicity makes it an attractive model to understand how the parameters of variable-rate traffic sources affect overall utilization and efficiency. Let's look at how these two approximations combine versus the parameters of peak-to-link ratio η and burst-to-buffer ratio β / B through several examples. All of these examples numerically compute the maximum number of identical sources that yield a loss of $\epsilon = 10^{-6}$.

Figure 24-25 plots utilization U versus the ratio of source peak rate to link speed $\eta = P/L$ for an average source burst size to switch buffer size ratio β/B of 10 percent with burstiness $b = P/A$ as the parameter. This figure illustrates the point that the source peak

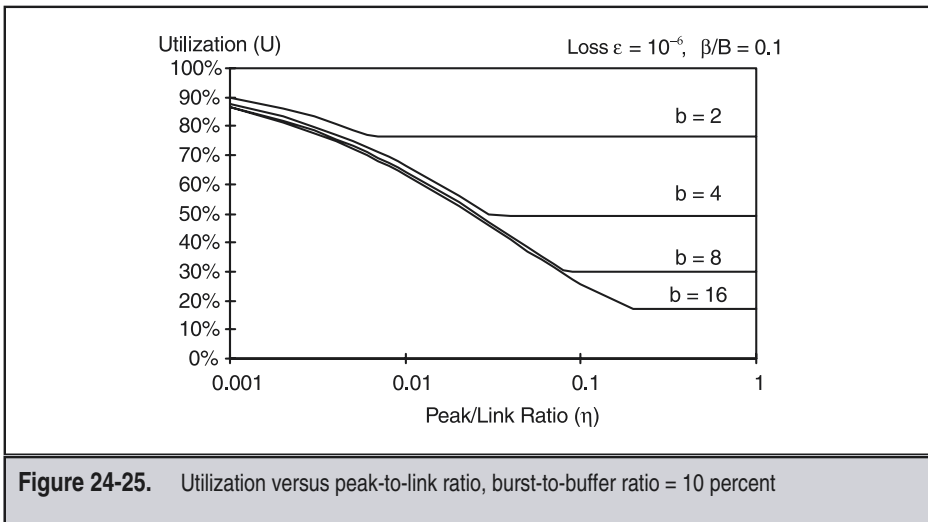


Figure 24-25. Utilization versus peak-to-link ratio, burst-to-buffer ratio = 10 percent

rate should be much less than the link rate when multiplexing traffic with bursty traffic (i.e., larger values of b). This occurs because burstier traffic has a higher variance in the Gaussian model, and hence has a higher probability of overflowing the buffer capacity. As the peak-to-link ratio η increases, the asymptotic statistical multiplexing gain places a lower limit on utilization.

Figure 24-26 plots Utilization U versus the ratio of source peak rate to link speed for the same parameters, but now the average source burst size to switch buffer size ratio β / B is 1 percent. In other words, the buffers employed in the router or switch are ten times larger than those utilized in the previous chart. Note that the utilization is now independent of the source peak-to-link ratio, since the fluid flow term of equivalent capacity now dominates. Intuitively, the larger buffer smoothes out short-term fluctuations in source activity for the statistical multiplexing term of equivalent capacity. Adaptive flow control, such as IP's slow-start TCP protocol or ATM's Available Bit Rate (ABR), would also achieve effects similar to the large buffer modeled in this example. This analysis stresses the importance of large buffers and/or small burst sizes in order to achieve high link utilization.

Figure 24-27 illustrates the effect of the ratio of average source burst size to switch buffer size β/B on utilization U . The PCR divided by the link rate η is now the parameter. The average source activity ρ is 50 percent in this example. For smaller values of burst-to-buffer ratio, the utilization is independent of the peak-to-link ratio. However, for larger values of burst-to-buffer ratio, the utilization approaches that of peak rate allocation. This graphic clearly illustrates the benefits of larger buffers, particularly when the peak-to-link ratio is larger than a few percent.

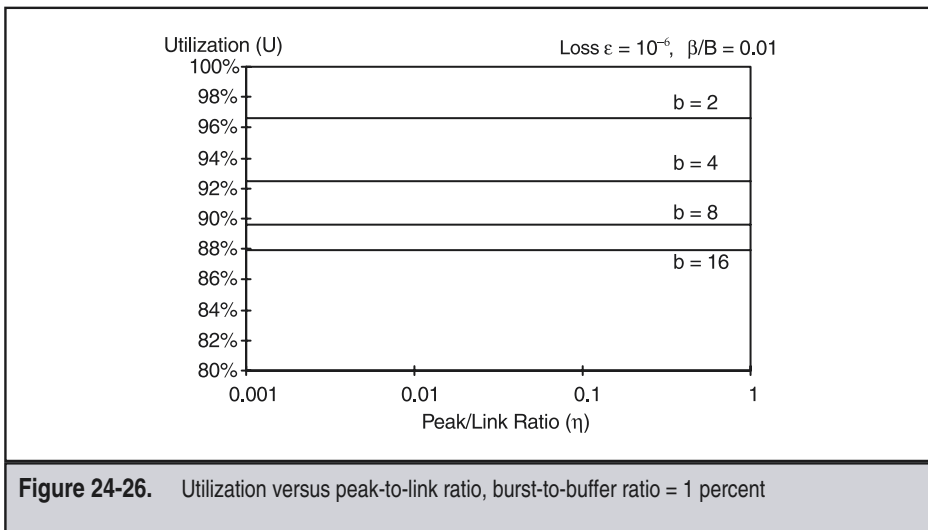
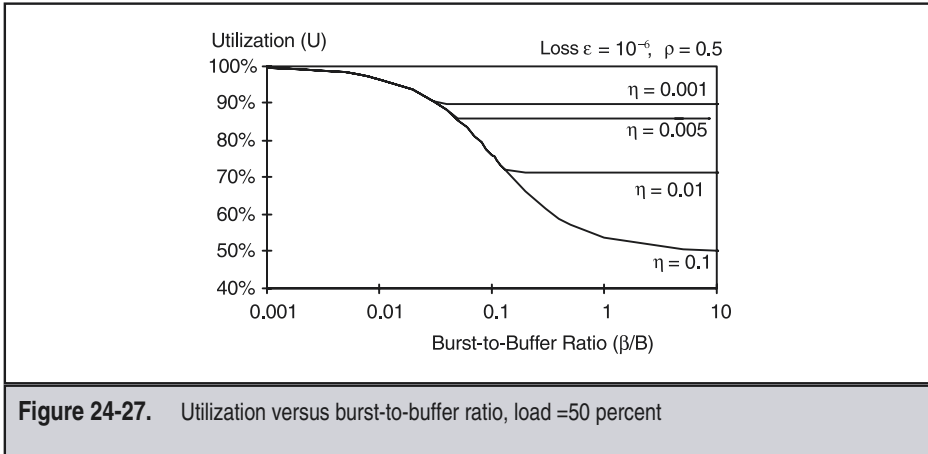
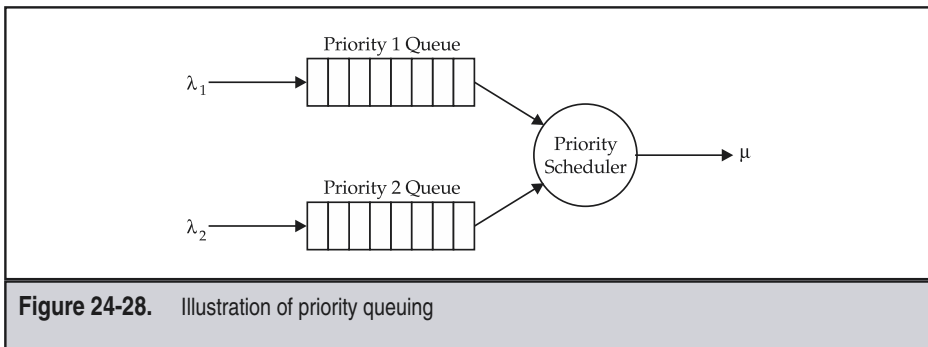


Figure 24-26. Utilization versus peak-to-link ratio, burst-to-buffer ratio = 1 percent



PRIORITY QUEUING PERFORMANCE

This section illustrates the capability of priority queuing to provide different delay (or loss) performance for two classes of traffic. Figure 24-28 illustrates traffic originating from two sources arriving at separate queues served by a priority scheduler with service rate μ . High-priority traffic arrives at rate λ_1 , while low-priority traffic arrives at rate λ_2 according to the usual numbering convention for priorities, where the smaller number indicates higher priority. The priority scheduler services priority 1 traffic first and only services priority 2 traffic when there is no priority 1 traffic. The priority scheduler completes service of any packet or cell before processing the next packet or cell. In other words, service is nonpreemptive. In the following analysis, we assume that the queues are of infinite length. In a real switch or router, of course, the queues may overflow and create loss.



The following analysis assumes that priority 1 and 2 traffic have a Poisson arrival rate of λ_1 and λ_2 bursts per second, respectively. All traffic has a negative exponentially distributed service time with mean μ^{-1} . The net effect of priority queuing is that the priority 1 traffic observes a queue occupancy that is almost independent of the priority 2 traffic load. The only effect that priority 1 traffic sees is the potential of an arriving priority 1 packet having to wait until a priority 2 packet completes service. On the other hand, the priority 2 traffic sees delay as if the transmission capacity were reduced by the average utilization taken by the priority 1 traffic. The formulas for the average waiting time of priority 1 and priority 2 traffic in queue are [Gross 85]

$$W_{q1} = E[\text{Priority 1 Burst in Queue}] = \frac{\rho/\mu}{1-\rho_1}$$

$$W_{q2} = E[\text{Priority 2 Bursts in Queue}] = \frac{\rho/\mu}{(1-\rho)(1-\rho_1)}$$

where $\rho_1 = \lambda_1 / \mu$ and $\rho_2 = \lambda_2 / \mu$ are the offered loads for priority 1 and 2 traffic, respectively, and $\rho = \rho_1 + \rho_2$ is the total offered load to the system. The average waiting time is $W_q = E[q] / \lambda$. Of course, the average waiting time for the system without any priority is equal to the weighted average of the waiting time for priority 1 and 2 users in the system as follows:

$$W_q = \frac{\rho/\mu}{1-\rho} = \frac{\rho_1 W_{q1} + \rho_2 W_{q2}}{\rho}$$

Figure 24-29 illustrates the effect of priority queuing by plotting the average waiting time for a single-priority system according to the M/M/1 model against the average waiting time for priority 1 and priority 2 users. The figure plots the waiting times W_q normalized by multiplying by μ . This example assumes that 50 percent of the traffic is priority 1,

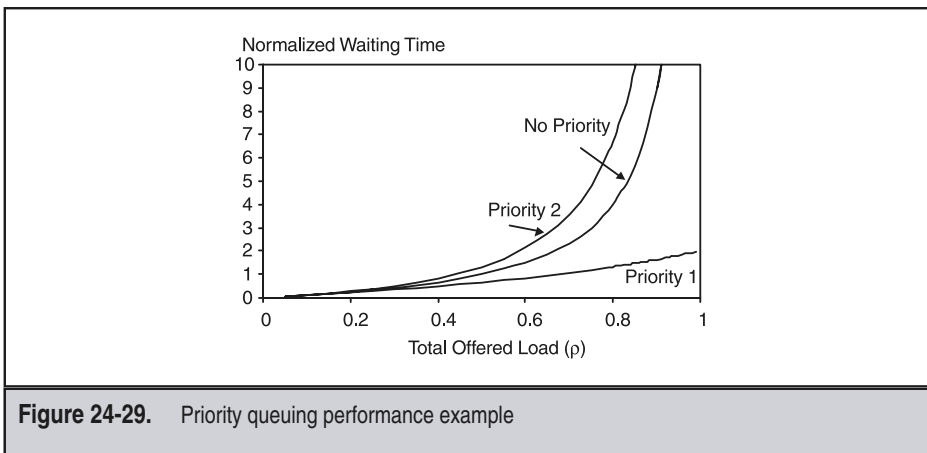


Figure 24-29. Priority queuing performance example

that is, $\rho_1 / \rho = 0.5$. Observe that the priority 1 performance is markedly better than the single-priority system, while priority 2 performance degrades only slightly as compared to the system with no priority. Thus, priority queuing provides an important means to deliver different levels of QoS to different traffic classes. When the high-priority traffic becomes a significant portion of the total traffic, the benefit of priority queuing diminishes.

REVIEW

This chapter first discussed several key aspects of traffic engineering philosophy: source modeling, performance measurement, and switch performance modeling. These affect the accuracy and complexity of the traffic engineering calculations. In general, the more accurate the model, the more complicated the calculation. This book opts for simplicity in modeling and introduces only a few of the popular, simple source models. Next, the text covers key aspects of modeling switch performance: call blocking and queuing performance, a comparison of buffering methods, constant rate source performance, a fluid flow approximation, statistical multiplexing, equivalent capacity, and priority queuing. We demonstrated the superiority of output buffering over input buffering and the increased efficiency of shared buffering over dedicated buffering. The coverage then presented simple formulas to evaluate loss formulas for constant- and variable-rate traffic. The text then introduced the equivalent capacity model consisting of a fluid flow and stationary statistically multiplexed approximation. The fluid flow model considers a single source in isolation, computing the smoothing effect of buffering on transient overloads. The stationary statistical multiplex gain model demonstrates the classic wisdom that in order to achieve gain, there must be a large number of bursty sources, each with a peak rate much less than the link rate. The equivalent capacity model combines these two models, demonstrating key trade-offs in the relationship of burst-to-buffer size and peak-to-link ratio over a wide range of network operating parameters. Finally, the text showed how priority queuing yields markedly improved delay performance for the high-priority traffic over that of a single-priority system.

CHAPTER 25

Design Considerations

This chapter explores important network and application design considerations. First, the text analyzes how delay and loss impact an application's performance. The chapter then covers how the accumulation of delay variation in multiple-hop networks impacts delay variation-sensitive applications, such as video, audio, and real-time interactive application traffic. Next, the coverage continues with a discussion regarding TCP performance considerations. The treatment then analyzes the statistical multiplex gain achievable for voice traffic, along with the savings occurring by integrating voice and data onto a single transmission facility. The chapter then summarizes the important steps involved in the network design and planning process. The discussion includes an overview of the principal functions implemented in modern network design tools.

IMPACTS OF DELAY, LOSS, AND DELAY VARIATION

This section reviews the impacts of delay, loss, and delay variation on applications. Delay impacts the achievable throughput for data applications. Loss impacts the usable throughput for most network and transport layer protocols. Delay variation impacts the performance for applications requiring a guaranteed playback rate, such as voice and video.

Impact of Delay

The human ear-brain combination is sensitive to delay in several scenarios. A common impairment encountered in telephony occurs when a speaker receives an echo of his or her own voice. Indeed, for a round-trip delay of greater than 50 ms, standards require echo cancellation [ITU G.131]. One-way communication like video broadcast (e.g., television) or an audio signal (e.g., radio) can accept relatively long absolute delays. However, delay impedes two-way, interactive communication if the round-trip latency exceeds 150 ms. For example, conducting a voice conversation over a satellite link illustrates the problem with long delays, since the listener can't tell if the speaker has stopped or merely paused, and simultaneous conversation frequently occurs. Combined video and audio is very sensitive to differential delays. Human perception is highly attuned to the correct correlation of audio and video, as is readily apparent when the speech is out of synch with lip movement, as occurs in some foreign language dubbed films.

Somewhat different considerations apply to non-real-time data applications. Two situations occur when a source sends a burst of data at a certain transmission rate (or bandwidth) across a network with a certain delay, or latency: we call these situations bandwidth limited and latency limited. A *bandwidth-limited application* occurs when the receiver begins receiving data before the transmitter completes sending the entire burst. A *latency-limited application* occurs when the transmitter finishes sending the burst of data before the receiver begins receiving any data.

Figure 25-1 illustrates the consequence of sending a burst of length b equal to 100,000 bits (100Kb) at a peak rate of R Mbps across the domestic United States with a one-way propagation delay τ of 30 ms. In other words, it takes 30 ms for the bit stream to

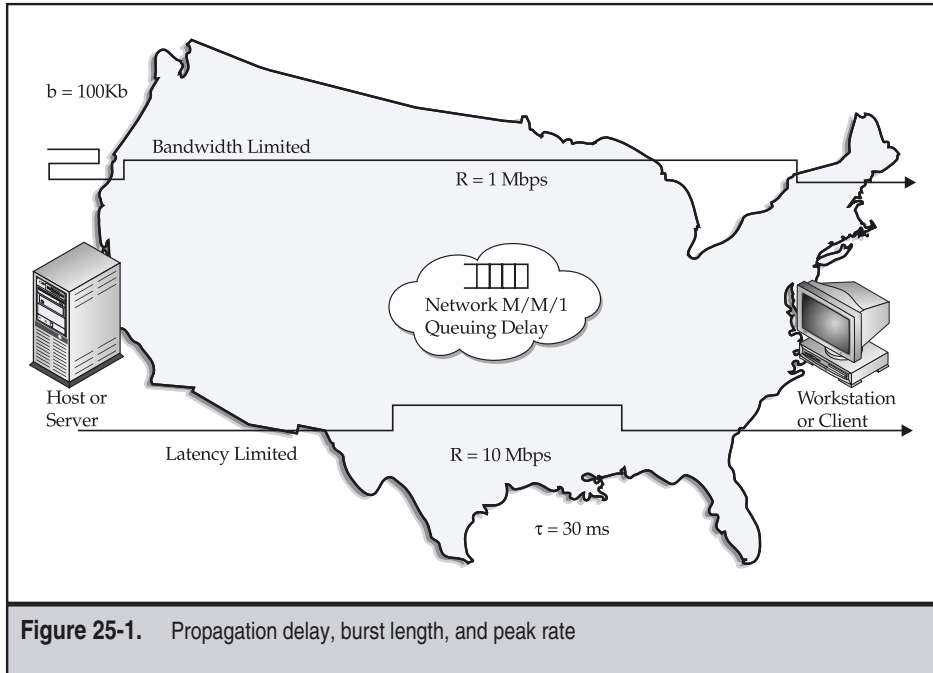


Figure 25-1. Propagation delay, burst length, and peak rate

propagate from the originating station to the receiving station across approximately 4000 miles of fiber, since the speed of light in fiber is less than that in free space, and fiber is usually not routed along the most direct path. When the peak rate between originator and destination is 1 Mbps, and after 30 ms, only about one-third of the burst is in the transmission medium, the remainder is still buffered in the transmitting device. We call this a *bandwidth-limited* application, since the lower transmission rate limits the transmitter from releasing the entire message before the receiver begins receiving the burst.

Now let's look at the *latency limited* case where the transmitter sends the *entire* transmission before the receiver gets any data. Increasing the peak rate to 10 Mbps significantly changes the situation: the workstation sends the entire burst before even the first bit reaches the destination. Indeed, the burst occupies only about one-third of the bits on-the-fly in the fiber transmission system, as indicated in the lower part of Figure 25-1. If the sending terminal must receive a response before sending the next burst, then observe that a significant reduction in throughput results. We call this situation *latency limited*, because the latency of the response from the receiver limits additional transmission of information by the sender awaiting an acknowledgment.

Now let's apply the basic M/M/1 queuing theory from Chapter 24 as an additional element of end-to-end delay that increases nonlinearly with increasing load, and thus becomes a better model of a real-world application. The average M/M/1 queuing plus transmission delay in the network is $b / R / (1 - \rho)$, where ρ is the average trunk utilization in the network. The point where the time to transfer the burst (i.e., the transmission plus queuing time) exactly equals the propagation delay is called the *latency/bandwidth crossover point*, as illustrated in Figure 25-2.

In the previous example, for a file size of $b = 100\text{Kb}$, the crossover point is 3.33 Mbps for zero utilization and increases to 10 Mbps for 66 percent utilization. (See [Kleinrock 92] for a more detailed discussion on this subject.)

If the round-trip time is long with respect to the application window size, then the achievable throughput is markedly reduced. This is a basic aspect of all flow control methods, as seen in Chapter 22 on congestion control. The amount of buffering required in the network to achieve maximum throughput is proportional to the delay-bandwidth product. In our example in Figure 25-2, the round-trip delay-bandwidth product is 60,000 bits (or approximately 8KB) for a 1 Mbps link ($30\text{ ms} \times 1\text{ Mbps} \times 2$) and 600,000 bits for a 10 Mbps link.

This situation is analogous to what occurred in early data communications over satellites, where the data rates were low but the propagation delay was very high. The delay-bandwidth product is high in satellite communications because the propagation delay is large; while in ATM and MPLS communications, the delay-bandwidth product becomes large because the transmission speeds are high.

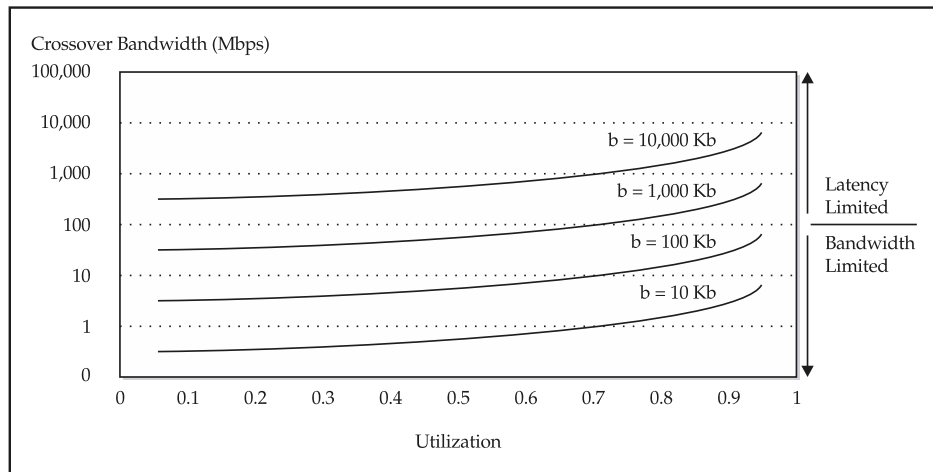


Figure 25-2. Latency/bandwidth crossover point

Impact of Loss

The blink of an eye is approximately one-fiftieth of a second, or 20 ms. When video display devices play back frames at rates of between 25 and 30 frames per second using the image persistence provided by display systems like the cathode ray tube, the human eye-brain perceives continuous motion. When loss or errors disrupt a few frames in succession, the human eye-brain detects a discontinuity, which is subjectively objectionable.

The human ear-brain combination is less sensitive to short dropouts in received speech, being able to accept loss rates ranging from 0.5 percent to 10 percent [Kostas 98], depending upon the type of voice coding employed. This level of loss may cause an infrequent clicking noise, or loss of a syllable. More severe levels of loss can result in the loss of words or even phrases. Although few relationships or business transactions have ended because of low voice quality, a poor communication channel can only make a tense conversation worse. If the price is low enough, however, some users may accept reduced performance to achieve cost savings. For example, voice over IP sometimes does not work at all; however, in some areas of the world, it is still essentially free. Once this economic disparity normalizes, however, a voice service must have acceptable quality to be competitive.

Loss is another enemy of data communications applications. For many applications, the loss of a single cell results in the loss of an entire packet (or an entire protocol data unit) because the SAR sublayer in the AAL fails during the reassembly process. Loss (or even excessive delay) results in a time-out or negative acknowledgment in a higher-layer protocol, typically the transport layer—for example, in the Internet Protocol's Transmission Control Protocol (TCP) studied in Chapter 8.

Higher-layer protocols recover from detected errors, or time-outs, by one of two basic methods: either they retransmit all information sent after the packet that resulted in the detected error or time-out, or they selectively resend only the information that actually was in error or timed out. Resending all of the information means that if the transmitter sent N packets after the packet that caused the detected error or time-out, then the transmitter resends these same N packets again. You'll see this scheme called a *Go-Back-N* retransmission or a cumulative acknowledgment strategy in the technical literature. Obviously, resending the same information twice reduces the usable throughput. The second method requires a higher-layer protocol that explicitly identifies the packet in error. Then, the transmitter resends only the errored packet, improving efficiency when retransmissions occur. The usable throughput increases because only the errored or timed-out information is retransmitted. Standards call this a selective reject, selective retransmission, or selective acknowledgment strategy. However, this type of protocol is more complex to implement. The ATM Service Specific Connection Oriented Protocol (SSCOP) defines a sophisticated selective reject algorithm [ITU Q.2110]. The IETF defines optional selective acknowledgment strategies as extensions to TCP [RFC 1072] in support of communication over long delay paths; however, few implementations support them.

The classical analysis of retransmission protocols assumes that lost packets occur randomly with probability π . Typically, these analyses assume that the source of packet loss

is due to random errors corrupting the received packet. The analysis also applies to networks that lose packets randomly in an uncorrelated manner. Note that the probability that the receiver loses an individual packet π due to a specific cell loss ratio (CLR) in the ATM network is approximately

$$\pi \approx \left[\frac{P}{48} \right] \text{CLR}$$

where P is the packet size in bytes.

Observe that packets transferred from the sender to the receiver experience a one-way delay τ , followed by acknowledgments of these same packets arriving at the receiver τ seconds later. Since it takes $8P/R$ seconds to transmit a P byte packet at R bps, the maximum window of packets (of length P) that can be sent before receiving an acknowledgment is

$$W_{\max} \equiv \frac{\text{Round-trip delay}}{\text{Packet transmit time}} = \frac{2\tau}{8P/R}$$

If a sender's packet transmit window is less than W_{\max} , then throughput must be less than 100 percent, since the sender must stop and wait for acknowledgments before proceeding. The following analysis assumes a greedy sender. That is, the sender continues sending packets up to its fixed unacknowledged packet window size W .

In the Go-Back-N strategy, if a single packet is in error at the beginning of a window of W packets, then the sender must retransmit the entire window of W packets. For the Go-Back-N retransmission strategy, the effective throughput $\eta(\text{Go-Back-N})$ is approximately the inverse of the average number of times the entire window must be resent, which is approximately [Stallings 98, Bertsekas 92]

$$\eta(\text{Go-Back-N}) \approx \begin{cases} \frac{1-\pi}{1+\pi W} & W \geq W_{\max} \\ \frac{W(1-\pi)}{(W_{\max}+1)(1-\pi(W-1))} & W < W_{\max} \end{cases}$$

In the selective-reject retransmission strategy, if a single packet is in error, then the sender retransmits only that packet. For the selective reject retransmission strategy, the usable throughput $\eta(\text{Selective Reject})$ is approximately the inverse of the average number of times any individual packet must be sent, which is

$$\eta(\text{Selective Reject}) \approx \begin{cases} 1-\pi & W \geq W_{\max} \\ \frac{W(1-\pi)}{W_{\max}+1} & W < W_{\max} \end{cases}$$

The preceding formula applies to more-sophisticated protocols that can retransmit multiple packets within a round-trip delay interval, such as ATM's (SSCOP) or TCP selective retransmission used for reliable transport of signaling messages and high-per-

formance satellite links. The factor $W / (W_{\max} + 1)$ in these formulas accounts for the fact that senders with window size W smaller than the value W_{\max} required for maximum throughput experience proportionately lower effective throughput.

Figure 25-3 plots the effective throughput η for Go-Back-N and selective reject retransmission strategies for an OC3/STM-1 rate R of 150 Mbps, a packet size P of 1500 bytes, and a one-way delay τ equal to 30 ms. The resulting maximum window size W_{\max} is approximately 1500 packets. Both retransmission protocols have nearly 100 percent usable throughput up to a packet loss rate of 10^{-5} . As the loss rate increases, however, the effective throughput of the Go-Back-N protocol decreases markedly because the probability that an individual window is error free decreases markedly. As the loss rate increases above 10 percent, the probability of individual packet loss dominates the overall performance, and the selective reject protocol's effective throughput falls off as well.

These examples illustrate the importance of the QoS loss parameter on acceptable effective throughput η . Since most IP Transmission Control Protocol (TCP) implementations use a Go-Back-N type of protocol, low loss rates due to errors are essential to good performance. However, errors are only one source of loss. In fact, on fiber optic networks, loss due to errors is a relatively rare occurrence. A subsequent section examines retransmission strategy performance in the face of a more commonly encountered situation—namely, congestion.

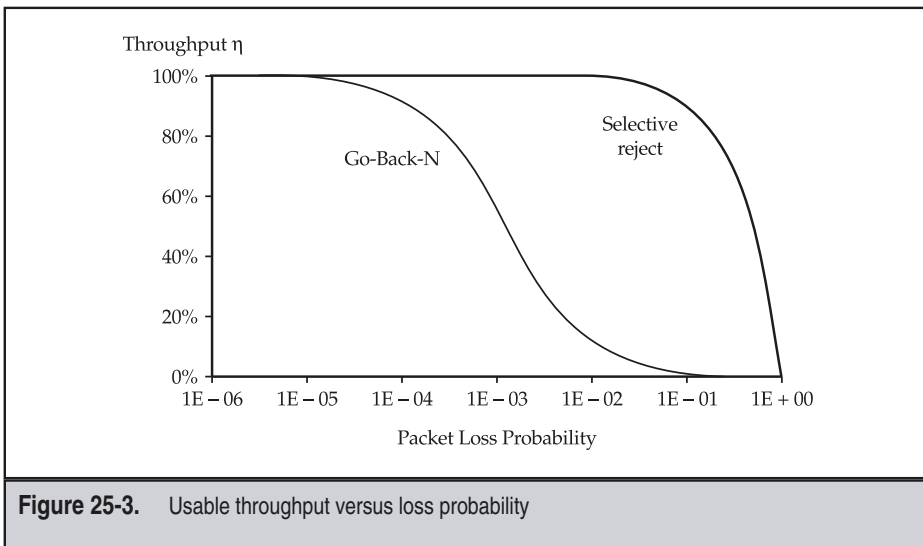


Figure 25-3. Usable throughput versus loss probability

When loss levels are relatively low, a good approximation across an ATM or MPLS network (assuming that loss is independent on each switch and link) is to simply add up the individual loss probabilities.

Impact of Delay Variation

Most audio and video protocols are sensitive to delay variation. Delay variation is a QoS parameter that measures the clumping or dispersion that occurs when multiplexing packets or cells from multiple sources in an end system, switch, or router. The resulting effect accumulates after traversing multiple routers or switches. Streaming audio and video protocols employ a limited playback buffer to account for delay variation. If too little or too much data arrives while the application is playing back the audio or video, then the application either starves for data or overflows the playback buffer. In either case, the net effect from the application's point of view is that of loss, which is subjectively objectionable.

The playback buffer ensures that underrun or overrun events occur infrequently in response to the clumping and dispersion of packets or cells that accrue across a network. In our examples, the nominal cell or packet spacing is four unit intervals, and the playback buffer has space for four cells or packets. In our example, the playback buffer begins operation at a centered position (i.e., holding two cells or packets).

In the overrun scenario depicted in Figure 25-4, cells or packets arrive too closely clumped together, until finally a cell or packet arrives and there is no space in the playback buffer; thus cells or packets are lost. This is a serious event for a video-coded signal because an entire sequence or frame may be lost due to the loss.

In the underrun scenario shown in Figure 25-5, cells or packets arrive too widely dispersed in time, such that when the time arrives for the application to remove the next cell or packet from the playback buffer, the buffer is empty. This, too, has a negative consequence on a video application because the underrun will disrupt continuity of motion or even create the need to resynchronize.

Most audio or video applications also require an accurate clock with which to remove data from the playback buffer and present the signal to the end user. Audio is sensitive to the playback clock, since the human ear perceives variation in the playback rate as a change in pitch, which can affect speaker recognition. In interactive video applications, variations in delay greater than 20–40 ms cause perceivable jerkiness.

A simple approximation for the accumulation of delay variation across multiple switching nodes is the square root rule from the ATM Forum B-ICI specification. This states that the end-to-end delay variation is approximately equal to the delay variation for an individual switch times the square root of the number of switching nodes in the end-to-end connection. The reason that we don't just add the variation per node is that the extremes in variation are unlikely to occur simultaneously and tend to cancel each

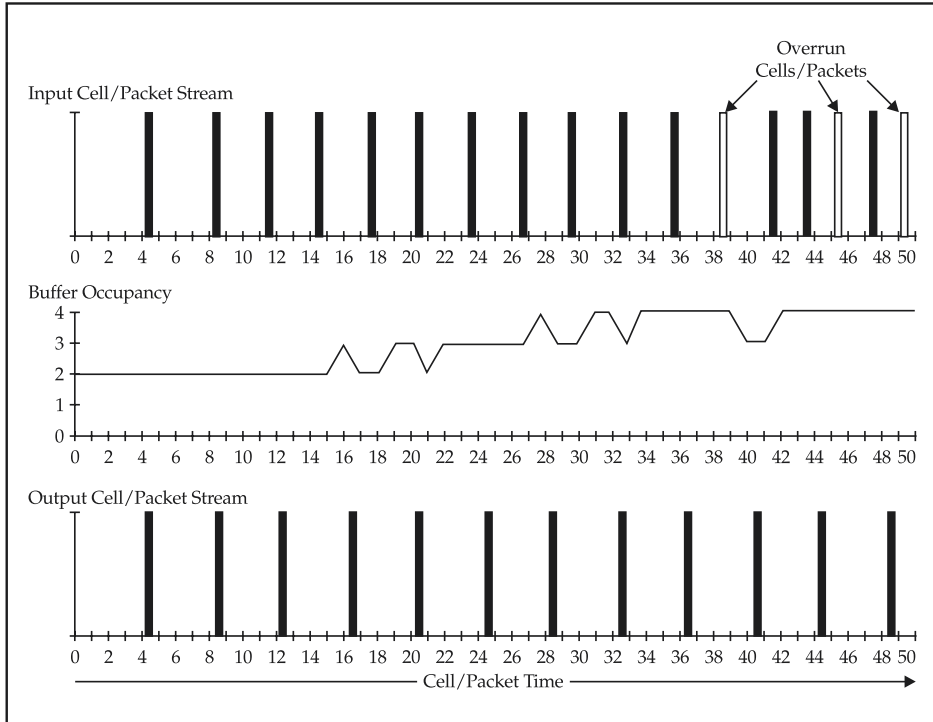


Figure 25-4. Playback buffer overrun scenario

other out somewhat. For example, while the variation is high in one node, it may be low in another node. Figure 25-6 illustrates the concept of accumulated delay variation by showing the probability distribution for delay at various points in a packet-switched network.

The assumed delay distribution has a normally distributed delay with equal mean and standard deviation. Therefore, the average delay added per node is the sum of the fixed and mean delays. Starting from the left-hand side at node A, the traffic has no variation. The first switch/router adds a fixed delay plus a random delay, resulting in a modified distribution shown in the upper left-hand corner of the figure. The next switch/router adds the same constant delay and an independent random delay. The next

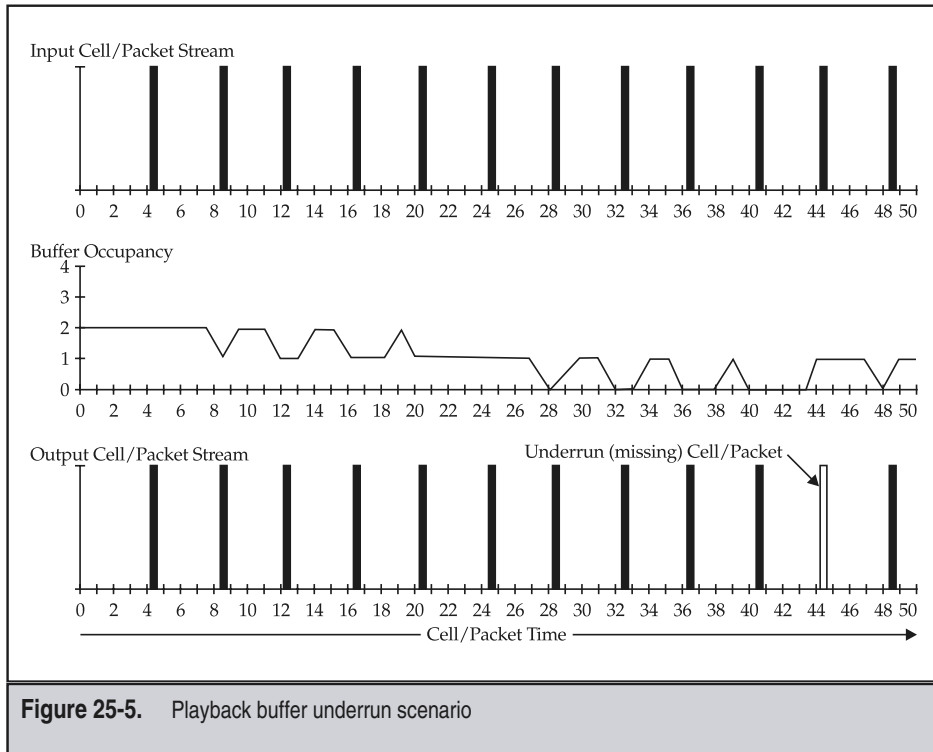


Figure 25-5. Playback buffer underrun scenario

two switches/routers add the same constant and random delay characteristics independently. The resulting delay distribution after traversing four nodes is markedly different, as can be seen from the plots—not four times worse but only approximately twice as bad. Furthermore, correlation between the traffic traversing multiple nodes makes this model overly optimistic, as described in the informative Appendix V of the ATM Forum’s Traffic Management 4.0 specification [ATMF TM 4.0].

Figure 25-7 plots the probability that the delay exceeds a certain value x after traversing N nodes. In this example, the mean and standard deviation of the delay distribution is 100 microseconds. The random delays are additive at each node; however, in the normal distribution, the standard deviation of the sum of normal variables grows only in proportion to the square root of the sum. This equation illustrates the basis for the square root rule of thumb in the ATM Forum B-ICI specification. The sum of the fixed and average delays is 200 μ s, and, therefore, the additional variation is due to the random delay introduced at each node.

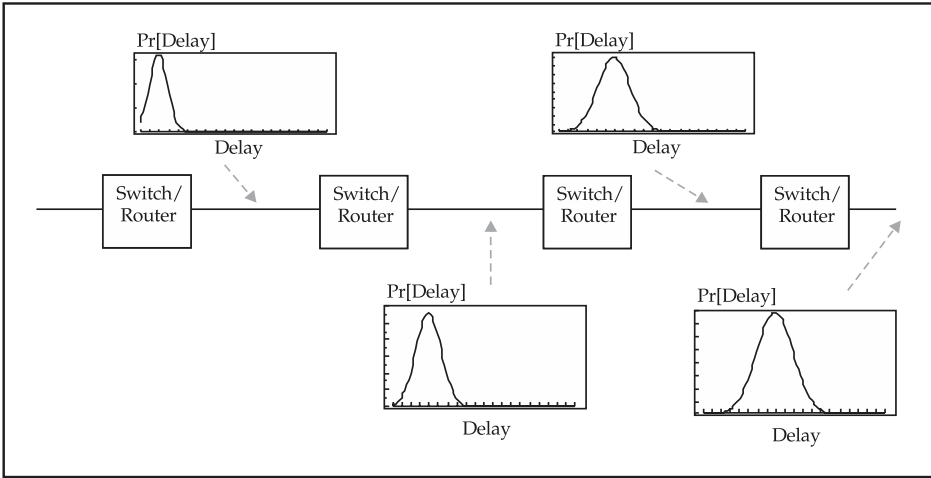


Figure 25-6. Illustration of delay variation in a network

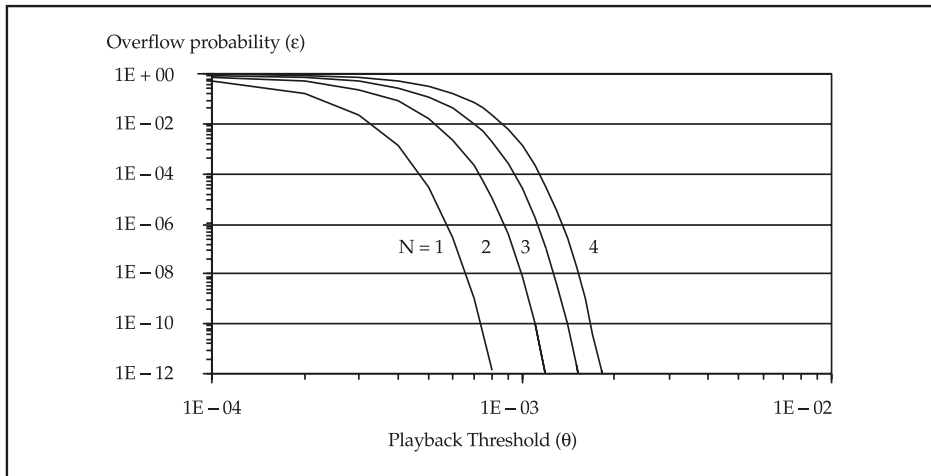


Figure 25-7. Delay probability distribution for a network of N nodes

The equation for the plots of Figure 25-7 for an average plus fixed delay given by α (100 μ s in this example) and a standard deviation σ (100 μ s in this example) for the probability that the delay is greater than x after passing through N switches or routers is

$$\text{Prob}[\text{Delay} \geq x] = Q\left(\frac{x - N\alpha}{\sqrt{N}\sigma}\right)$$

Chapter 23 defined the expression for $Q(x)$ for the normal distribution. The presence of delay variation in IP, MPLS, and ATM networks means that the end equipment must absorb the jitter introduced by intermediate networks. One option to balance this trade-off is the capability to reduce delay variation within switches or routers in the network via shaping, spacing, or framing, as described in Chapter 21.

TCP PERFORMANCE CONSIDERATIONS

This section describes several things to consider when tuning TCP performance. These include the effect of window size on performance, the TCP reaction to periods of congestion, and the interaction with ATM traffic and congestion control.

TCP Window Size Impact on Throughput

Network congestion resulting in loss affects end-to-end transport layer (e.g., TCP) and, ultimately application layer (e.g., HTTP) performance. A key design issue for ATM networks is the selection of service categories and traffic parameters that minimizes loss, and hence maximizes usable throughput performance. Add to this the fact that standard TCP implementations support only a 64KB window size, and you'll see that applications running over TCP can't sustain transmissions at 10 Mbps Ethernet speeds across one-way WAN distances greater than approximately 2500 miles. For applications capable of transmitting at 100 Mbps, the standard TCP window size limits the distance over which maximum throughput is achievable to 250 miles. To achieve higher throughput over networks with large bandwidth-delay products, look for devices that implement TCP window scaling extensions defined in IETF RFC 1323. A general rule of thumb is that the total throughput of TCP (in Kbps or Mbps) increases with the window size in kilobytes linearly up to the point where either the maximum window size is reached, or the window equals the product of the round-trip delay and the bottleneck bandwidth. See Chapter 8 for further details and examples of the TCP windowing protocol.

TCP over ATM: UBR and ABR

This section looks at a few more issues about placing TCP traffic over the ATM service categories called the Unspecified Bit Rate and the Available Bit Rate (UBR and ABR, respectively), detailed in Part 5. TCP/IP is the most common data traffic running over ATM networks today. TCP's adaptive windowing protocol dynamically acts to fill the avail-

able bandwidth, if user demand is high enough. Thus, ATM is an ideal network infrastructure to support TCP's ability to fill in bandwidth unused by higher-priority multimedia traffic like voice and video. TCP can use either the ABR or UBR service category, with advantages and disadvantages as summarized in the text that follows.

Many UBR implementations use a simple cell discard threshold in a buffer based upon the Cell Loss Priority (CLP) bit in the ATM cell header. Once the buffer fills beyond the threshold, the switch discards lower-priority cells (i.e., CLP = 1 tagged UBR cells). ATM switches should have large buffers, sized to be on the order of the product of the round-trip delay and bandwidth bottleneck when supporting TCP over UBR.

During congestion conditions and subsequent loss of cells, the ATM network device does not notify the sender that retransmission is required. Instead, higher-layer protocols—TCP, for instance—must notice the loss via a time-out and retransmit the missing packets. Not only does one cell loss cause the missing packet to be retransmitted, but also all packets after it up to the end of the transmit window. Excessive packet discards within a TCP window can degrade the recovery process and cause host time-outs—causing interruptions on the order of many seconds to minutes. Loss also touches off TCP's slow start adaptive windowing mechanism, further reducing throughput. If you plan to operate TCP/IP over UBR, be sure that your ATM switches or service provider support Early/Partial Packet Discard (EPD/PPD) as defined in Chapter 22. The EPD/PPD functions ensure that the switch discards entire packets during periods of congestion. This is especially important when a relatively large number of TCP sources contend for a particular bottleneck.

Possibly, the best ATM service category for TCP traffic is ABR, which employs a closed-loop, rate-based mechanism for congestion control using explicit feedback. For ABR traffic, the TCP source must control its transmission rate. ABR defines a minimum cell rate (MCR) for each virtual connection, which defines the lowest acceptable value of bandwidth. Note that the MCR may be zero. Operating in an ATM network, the ABR protocol utilizes Resource Management (RM) cells to control the input rate of each source (and thus each connection) based upon the current level of congestion in the switches along the route carrying that connection's traffic. In ABR, the switch buffer size requirements are similar to that in the UBR case.

The network may police ABR connections to ensure that they conform to the traffic contract. The standards allow networks to do this so that an unregulated user on a single connection cannot affect the service of all other users sharing bandwidth with that connection [Pazos 95, Li 96].

TCP/IP Performance in a Congested Scenario

This section defines a simple model for the performance of a number of greedy TCP/IP sessions contending for a common buffer in a switch or router. Figure 25-8 illustrates the scenario where N TCP clients simultaneous transfer data to a single server attached to a switch egress port and buffer. All hosts connect to the switch/router via transmission lines running at R bits per second. A single trunk line also running at R bps empties a

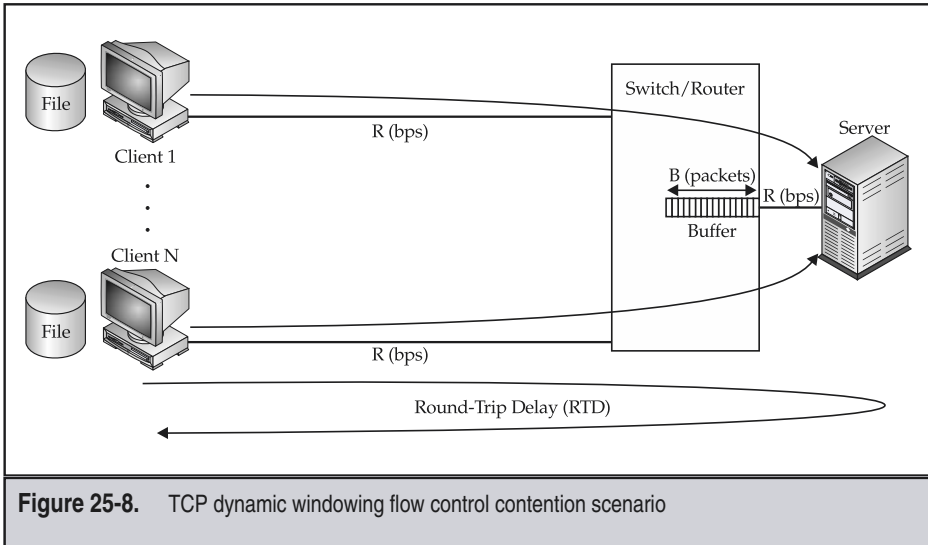


Figure 25-8. TCP dynamic windowing flow control contention scenario

buffer that has capacity B packets. All clients have the same round-trip delay (RTD) to the server. Since this configuration is completely symmetric, congestion causes all sources to begin retransmission simultaneously. This phenomenon is called “phasing” in the study of TCP performance.

Figure 25-9 illustrates the effect of two ($N = 2$) contending, identical sources on the TCP window size (measured in packets here) and the buffer fill versus the horizontal axis measured in units of round-trip delay (RTD). The figure shows the window size by squares on the figure every RTD time. Recall from Chapter 8 that the TCP slow start protocol increases the window size by multiplying by two each RTD until it reaches a value of one half the previous maximum window size. After passing the one half value, TCP increases the window size linearly by one packet for each RTD interval. The figure indicates the number of packets sent in each RTD interval by a horizontal line. Once the buffer overflows, TCP fails to receive an ACKnowledgment. In our example, we assume that TCP times-out after one RTD interval. When the time-out occurs, TCP decreases the window size to one. The buffer fill plot in the figure depicts the worst-case situation, where each TCP source generates packets according to this same function over time. Notice how the buffer fill returns to zero with the very short window sizes at the beginning of the start-up interval, but gradually increases as the window size increases in the linear increase region. In this example, the buffer capacity is $B = 32$ packets, which overflows on the 14th, 15th, and 16th RTD intervals, as shown in the figure. The maximum number of packets on-the-fly (F) is 20 in this example.

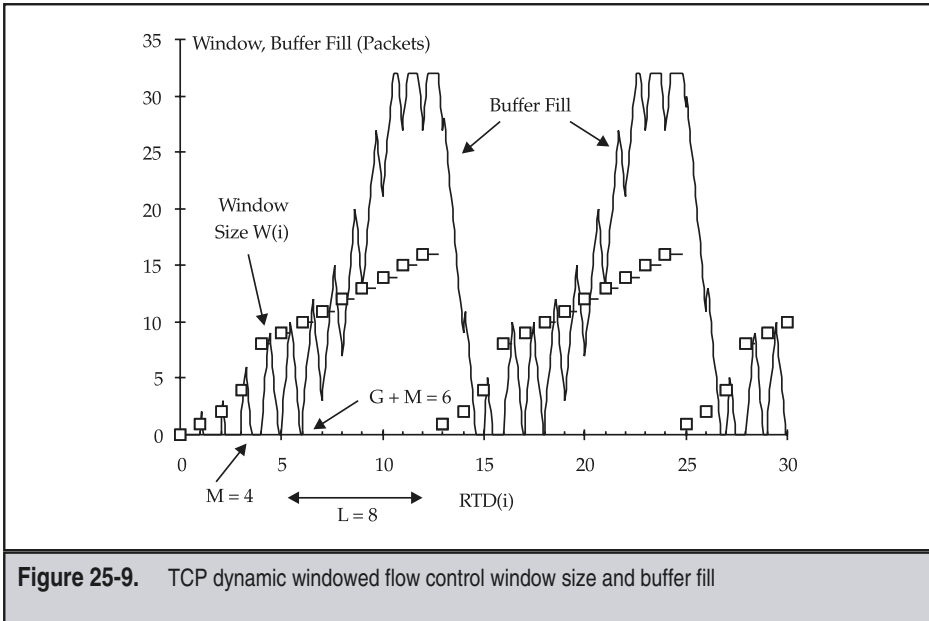


Figure 25-9. TCP dynamic windowed flow control window size and buffer fill

The net effect of this protocol is that each TCP session on average gets about half of the egress port bandwidth. This sawtooth type of windowing pattern is typical for TCP.

Voice and Data Integration

This section gives a time-tested model for voice transmission statistics within an individual call for use in a simple statistical multiplex gain model. Next, we use this model to estimate the savings resulting from integrating voice and data on the same transmission facility.

Voice Traffic Model

Most people don't speak continuously in normal conversation; natural pauses and gaps create an on-off pattern of speech activity, as illustrated in Chapter 24. On average, people speak only 35 to 40 percent of the time during a typical phone call. We call this the speech activity probability p . Furthermore, the patterns of speech activity are independent from one conversation to the next. Therefore, the binomial distribution introduced in Chapter 23 is a good model for the probability that k people are speaking out of a total set of N conversations using the shared facility as follows:

$$Pr[k \text{ out of } N \text{ speakers active}] = b(N, k, p) = \binom{N}{k} p^k (1-p)^{N-k}$$

$$\text{where } \binom{N}{k} \equiv \frac{N!}{(N-k)!k!}$$

The results of many studies performed since the introduction of digital telephony in the late 1950s show that most listeners don't object to loss of the received speech ranging between 0.5 percent and a few percent. Of course, the speech encoding and decoding mechanism determines the resulting quality in the event of loss—a very sensitive mechanism can magnify small amounts of loss, while a more resilient mechanism can hide the effects of greater values of loss. The parameter of interest is then the fraction of speech lost, commonly called the *freezeout fraction* FF [McDysan 92], defined by the following formula:

$$FF(N,C,p) = \frac{1}{Np} \sum_{k=C}^L (k-C) \binom{N}{k} p^k (1-p)^{N-k}$$

This expression has the interpretation equivalent to the fraction of speech lost by an individual listener. What we need in the following analysis is a function that determines the capacity required C for a given number of speakers N , each with a source activity probability p . The subsequent analysis denotes this function as $C(N, p, FF)$.

Statistically Multiplexing Voice Conversations

Satellite communication and undersea cable communication of voice have long used statistical multiplexing of many voice conversations to reduce costs. Often, experts refer to the devices performing this function as Digital Circuit Multiplication Equipment (DCME), since statistical multiplexing effectively packs multiple conversations into a single equivalent voice channel. ATM Adaptation Layer 2 (AAL2) or VOMPLS could support the next generation of DCME, as well as integrated voice and data access. However, this gain occurs only when a system multiplexes enough conversations together. Unfortunately, the statistical multiplexing of voice reaches a point of diminishing returns after reaching a critical mass. Let's look at an example to see why this is true. First, the statistical multiplex gain $G(N,p,FF)$ for N conversations with voice activity probability p and a freezeout fraction objective FF is

$$\text{Voice Stat Mux Gain: } G(N,p,FF) = \frac{N}{C(N,p,FF)}$$

where $C(N,p,FF)$ is the required number of channels, as defined in the preceding text. The following example assumes that the voice activity probability $p = 0.35$ and $FF = 0.5$ percent for all sources. Figure 25-10 plots the results of the required capacity function $C(N,p,FF)$ and the resulting statistical multiplex gain versus the number of sources N . The curve is not regular because the required capacity function returns integer values.

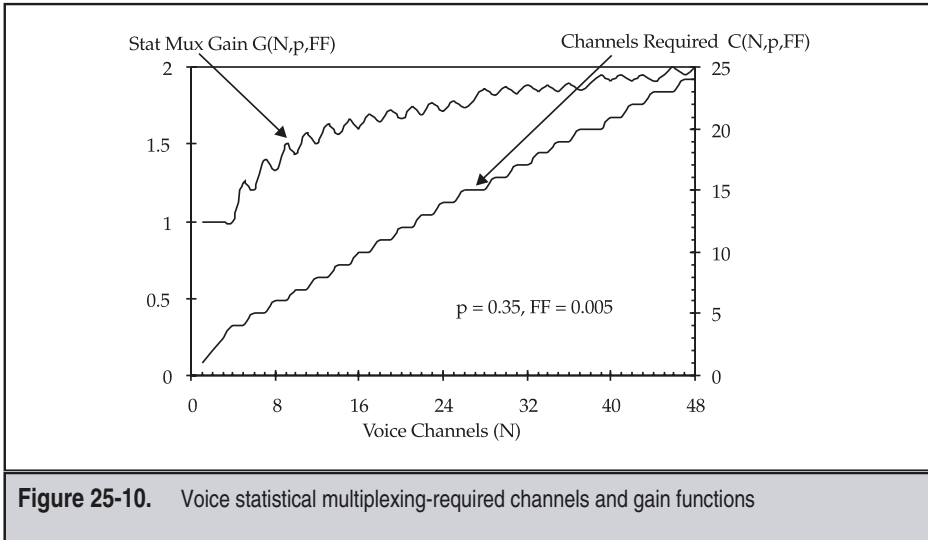


Figure 25-10. Voice statistical multiplexing-required channels and gain functions

This analysis illustrates several key points. First, until the number of sources reaches 5, there is no gain. After this point, the gain increases slowly and reaches a maximum value of 2. Indeed, although not shown on the chart, the gain for 128 sources increases to only a factor of 2.2.

Voice/Data Integration Savings

The curious reader may now be thinking what gains remain for integrated voice and data transmission. Figure 25-11 illustrates the block diagram of an integrated voice/data multiplexer (IVDM). It has N voice channel inputs with activity probability p , and freezeout fraction QoS parameter FF served by a link of capacity $C(N,p,FF)$. The quantity determined in this section is the additional data traffic $D(N,p,FF)$ that the IVDM can carry by utilizing the capacity unused by periods of voice inactivity.

The answer lies in the observation that although the voice multiplexing system reserves capacity $C(N,p,FF)$ for transmission of speech, the speech utilizes an average capacity equal to Np —the mean of the binomial distribution. Therefore, the percentage of additional data traffic carried by an integrated voice/data system when compared with separate voice and data systems is [McDysan 89]

$$\text{Data Integration Savings: } D(N,p,FF) \leq \frac{C(N,p,FF) - Np}{C(N,p,FF)}$$

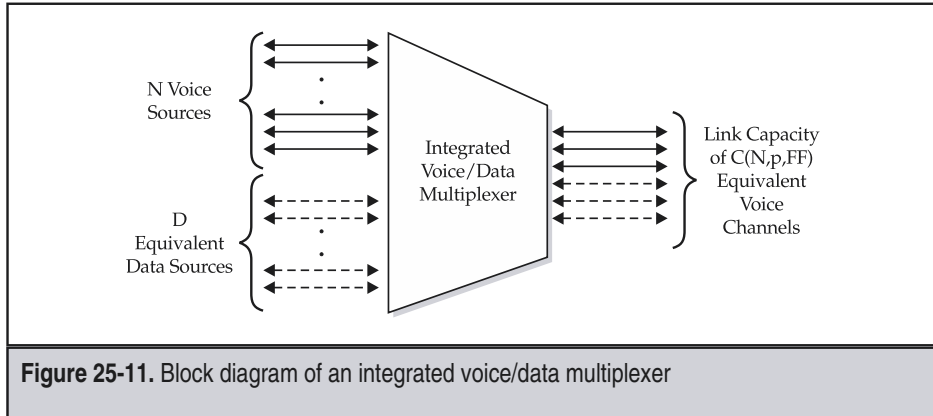


Figure 25-12 plots the fraction of additional data traffic supportable $D(N,p,FF)$ versus the number of voice sources N for the same voice activity probability $p = 0.35$ and $FF = 0.5$ percent used in the previous example. For relatively small systems, integrating voice and data makes great sense, and ATM or VoIP are standard ways to implement it. Here again, however, a point of diminishing returns occurs as the number of voice sources increases above 30, where the additional capacity available for data is approximately 25 percent. For a system with 128 voice sources, the savings decreases to 11 percent. Hence, don't expect an economy of scale when multiplexing large amounts of data and voice traffic.

This means that the maximum benefits accrue when multiplexing voice and data together on access lines operating at rates ranging from $n \times DS0$ up to DS1 or E1. This offers the greatest benefits to medium-sized business locations because of reduced access line charges. Another benefit for carriers is better utilization of expensive transoceanic cables, satellite communications, and radio frequency networks via integrated voice and data multiplexing.

OVERVIEW OF THE NETWORK PLANNING AND DESIGN PROCESS

This section summarizes the network design and planning process. When done properly, this process is a closed recurring cycle of measurement, forecasting, design, analysis, implementation, and then beginning with measurement again, as depicted in Figure 25-13. These steps in the process feed each other, but they also require external inputs, as shown in the figure. The following sections describe this process further and place these activities in the overall context of network planning and design.

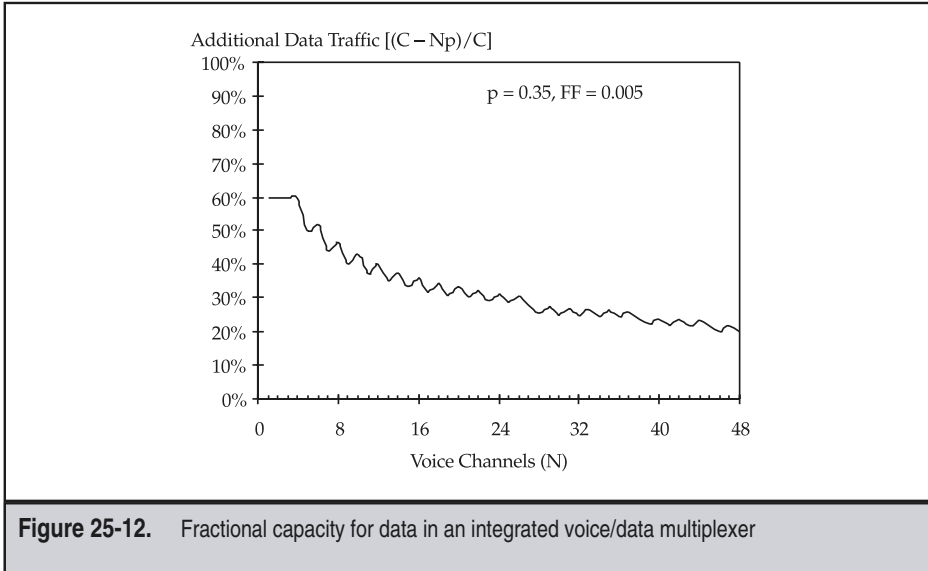


Figure 25-12. Fractional capacity for data in an integrated voice/data multiplexer

Network Design Approaches and Modeling Philosophy

There are several approaches to network design. As always, one approach is to do nothing proactive at all. That is, simply wait and see what happens and react. While this may be acceptable for smaller networks without stringent quality requirements, it is not ap-

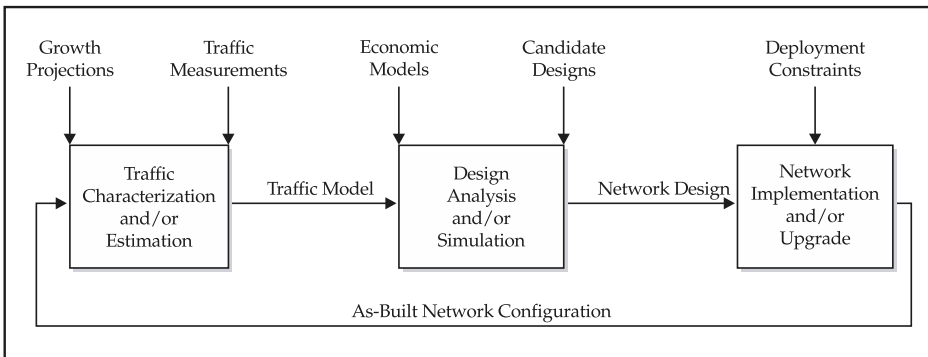


Figure 25-13. Flow diagram for the network design and planning process

propriate for QoS-aware applications or networks that provide service to multiple customers. Network designers need some way of predicting network performance; otherwise, users complain and productivity declines.

An essential aspect of traffic engineering philosophy relates to the required accuracy of the model. As expected, the more complicated the model, the more difficult the results are to calculate. A good guideline is to make the accuracy of the switch and network model comparable to the accuracy of the traffic model.

If you know only approximate, high-level information about the offered traffic, then an approximate, simple switch and network model is appropriate. The old computer science adage—garbage in, garbage out—applies here. If you know a great deal of accurate information about the traffic, then investment in an accurate switch and network model, such as a detailed simulation, is appropriate. Using a detailed traffic model in conjunction with an accurate switch and network model yields the most realistic results. Beware that detailed models are quite complex and require simulation via a high-performance computer system.

When neither traffic nor network details are available, approximations are the only avenue that remains. Approximate modeling is usually simpler, often requiring only the analytical methods described in the body of this part. One advantage of the analytical method is that insight into relative behavior and trade-offs is much clearer. One word of caution remains, however: these simplified models may yield overly optimistic or pessimistic results, depending upon the relationship of the simplifying assumptions to a particular real-world network. Therefore, modeling should be an ongoing process. As you obtain more information about traffic characteristics, switch/router performance, and quality expectations, feed this back into the modeling effort. For this reason, modeling has a close relationship to the performance measurement aspects of network management, as discussed in the next part of this book.

Measuring Traffic and Performance Data

In order to apply the techniques described in this book, the designer must select a traffic model. The best way to determine the appropriate statistical model is via measurement. External measurement devices can collect detailed performance data [Petrovsky 98, Apisdorf 97, Claffy 98]. These tools provide the most detailed information possible, since they examine the header of every packet or cell traversing an interface on a specific switch or router. Thus, we have the entire time series of the actual traffic and can estimate a number of statistical parameters, including the mean, variance, and correlation. References [McDysan 00, Stallings 97] describe some basic tests useful to determine whether your traffic data meets a particular statistical model. In particular, it may be important to determine the degree of self-similarity of specific traffic patterns. Unfortunately, these tools can result in reams of data that make higher-level patterns quite difficult to discern.

Another technique involves collection of statistical data through standard management information bases (MIBs), such as the IETF RMON MIB. This provides higher-level summary information about a number of switch or router interfaces. Unfortunately, collecting statistics like the number of packets or cells handled by an interface over a number of minutes reduces our knowledge to the average value of the offered traffic. Other parameters, such as the number of dropped packets or cells due to buffer overflow, indirectly imply other parameters about the traffic if we know the buffer capacity and queue service policy of the switch or router.

Collecting information about the traffic matrix is a difficult problem. Traditionally, analysis of call records from telephone networks yielded one means of developing a traffic matrix. A similar method is applicable for ATM SVC networks. For connection-oriented ATM PVC networks, the MIB or external traffic monitoring device method performs best, since the source and destination end points are fixed. Similarly, statistics from MPLS label switched paths (LSPs) provide a source of traffic matrix data. For connectionless IP networks, the data may be collected from routers or using an external traffic monitor.

Analyzing and Simulating Candidate Networks and Technology

Typically, network designers base decisions upon long-term, historical trending and projections of traffic characteristics. This method applies to private as well as public networks. The types of decisions involved range from deciding on a replacement network technology to selecting sites that require upgraded switches or routers, to changing the homing or installing additional transmission capacity.

A number of commercially available network analysis tools exist that implement many of the analytical models described in this book, as well as a number of heuristic design techniques. Typically, these tools require that the user enter some information about traffic patterns, volumes, and characteristics. Most of these tools also include the means to input economic data about the cost of switches/routers, transmission capacity, and other charges. These network planning tools then provide answers to “what if” questions regarding candidate network designs, upgrades, or changes using analytical methods. In the end, the economic considerations involved in network design are a pivotal decision point for most enterprises.

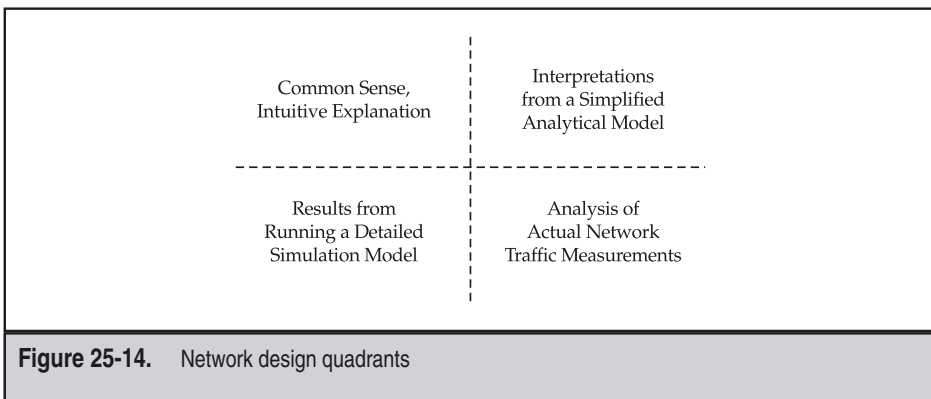
Another class of tool is the event-based simulation model. This tool enables a network designer to model the operation of a network, or even the discrete components of a switch or router, at any level of detail desired. However, beware that modeling at low levels of detail requires extensive computing resources to simulate even small periods of real time. Therefore, a practical compromise is to simulate components of a network element and then analyze networks of such elements using approximations that take less computer time to perform. A number of commercial tools exist that operate using the event-driven simulation paradigm.

Practice Makes Perfect

Frequently, the designer must make a trade-off between cost and performance. For example, a more expensive network may accommodate all traffic during the busy period with high quality, even in case of failures. On the other hand, a less-expensive network may deliver lower quality during busy periods and intervals when failures disrupt part of the network. However, the job doesn't end after the network design is done.

Developing a plan to continuously monitor network performance during the network design phase is not enough. You must collect the measurement data and analyze it to truly optimize a network. When acceptable QoS and capacity requirements are critical to your enterprise, proactively monitoring actual performance and comparing it against the design objectives is a critical step in completing the cycle. Careful analysis of actual performance helps immeasurably when tuning network parameters, planning for network upgrades, or identifying problem areas. Furthermore, keeping on top of your network's performance frequently means happier users and greater productivity for your enterprise. Keep in mind the concept of continuous improvement. As your experience grows, so will the effectiveness of your decision making. In a growing network, each incremental reduction in cost or improvement in productivity returns an increasingly larger return as traffic volume grows.

Ideally, you should strive to explain, justify, and validate your design decisions using the four quadrants of network design illustrated in Figure 25-14. If all of these views are consistent, then you should have high confidence that your plan will work. This book focuses on the quadrants of intuitive explanation and analysis, which are typically the first steps in the design process. The next section provides an overview of network design and modeling tools.



NETWORK DESIGN AND MODELING TOOLS

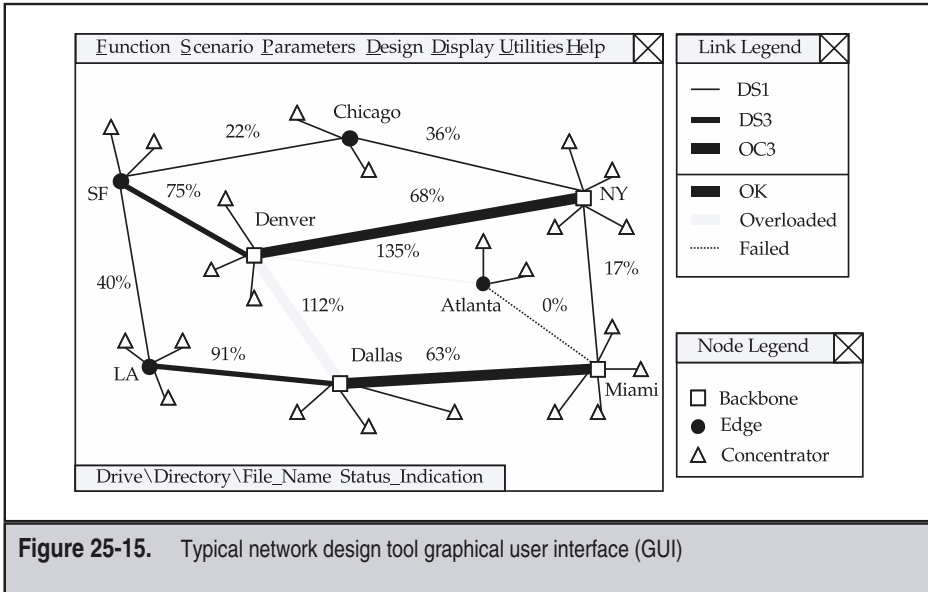
Designing a network is a complex task. The analysis in the examples in this part of the book is useful in analysis of individual nodes, a set of links, and very simple network topologies. However, these techniques do not directly model the performance of real-world networks. Furthermore, a real-world model must support economic analysis of various alternatives. The next step is to purchase an off-the-shelf tool, start with a public domain program and modify it, or write your own. Most of these tools have the following attributes:

- ▼ A graphical user interface (GUI) for managing the topology database of network topology and economic information, and displaying results
- A set of functions for manipulating and generating offered traffic scenarios
- The capability to model a particular admission, policing, and routing algorithm for the network configured in the topology database
- The means to compute and display the results predicted by the analytical and/or simulation model for a set of scenarios specified by the user
- ▲ A set of utilities for managing database versions of topology, economic data, “what if” scenarios, and results

The following sections summarize the important attributes for each of the preceding items. For an inexpensive introduction to network design tools, [Cahn 98] describes a number of examples using the downloadable “Delite” network design tool that illustrates many of these concepts. The author provides periodic updates and bug fixes to the program on the Web site identified in the book.

Design Tool Graphical User Interface (GUI)

The GUI provides for data entry and display of results. Figure 25-15 shows a representative view of such a tool. The format typically involves symbols that indicate nodes, with specific shapes and/or colors indicating the type of equipment at that location and its status. Frequently, the tool allows the user to move nodal symbols around to make the overall topology less cluttered than an accurate geographical map would depict. The other principal component of the GUI is links interconnecting the nodes. The color, line style, and/or thickness of the lines often have particular meaning. For example, green lines may indicate links with sufficient capacity, while red lines may indicate links with insufficient capacity. Most tools also display text or numbers next to nodes and links indicating parameters associated with these network elements. Additionally, most tools have a menu of some sort—for example, a bar at the top of the screen, as shown in Figure 25-15. Finally, the tools also must have some means for data entry, such as a pop-up menu, as shown in the figure, or, preferably, input from a database. Most of these tools assume that the user is familiar with the basic concepts of probability, queuing theory, and traffic engineering summarized in this part.



Specifying Design Scenarios

Models implement the capability for users to specify a number of generic classes of scenarios, including

- ▼ Definition of traffic growth projections to determine scalability of an existing network design
- Specification of link and nodal failure scenarios for testing network reliability
- ▲ Alternative selections for nodal and link locations

Usually, network design is not a static problem. Typically, traffic grows, or in some cases, declines. Traffic rarely stays at a constant level. Therefore, a conscientious network designer must diligently determine whether performance will suffer from growth and recommend incremental capacity at the most economical level. Increasingly, network availability in the event of failures is an important design criterion. Finally, selection of new or changed nodal sites and choice of links can have significant economic and performance impacts.

Modeling Network-Specific Capabilities

An important aspect of a design tool is the accuracy with which it models the specific functions and performance of the actual switches/routers, links, and any other functions in the network design. These include

- ▼ Parameters that determine the operation of admission control algorithms implemented by the nodes
- Capacity, efficiency, availability, and error rates of links that connect nodes
- Information that controls the operation of the routing and signaling algorithms implemented by network nodes
- ▲ The maximum capacity and detailed configuration rules of vendor-specific switches/routers and other network elements

Some commercial network design tools often accurately model the admission control and routing algorithms in conjunction with configuration rules for popular vendor equipment. Therefore, be certain to investigate whether a specific tool models the vendor(s) in your network. Some parameters such as capacity, protocol efficiency, and error rate are generic, and hence vendor independent. In some cases, if vendors implement a standard algorithm such as OSPF, IS-IS, or PNNI, then a generic tool is sufficient. In other cases, vendors offer design tools or consulting services tailored to model their equipment.

Displaying and Comparing Results

There are three principal types of results that network design tools generate:

- ▼ The least-cost network topology that meets the constraints specified for a particular design scenario
- The predicted performance and cost for the specified network topology subjected to the offered traffic load under the conditions of the user-specified scenario
- ▲ Cost and performance comparisons of a number of heuristically derived network topologies

Unfortunately, algorithms exist that compute optimal topologies for only the simplest networks under very simple assumptions. Most real-world network design problems fall under the second or third categories. Many tools simply predict performance for an existing network or planned upgrades. Other tools enable the network designer to trade off cost against performance. Typically, lower-cost networks have greater delays and lower availability than a well-designed higher-cost network. In general, achieving the lowest cost simultaneously with high performance is not possible. The approach that does work is to set realistic performance objectives and then design the network that just meets these requirements at the lowest possible cost.

REVIEW

This chapter covered several key considerations in the design of networks and switches. It began by reviewing the impact of delay, loss, and delay variation on interactive and non-real-time applications, giving guidelines for the selection of link speed, burst duration, and loss ratio. The text covered the effect of loss on effective throughput for Go-Back-N and selective retransmission. The discussion also showed how delay variation accumulates nonlinearly across a network of switches or routers. Next, the text covered the factors that affect performance of the Internet's popular Transmission Control Protocol (TCP). Then, the chapter covered the subject of voice statistical multiplexing gain, along with an assessment of the savings achievable in an integrated voice/data network. Finally, the chapter introduced the overall network design and planning process. This description centered on how to use functions commonly found in network design tools over the life cycle of a network.

PART VII



Operations and Network Management for ATM and MPLS

Now that we've defined the details of ATM and MPLS networking, this book now takes on the challenge of how to manage these networks. Toward this end, this part provides the reader with an overview of operations, network management architectures, network management protocols, and object-oriented databases, as well as the specific protocols used for management and performance measurement. Since at the time of writing, the standards for management of ATM were the most mature, the focus is on ATM. First, Chapter 27 defines the

philosophy of Operations, Administration, Maintenance, and Provisioning (OAM&P) to set the stage. The coverage continues with a presentation of network management architectures defined by the standards bodies and industry forums. Chapter 28 then covers the network management protocols developed by standards bodies and industry forums to solve the network management problem. This includes the IETF's Simple Network Management Protocol (SNMP), and the ITU-defined Common Management Information Protocol (CMIP). We then give a summary of key Management Information Bases (MIBs) defined in support of ATM and MPLS networks. Finally, Chapter 29 addresses the topics of ATM layer management and performance measurement. The text defines ATM-layer Operations and Maintenance (OAM) cell flows and formats. The text first covers fault management, which is the basic determination of whether the ATM service is operating correctly. The chapter summarizes how these performance measurement procedures work to confirm that the network indeed delivers the specified ATM-layer Quality of Service (QoS) objectives. This chapter also summarizes the emerging alternatives for management and performance measurement of MPLS.

CHAPTER 26



Operational Philosophy and Network Management Architectures

This chapter covers the important topic of operational philosophy and network management architectures. Starting things off, the text first discusses basic Operations, Administration, Maintenance, and Provisioning (OAM&P) philosophy. We identify generic functions that apply to almost any type of communication network. The text summarizes network management architectures defined by a number of standards bodies and industry forums that have specific application to ATM or MPLS. The next chapter describes the network management protocols and databases employed by these architectures.

OAM&P PHILOSOPHY

Network management is about achieving quality. If your network requires quality, then the expense and complexity of comprehensive network management technology is well justified. In order to set the overall stage and context for this part of the book, this section gives a brief definition of each element of Operations, Administration, Maintenance, and Provisioning (OAM&P) and describes how they interrelate as depicted in the flow diagram of Figure 26-1. Each of the major functional blocks performs the following functions:

- ▼ Operation involves the day-to-day, and often minute-to-minute, care and feeding of the data network in order to ensure that it is fulfilling its designed purpose.
- Administration involves the set of activities involved with designing the network, processing orders, assigning addresses, tracking usage, and accounting.
- Maintenance involves the inevitable circumstances that arise when everything does not work as planned or it is necessary to diagnose what went wrong and repair it.
- ▲ Provisioning involves installing equipment, setting parameters, verifying that the service is operational, and also deinstallation.

Despite the order of the words in the acronym, OAM&P, these functions are better understood from the top-down work flow described previously. This sequence basically models the life cycle of an element in a network. First, a network planner administratively creates the need to augment or change the network. After the hardware and/or software is purchased, tested, and deployed, then operators must provision it. Once it is in service, operations continuously monitors and controls the network element, dispatching maintenance for diagnosis or repair as required.

Administration

Administration of a network involves people performing planning for the network. This work includes determining the introduction of new elements to grow the network, add features/functions, and remove obsolete hardware and software elements. In this context, a managed element is either an individual instance of hardware (such as a switch/router or an interface card), connectivity via underlying transport networks, or logical design elements (such as address plans or key network operation performance objectives). Administration

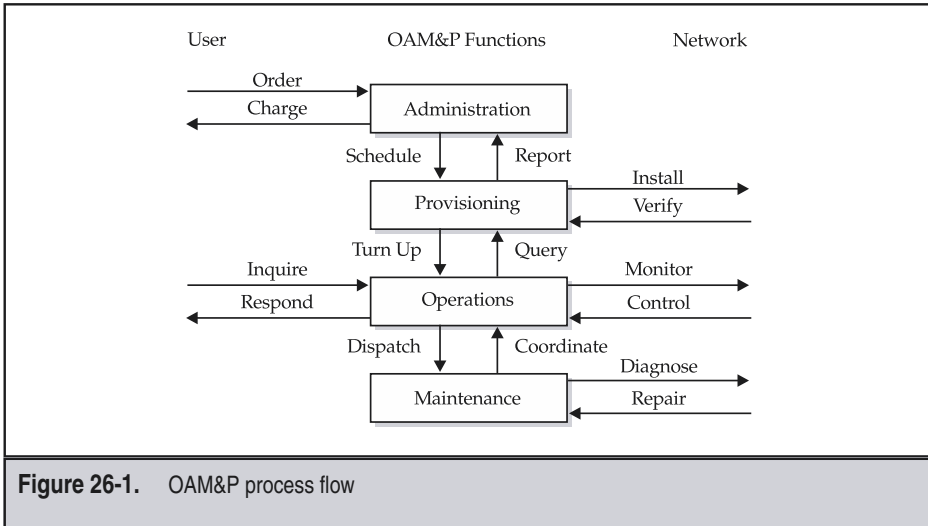


Figure 26-1. OAM&P process flow

includes some key management functions and some business functions. First, a network must be designed, either as a private, public, or hybrid network. Once design is complete, the elements must then be ordered, along with scheduling installation and associated support. The enterprise must develop an administrative plan for staging the provisioning, operations, and maintenance activities. This often involves automated system support for order entry, order processing, work order management, and trouble ticketing. Orders must be accepted from users, and the provisioning process initiated. Address assignments are made where needed.

Once the service is installed, usage data must be collected for traffic analysis and accounting. If departmental charge-backs or customer charges are important, an accounting and billing system must have access to the usage data. Forecasts, business requirements, and traffic analysis may show that changes to the network are required. Network planning frequently makes use of automated tools in the process of administration.

Provisioning

Provisioning of a network involves the day-to-day activities that actually introduce physical and logical changes in the network to implement the administratively planned growth or change in the network. Ideally, provisioning activity follows set rules and procedures developed in the design phase of a network, or else learned and refined from past provisioning experience. Provisioning controls commissioning and decommissioning of physical elements, as well as the configuration of logical elements. Physical actions associated with provisioning include installation of new or upgraded equipment, which may also include updating vendor switch software. Hardware changes require on-site support, while with proper infrastructure support, software upgrades may be

done remotely. Access line or trunk installation and turn-up is also part of provisioning. A key part of processing orders is the establishment of service-specific parameters. While this may be done manually for small volumes of orders, automation becomes cost effective for processing a large volume of provisioning work. Once the hardware and software parameters are in place, the final step of the provisioning process ensures that the service performs according to the objectives prior to release to the end user. Verifying performance often involves tests in conjunction with operations, such as circuit bit error rate testing, loopback testing, or throughput measurements.

Operations

Operating a network involves monitoring the moment-to-moment fluctuations in the performance of the network and deciding which events require intervention to bring the network into compliance with an ideal performance goal set by designers. Operations provides an organization with a collective memory of the past performance of the network. Included in that memory are the activities that identified past service breakdown and the activities that corrected the problem. Some of these corrective actions come from knowledge of the services supported and the network design; yet in the real world, some come from practical experience.

Monitoring the network involves watching for faults and invoking corrective commands and/or maintenance actions to repair them. It also involves comparing measured performance against objectives and taking corrective action and/or invoking maintenance. Invoking corrective actions involves operators issuing controls to correct a fault or performance problem or resolving a customer complaint. A key operational function involves assisting users to resolve troubles and effectively utilizing network capabilities. The operations function coordinates actions between administration, maintenance, and provisioning throughout all phases of the service.

Maintenance

Maintaining a network involves many instances of unplanned changes. Maintenance actions involve changes not instigated via the administrative design or service provisioning process. Examples of maintenance actions are changing interface cards, replacing failed common equipment, and troubleshooting physical circuit problems. Once operations identifies a problem, it works with maintenance engineers to isolate and diagnose the cause(s) of the problem. Maintenance engineers apply fixes to identified problems in a manner coordinated by operations. Control by operations is critical, since in complex networks like ATM and MPLS, maintenance actions applied incorrectly may result in additional problems. Operations coordination with Maintenance can involve dispatching service personal to the site of the fault, arranging parts delivery, or coordinating repair activities with service suppliers. Besides responding to problem events, an important maintenance activity is performing periodic, routine maintenance so that faults and performance degradations are less likely to occur. Routine maintenance may involve automated test cycles supplemented with application of preplanned inspections and cyclical service of physical elements.

Unique Challenges Created by ATM

Usually, standardization of network management occurs later in the technology life cycle, well after the service is introduced. It is commonly believed that only after you have built the network, determined what can go wrong, and discovered what is needed to make it work can you finalize how to operate, administer, maintain, and provision it. However, good planning can provide these OAM&P functions in a much more proactive manner soon after the introduction of technology. While there is no substitute for experience, network management is essentially the application of a control system model to a network; that is, feedback from observations of the network is used to apply actions to network elements to drive the behavior toward a desired state. Therefore, such planning should be a part of the original network design, just as measurements and controls are essential in any engineering endeavor. Experience becomes a feedback mechanism to refine the initial network management and operations designs. For example, the ATM OAM standards came after the first wave of detailed ATM networking standards, but now a solid international standard exists for ATM layer management and performance measurement.

Many of the management functions required for an ATM network are similar to those for a circuit-based network. This complicates ATM network management, since there is a need to support legacy systems, as well as to support new services. But since ATM changes the communications paradigm from that of TDM, many tried and true TDM troubleshooting procedures at least require revision. The direct inclusion of multiple services supported over ATM also extends the network management domain. Add to this the complexities of behavior introduced between ATM and the underlying physical layer by (sometimes) competing automated restoration and recovery schemes.

Unique Challenges Created by MPLS

As described in Chapter 10, the original motivation for MPLS arose as a better, more efficient, custom fit traffic engineering method as a replacement for IP over ATM in the core of service provider networks. As such, MPLS benefited from the experience gained from IP over ATM, resulting in a tighter integration of the routing and signaling control protocols with the forwarding component as described in Chapters 11 and 14. Since MPLS is a much newer protocol, the standards for management of it are much less mature. Furthermore, since the principal applications deployed to date are IP over MPLS, some IP management functions have been adopted in support of MPLS.

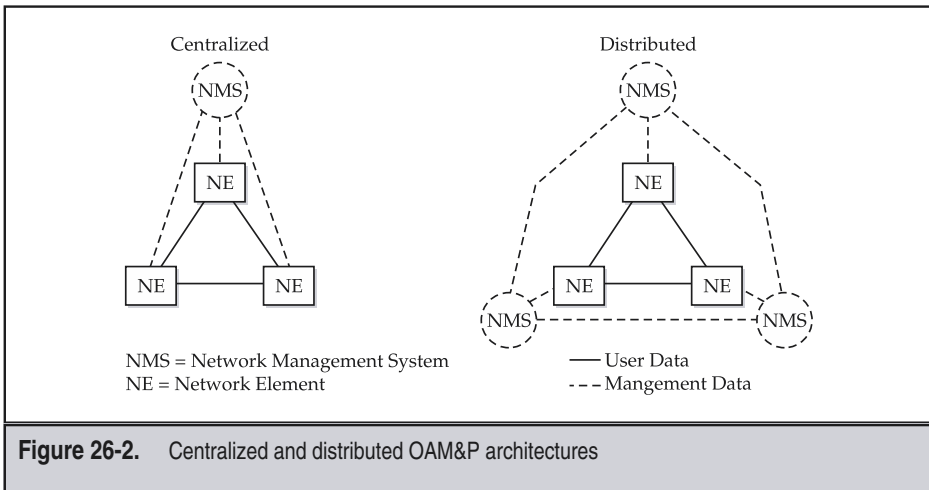
However, with the development of support for multiple services over MPLS as described in Part 4, many of the same management challenges that faced ATM now confront MPLS. At the time of writing, there were two competing approaches for how best to meet this need. The ITU proposed adapting the approaches developed for ATM to MPLS, while the IETF was pursuing reusing IP management protocols for MPLS. We compare and contrast these approaches in the remainder of this part when discussing management for MPLS. Since there was no adopted standard at publication time, we do not provide details of any specific protocol procedure but instead provide references for the reader interested in discovering the current state of these important evolving standards.

NETWORK MANAGEMENT ARCHITECTURES

Standards are the place where practical agreements are documented on how best to manage real networks on the basis of operational experience. Standards are also the place where theoretical network management approaches, some of which are never implemented, are also documented. Our focus is on the practical work from various standards bodies and industry forums for network management architectures, functional models, and protocols that have seen actual deployment. The following covers work done by OSI, ITU-T, ATM Forum, and IETF.

Centralized Versus Distributed Network Management

It is important to consider how the network management systems impact operational philosophy. An important decision is whether to adopt a centralized or distributed Network Management System (NMS) architecture for managing a network of ATM- or MPLS-based Network Elements (NEs); Figure 26-2 depicts these two design extremes. Some will opt for a centralized approach with the expertise concentrated at one location (possibly with a backup site for disaster recovery), with remote site support for only the basic physical actions, such as installing the equipment, making physical connections to interfaces, and replacing cards. In this approach, the software updates, configurations, and troubleshooting can be done by experts at the central site. The centralized approach requires that the requisite network management functions defined in the previous section are well developed, highly available, and effective. Furthermore, a centralized design requires connections from every device back to the centralized management system. Non-real-time functions are often best done in a centralized manner.



An alternative approach is to have several independent management sites. For instance, when riding the “bleeding-edge” of technology, the designers may want to have expertise at every site. Also, this approach may be required if the network management system is not sophisticated or if the equipment entails a number of actions that can only be done at the site. In some cases, the volume of information collected from the network elements drives users toward a distributed processing design. Finally, some lower-level, well-defined, automated functions are best performed in a distributed manner. Real-time functions are often best done in a distributed manner. Realistically, a network can be managed by a combination of centralized and distributed management connectivity; in this case, the management OSS might support local element managers reporting up to a central manager.

There is a performance trade-off between the centralized and distributed architectures. Transferring large volumes of management data to a central site often requires a large centralized processor. It is all too easy to get in the situation of having too much data and not enough information. On the other hand, defining the interfaces and protocols for a distributed system is complex. Choosing to adopt either a centralized or distributed Network Management System (NMS) architecture is only one example of the many different and sometimes conflicting design choices that are faced when implementing management of a complex network. Choosing network management systems frequently forces the choice of an entire operational philosophy. So, how can you make a cost-effective, technically sound choice? The starting point is usually selection of a particular architectural approach.

OSI Network Management Functional Model

The ISO and the ITU-T adopted the following five generic functional areas for network management [ITU M.3400], commonly abbreviated as FCAPS:

- ▼ **Fault management** Enables the detection, isolation, and correction of abnormal operation of a network. It also provides quality assurance measurements for reliability, availability, and survivability.
- **Configuration management** Provides functions that identify, exercise control over, collect data from, and provide data to network elements.
- **Accounting management** Enables measurement of network service usage, determination of service provider costs, and determination of the price for service.
- **Performance management** Involves evaluation and reporting on the behavior of networks or network elements by gathering and analyzing statistical data for the purpose of monitoring and correcting network behavior; it also serves as an aid to planning, provisioning, maintenance, and the measurement of quality.
- ▲ **Security management** Provides for the management of all of the preceding management functional areas, as well as for all management-related transactions.

Many standards and specifications make use of this tried-and-true functional taxonomy of network management, and we will refer to FCAPS where appropriate. Some aspects of ATM and MPLS configuration management, connection provisioning, fault management, and performance management are covered by ATM Forum- and IETF-defined MIBs and protocols, as described in Chapter 27. At present, the ITU-standardized ATM OAM-cell-based management described in Chapter 28 primarily covers fault management and performance management. At publication time, MPLS standardization was developing in the areas of fault and performance management, in the IETF using IP-based protocols, while the ITU-T approach was based upon refinements and extensions to the ATM-OAM approach. We summarize the direction of these approaches in Chapters 27 and 28, respectively.

ITU Telecommunications Management Network (TMN)

Figure 26-3 depicts the layered model for the Telecommunications Management Network (TMN) operations functions described in ITU-T Recommendation M.3010. A subsequent section defines all interfaces between the layers, labeled Q3, and interfaces between the layers and their peers, labeled X. This model abstracts lower-level details further up the hierarchy, enabling effective service and resource management. Starting at the bottom of the figure, physical network elements are devices, such as ATM switches,

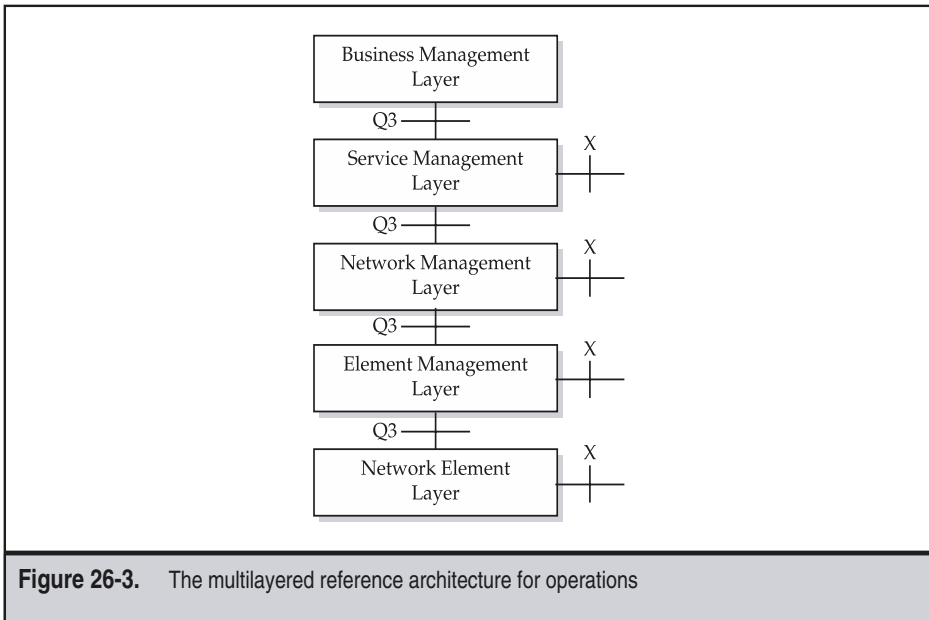


Figure 26-3. The multilayered reference architecture for operations

LAN bridges, routers, or workstations. The element management layer manages network elements, either individually or in groups, to develop an abstraction of the network element functions to higher-layer operations functions. It is concerned with maintenance but also involves performance management, configuration, and possibly accounting. Many vendors provide proprietary element managers that control an entire network of their devices. The network management layer addresses functions required across an entire geographic or administrative domain, which includes configuration and maintenance. This layer also addresses network performance by controlling network capabilities and capacity to deliver the required Quality of Service. The service management layer is responsible for the contractual aspects of services provided to customers by carriers, for example, subscriber administration and accounting management. It also includes statistical data reporting, managing the status of interfaces with other carriers, and interactions with other services. The scope of the business management layer is the entire enterprise, encompassing proprietary functions. Since this layer performs proprietary functions, it usually does not provide the standard X interface to a peer NMS layer. Please note that the layers in this model represent logical functions, not physical systems.

The Q3 interface between the network element, element management, and/or network management layers is defined for the TMN architecture ITU-T Recommendation M.3010 using the OSI management defined in ITU-T Recommendation X.701. Practically, this becomes the use of GDMO (Guidelines for the Definition of Managed Objects) as described in ITU-T Recommendation X.722, which gives a notation for objects derived from the M.3100 information model, providing CMIS (Common Management Information Service, [ITU-T X.710] management services, communicated via the Common Management Information Protocol (CMIP) [ITU-T X.711], carried over ROSE [ITU-T X.219, ITU-T X.229]. For ATM, the Q3 interface often refers to the implementation of the ATM Forum M4 interface described later. Recommendation M.3010 indicates that other models are valid, so that systems without all of these layers, or systems with different layers and protocols, are also acceptable. For example, the Common Object Request Broker Architecture (CORBA) protocol suite is often considered comparable to CMIP.

Figure 26-4 illustrates several possible physical implementations of the preceding logical reference architecture, showing how the lowest three logical layers may map to physical systems. Figure 26-4a shows separate systems implementing each layer, where element management is performed on a one-for-one basis for each network element. This design could use computers at the element level to convert from a proprietary network element interface to a standard interface; these computers are in turn managed by a standard NMS. Figure 26-4b illustrates a system that integrates the network and element level management into a single overall management system. Proprietary vendor management systems often implement this architecture. Figure 26-4c illustrates a system in which network management intelligence and standard interfaces are distributed to each network element. Switch vendors who implement all MIB standards and provide open access to their proprietary MIBs follow this model. Finally, Figure 26-4d illustrates a hierarchical system in which element management systems manage groups of Network Elements (NEs) and then feed these up into an NMS that manages an entire network. Sometimes the processing requirements of larger networks dictate this hierarchical structure.

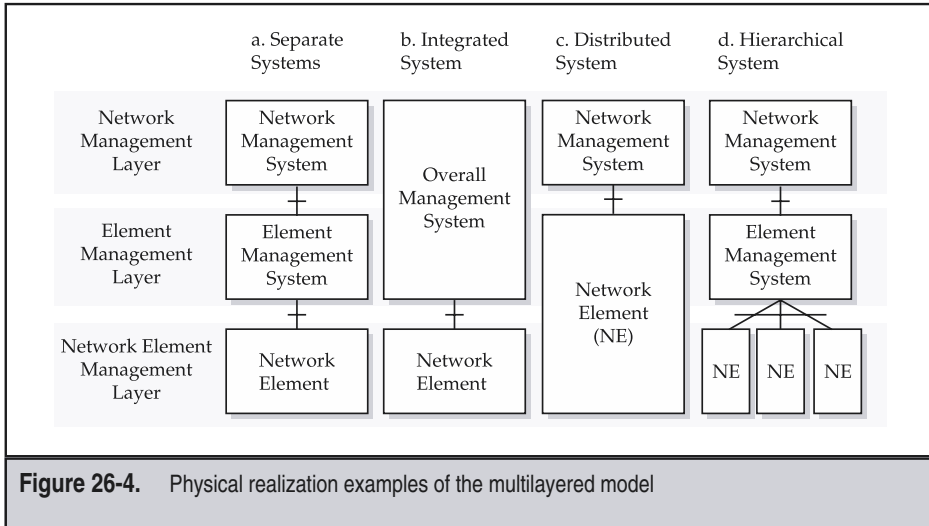


Figure 26-5 depicts the ITU-T vision for the standardized Telecommunications Management Network (TMN) architecture [ITU M.3010]. Starting at the top of the figure, a carrier's Operations System (OS) connects to a packet- or circuit-switched data communications network (DCN) using one of the TMN standard interfaces, denoted by the letter X, F, or Q. The X interface supports connections to TMNs in other carriers. For example, the X interface supports coordination of restoration requests between carriers; the ITU-T X.790 trouble ticket exchange specification is an example of such coordination. The F interface allows operators to retrieve and modify management information; for example, via a workstation, as shown in the figure. The Q3 interface comprises layers 4 through 7 of the OSI protocol reference model. The ITU-T utilizes the OSI standardized Common Management Information Service Elements (CMISE) and the associated Common Management Information Protocol (CMIP) for the Q3 interface. The Qx interface supports protocols other than the standard Q3 interface; for example, SNMP. Mediation devices (MDs), today more often called gateways, convert from these Qx interfaces to the standard Q3 interface. Automatic conversion routines are available between popular protocols, such as for conversion between SNMP and CMIP.

The software architecture of TMN includes functionally grouped capabilities called *operations systems functions* that perform the layered functions described earlier with reference to Figure 26-3. It is important to remember that network element functions at the individual device level constitute the source and sink of all network management observations and actions. They include traffic control, congestion control, ATM layer management, statistics collection, and other ATM-related functions. The mapping of these software functions onto the hardware architecture is an implementation decision.

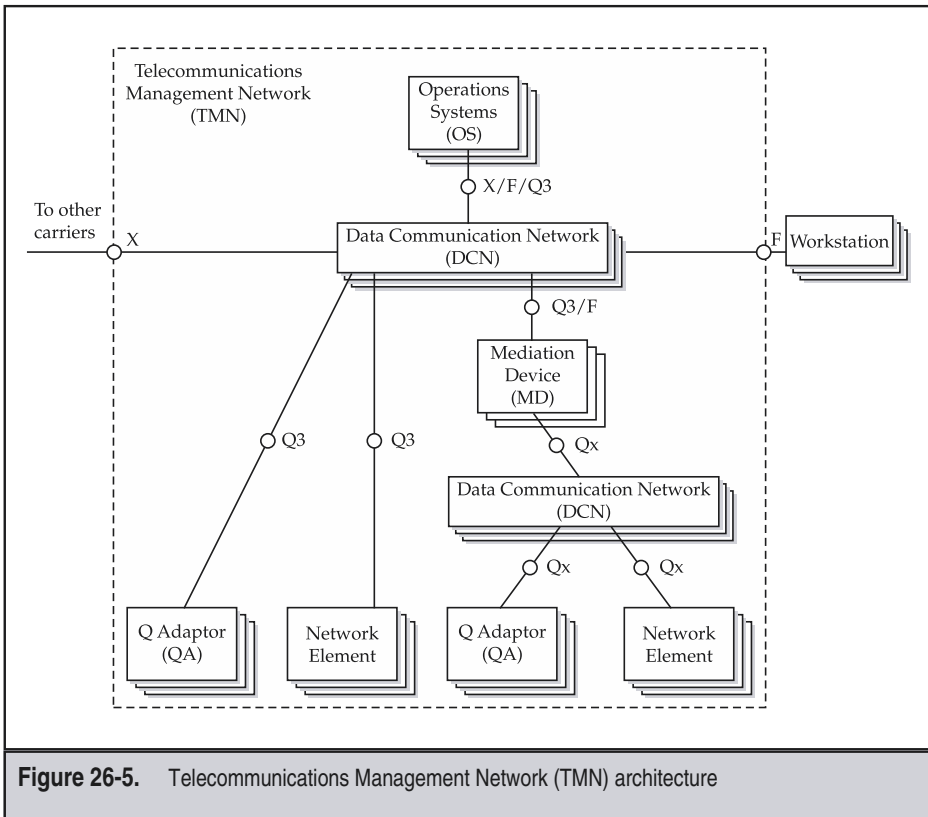


Figure 26-5. Telecommunications Management Network (TMN) architecture

ITU-T Generic Transport Network Architecture

Beginning in the early 1990s, the Telecom Information Networking Architecture Consortium (TINA-C) developed methods for integrated management of all parts of a communication network by applying principles from software integration, Open Distributed Processing (ODP) and Distributed Communication Environment (DCE), and most importantly, object-oriented design. However, the principal impact of TINA was influence on other standards. Inheritors of the concepts of TINA include the ITU-T G.8xx series of recommendations, specifically Recommendation G.805, covering generic functional architecture of transport networks, which modeled the protocol and management relationships of ATM, including the TDM Plesiochronous and Synchronous Digital Hierarchies (PDH and SDH), as well as Recommendation G.803, which covered the architecture of transport networks based on the Synchronous Digital Hierarchy (SDH). These documents strive to describe a functional architecture of transport networks in

a technology-independent way. This generic functional architecture is used as the basis for an intertwined set of functional architecture recommendations for ATM and SDH transport networks. It provides a basis for a series of recommendations for management, performance analysis, and equipment specification.

The G.805 recommendation defines a recursive client/server architecture for relating one functional layer of a transport network to another using a graphical method illustrated in the example of ATM carried over an SDH higher-order path (HOP) shown in Figure 26-6. It is important to note that this layering relationship is not equivalent to the data communications layered models described in Chapter 5, such as OSI, IP, or SNA. In-

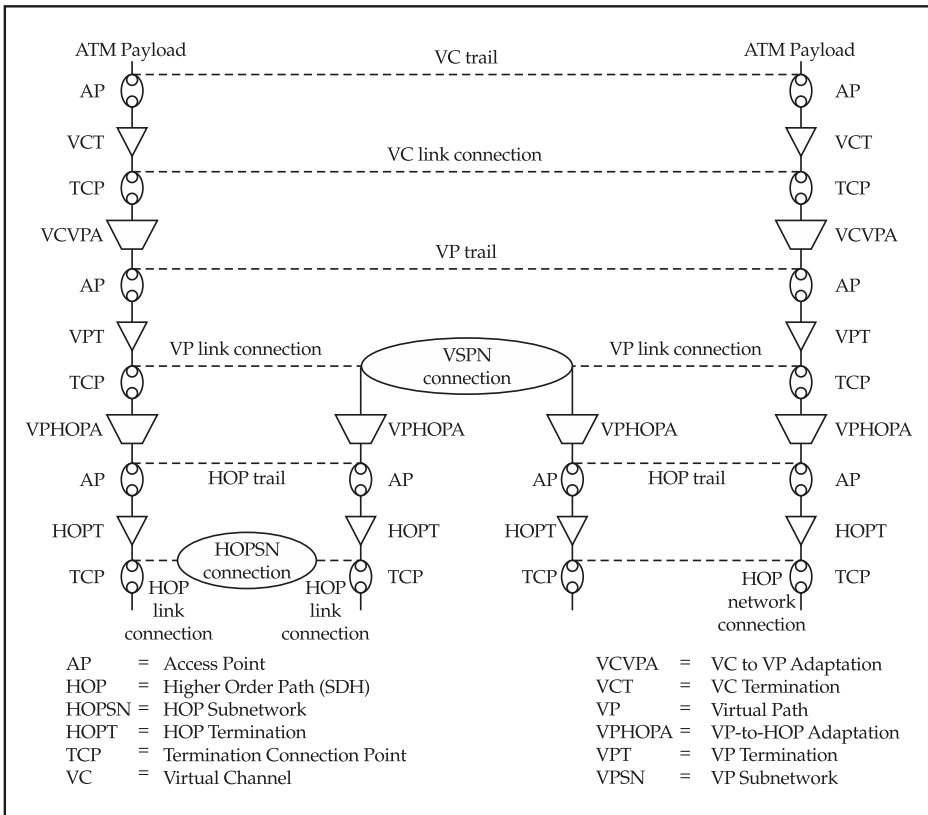


Figure 26-6. ITU-T G.805 generic transport architecture applied to ATM and SDH

instead, the layering defined in G.805 is focused on identifying the adaptation of one set of characteristic information so that another layer can transport it, as well as on describing how monitoring is performed. Starting from the top, an ATM payload at a pair of access point (AP) references creates a VC trail, which is the characteristic information carried between two VC endpoints, as described in Chapter 11. The VC termination (VCT) function adds monitoring information on transmit to create a VC link connection for transmission to the corresponding termination connection point (TCP). On receipt, the VCT removes the monitoring information before delivery to the AP. The important point here is that in general G.805 terminology, a connection carries monitoring information while the corresponding trail does not.

Continuing the example of Figure 26-6, there will often be a virtual channel (VC) to virtual path (VP) adaptation (VCVPA) function where many VCs are multiplexed into a VP. There is then a VP trail between access points, which in the example is composed of two VP link connections connected via a VPN subnetwork (VPSN). As before, the VP termination (VPT) functions insert and remove VP-level monitoring information on the VP link connections, which allows monitoring of segments of the end-to-end VP trail. Finally, as shown in the lower left-hand corner of the figure, the SDH HOP continues a similar recursion. The recursion continues on down through the SDH multiplex and regeneration section layers of the SDH layered management model, but it is not shown at the bottom of the figure in the interest of brevity.

When implemented correctly, this generic model of the handling and reporting of faults in a lower server layer instead of that in the higher client layers can significantly reduce the number of alarm messages sent to an NMS. For example, a failure in an SDH layer (e.g., an HOP link connection) would be indicated to the VP link connection layer, which would be able to correlate defects detected on many VPs to the single SDH layer failure. Chapter 28 provides more details on the specifics of this protocol interaction. These G.8xx series recommendations provide a foundation for building integrated systems that empower the operations groups to manage different transport network technologies as a single system.

ITU-T Recommendation Y.1710 uses G.805 as the basis for a set of requirements for managing the forwarding component of MPLS networks. At the time of writing, a draft recommendation Y.1711 was in development to define protocol specifics for monitoring, reporting alarms, implementing loopback, and measuring performance in a manner similar to that defined for ATM in ITU-T Recommendation I.610, which we summarize in Chapter 28. This approach assumes that MPLS is a transport, connection-oriented technology, which as described in Chapter 14 is a valid assumption for only some MPLS control modes in specific contexts.

However, at the time of writing, the IETF was working on competing approaches to managing MPLS networks based upon modifications to IP-based management protocols. Furthermore, although these protocols are not approved as a standard, a number of vendors have implemented them to provide a means for service providers to better manage deployed MPLS networks. In order to give a balanced view of the options being developed to manage MPLS networks, we summarize the IP-based management tools in Chapter 27.

ATM Forum Network Management Architecture

Figure 26-7 depicts the ATM Forum's network management reference architecture [ATMF M4 View], which identifies five distinct management interfaces. Interfaces M1 and M2 define the interface between a private network management system for one or more customer sites covering private networks and ATM end stations. The M3 interface allows public network carriers to provide standardized Customer Network Management (CNM) services from their management applications to a private management application. The M4 interface targets standardization of the interface to switches and element managers. M5 provides the management interface between different carrier's network management systems. ATM Forum work has concentrated on the M3 and M4 interface specifications.

What is the state of standardization and use of protocols at these reference points? Because SNMP is widely deployed by end users, the M1 and M2 interfaces embrace SNMP-based specifications defined by the IETF. These include relevant standard MIBs for transmission interfaces and the AToM MIB described in the next chapter. The M3 Customer Network Management (CNM) interface gives a customer a view into its carrier's network, including physical port status, VPC/VCC status, order parameters, and selected performance metrics. Several carriers have also deployed an M3-type interface to allow customers to dynamically control their services. Since the M4 interface provides network-wide and single element-level views for public ATM networks, it is the point where the private network manager and the carrier must be able to cooperatively control and monitor ATM service. The ATM Forum NM workgroup has focused on the M4 interface, having defined an SNMP MIB [AF M4 SNMP] and a CMIP MIB [AF M4 CMIP] that we further describe in the next chapter. Finally, the M5 interface targets the complex area of automatic management connections between carrier network management systems. The intercarrier M5 interface is perceived as the most complicated of the management interfaces in that it covers all of the TMN X interfaces described earlier at the network management and service management layers, as applied to ATM technology.

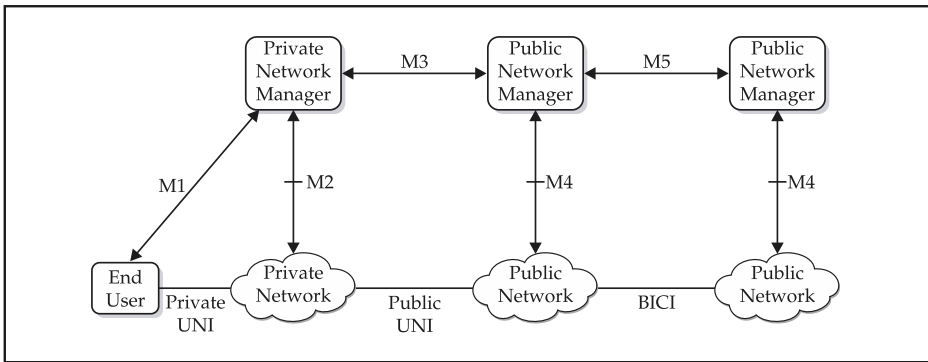


Figure 26-7. The ATM Forum management interface reference architecture

REVIEW

This chapter introduced a model of Operations, Administration, Maintenance, and Provisioning (OAM&P) functions and how they interact to meet the overall needs of network users. The text then covered specific issues that ATM and MPLS bring to the area of OAM&P. Next the discussion introduced standards from OSI, the ITU-T, the ATM Forum, and the IETF. The OSI architecture defines the functions of Fault, Configuration, Accounting, Performance, and Security (FCAPS). The ITU-T defines a physical and logical Telecommunications Management Network (TMN) architecture, which lays out a common structure for managing transmission, voice, and data networks. Also, the ITU-T generic transport architecture defines a recursive, extensible methodology for describing the relationships between various functional layers, which may be applicable to at least parts of MPLS. The chapter concluded with the ATM Forum's network management architecture, which defines a structure for MIB definitions.



CHAPTER 27



Network Management Protocols and Management Information Bases (MIBs)

This chapter summarizes the two major network management protocols used in ATM and MPLS as defined by the IETF, the ITU-T, and proprietary vendor implementations. The text then discusses the considerations involved in choosing the right protocol for a particular network. We first describe what an MIB is (another unfortunate acronym clash, since here it does not mean “Men in Black”) and then give a summary of ATM and MPLS Management Information Bases (MIBs) to summarize management support available for ATM and MPLS interfaces, switches, routers, and networks. We then summarize management tools used in IP networks that are being used to manage deployed MPLS networks.

NETWORK MANAGEMENT PROTOCOLS

This section describes network management protocols and their applications. We begin with the IETF’s Simple Network Management Protocol (SNMP), followed by the ITU-T’s Common Management Interface Protocol (CMIP). In network management parlance, databases are Management Information Bases (MIBs), so we introduce this concept as well.

IETF Simple Network Management Protocol (SNMP)

The IETF network management philosophy, protocol, and database structure (called SNMP for short) are widely used in the data communications industry. This section begins by defining the overall object-oriented network management model and summarizes the SNMPv2 and SNMPv3 messaging protocols.

Object Model of Network Management

SNMP is the protocol part of the IETF’s network management (NM) philosophy; however, it alone will not manage your network [Cikoski96]. The IETF has not invested in the elaborate information models and computational models that the ITU-T specifications lay out. Instead, most of this information was, and still is, passed by tradition and word of mouth between implementers. On the other hand, this works rather well, because the IETF is an open organization, and the IETF membership devoted to network management is accessible and has a strong mentoring tradition. This experience is recorded in some informative books written by the specification and MIB designers [Rose 95, Rose 96, Perkins 97].

Figure 27-1 illustrates the key components of an SNMP-based NM system. Typically, a single computer system interfaces to a number of network elements. The NM connections may not be physical, and indeed they may be carried in-band by the underlying network itself. SNMP is only a basic set of messages for monitoring and controlling the state of the network elements. The intelligence of an NM system lies in understanding what the state variables (called Management Information Base [MIB] objects in SNMP parlance) in the network elements actually mean. Here, the collective body of knowledge recorded in RFCs, in the archives of the IETF workgroup mail lists, in white papers, or in vendor

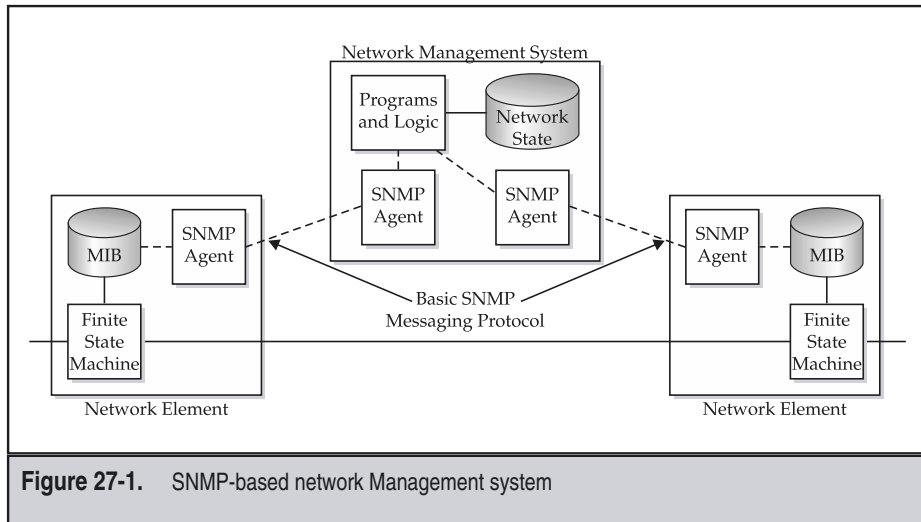


Figure 27-1. SNMP-based network Management system

documentation is the linchpin to successful management of network elements. In fact, a difference in interpretation of the meaning of MIB elements is the greatest interoperability problem encountered between agents and managers.

The condition of a physical interface—active or inactive—is an example of a state variable modeled by a MIB object. Continuing this example, knowing that two physical interfaces are supposed to be connected together requires intelligence. Unfortunately, the defined semantics provided by the Structure of Management Information (the SMI) does not provide for relating an element in one table with a reference in another table; even when they refer to the same thing in the real network. MIBs typically show this relation via indexed tables and references to common indexes in a special MIB called *ifIndex* (interface index table). Nevertheless, externally supplied configuration information is often mapped by SNMP managers to unchanging MIB references in *ifIndex* and in cross-connect tables in the MIBs. Keeping track of what is supposed to be connected and whether that connection is active is an important overall network state variable in a connection-oriented network.

Many network management engineers model more complex network conditions as finite state machines because of the precision and accuracy this provides in capturing the behavior of the elements, under stimulus, over time. In essence, the state machine model provides what is called the computational model in the TMN and TINA-C architectures described in Chapter 26. Other tools include rule engines and expert-systems knowledge bases. Advanced network management systems capture relationships using these tools and use them to know what variables to compare under what circumstances. These tools are also used to filter SNMP TRAPs and associate them with past operational and repair experience.

SNMP Message Types

Amazingly, SNMP allows a complex management system to monitor and control a large network of complex devices using only the following five simple messages, which operate on objects in a Management Information Base (MIB):

- ▼ GET retrieves a particular object.
- GET NEXT retrieves the next object in the management database structure.
- SET modifies a management object.
- RESPONSE is always paired with a stimulus SET, GET, or GET NEXT message.
- ▲ TRAP is the unsolicited notification of an event, such as a failure or a system restart.

SNMP normally operates over the User Datagram Protocol (UDP), which then usually operates over IP in the Internet Protocol (IP) stack but may operate over some other protocol. Note that the UDP/IP protocol does not guarantee delivery of packets, because there is no retransmission or sequence numbering. This means that if a TRAP were lost, then a management system relying on TRAPs alone would miss a failure notification. Real systems resolve this problem by periodically sending out traps for major failure conditions, or else the management system periodically polls the network elements for status.

SNMP utilizes a variant of Abstract Syntax Notation 1 (ASN.1) to define a MIB as a data structure referenced in SNMP messages. The SNMPv1 SMI (Structure of Management Information) allows MIB definitions of objects in primitives such as strings, integers, and bit maps, using a simple form of indexing. Each object has a name, syntax, and encoding. The MIB variables have a textual Object Identifier (OID), which is commonly used to refer to the objects. The MIB objects are defined as a tree structure that allows organizational ownership of subtrees to be defined. Every registered MIB in the entire world is a member of a single, official registration tree controlled by the IETF Internet Assigned Numbers Authority (IANA). The branches of the tree are identified by a dotted decimal notation. For example, the prefix of the subtree registered to the ATM Forum is 1.3.6.1.4.1.353. Vendor MIBs, also called Enterprise MIBs, use a branch of the tree where every vendor registers for a limb and then owns all of the subtending branches and leaves.

SNMPv2 and SNMPv3

The IETF has been standardizing improvements to SNMP to extend its applicability and correct deficiencies. SNMPv1 was standardized in 1989 in RFC 1098, which was rendered obsolete by RFC 1157 one year later. After six years, the IETF released the next version, called SNMPv2, as defined in RFCs 1902 through 1910. SNMPv2 uses a new syntax called SMIV2 for describing and naming objects in MIBs. Furthermore, extensions to the type of OIDs extend MIB features and smooth out problems experienced by SNMPv1 implementers. All MIBs that can respond to SNMPv2 commands issued by the manager refer to RFC 1905, which defines the protocol used for network access to managed objects. All

SNMP MIBs reference the core definition of RFC 1213, which defines MIB-II, the core set of managed objects for the Internet suite of protocols. Another important enhancement of SNMPv2 was the introduction of two new protocol messages: GETBULK and INFORM.

GETBULK overcomes the limitations in SNMPv1 that required manager applications to issue many GETNEXT commands to traverse a table. A single GETBULK request can return a large range of MIB values, which improves efficiency when retrieving a MIB table when compared with a series of GET and GET NEXT transactions. This is especially important in complex ATM and MPLS MIBs on a large switch or router, since good network management requires retrieval of large volumes of data.

INFORM messages provide unsolicited event notifications. They are similar to SNMPv1 TRAPs, except that the sender expects the receiver to respond. This provides for confirmed alarm notifications, unlike TRAPs, which are unacknowledged. Since a manager must respond to an INFORM, often only the most critical events, such as those that interrupt service, should be made as INFORMs instead of TRAPs. If the network element agent has logic to send a single indication of a problem, and if it is also intelligent enough to know it need not send another indication, an INFORM greatly lessens the burden on managers when compared with handling repetitive TRAP notifications (e.g., periodic link down notifications). Another effective design technique is for agents in network elements to send an alarm notification that covers a range of affected logical elements in a single notification. This is a fundamental premise of the ITU-T G.805 architecture, of which Chapter 28 provides an example for ATM.

Many developers of enterprise MIBs use SMIV2 [RFC 2578, RFC 2579], which provides, among other advantages, 64-bit counters. Since ATM switches and MPLS label-switching routers support high-speed interfaces, the older SNMPv1 SMI 32-bit counters wrap around much sooner in real time, creating the need for software to sample them more frequently to avoid wraparound. For instance, counting ATM cells on OC12, the 32-bit counter wraps around in less than an hour.

Unfortunately, the SNMPv2 standards development process failed to agree on a mechanism for SNMP security, and this left the door open for a sequel to this popular network management protocol. Since security was a clearly part of the working group's charter, this was a great disappointment. After a brief hiatus, a new workgroup attacked the security problem anew, and the initial standards for SNMPv3, the next version, were issued in 1999 in RFCs 2570 through 2575. At the time of writing, a replacement for RFC 2570 was nearing completion. SNMPv3 provides security between agents and managers, allowing each to authenticate the identity of the other and control which transactions are allowed. Additionally, SNMPv3 cleaned up a few problems with SNMPv2, such as full specification of proxy agents and clear identification of the source identity of the agent when a manager receives a TRAP.

There are many commercially available SNMP toolkits, managers, and related products. Classic managers include SunNetManager, IBM NetView, and HP Openview, which defined the initial SNMP management industry. Just about every network element vendor provides an SNMP-based management element, and some of the larger switch and router vendors provide SNMP-based element managers, which often "plug in" to these SNMP management systems.

ITU-T Common Management Interface Protocol (CMIP)

As described in Chapter 26, CMIP is the approved protocol for use in the ITU-T Telecommunication Management Network (TMN), although other protocols, such as CORBA, are also considered applicable. It is sufficiently mature that a wide range of vendor toolkits are available that are essential aids in building element agents and TMN managers. However, CMIP management is not a do-it-yourself activity, and it is best to contract out development of such a system to an experienced vendor. A great deal of effort is required to properly subset the features required and to align the information models derived from the ITU-T Recommendation M.3100 architecture to the target technology. The publications of the ATM Forum network management workgroup have made inroads in this area, but at the time of writing, CMIP was not actively being pursued for management of MPLS networks.

Like SNMP, CMIP uses a manager/agent model for defining the interface between management systems and network elements. An important difference is that CMIP requires a more active and function-rich role for the TMN manager and the CMIP agent. The manager actually creates object instances in the agent and essentially programs the functions desired into the behavior of the agent. The agent then supplies the information and functions to the manager as packages of information. As described in Chapter 26, CMIP is part of the Q3 interface specification of TMN with a standard protocol stack. In ATM networks, in-band transport of CMIP uses AAL5; while out-of-band CMIP transport can be IP or any other data communication network.

CMIP, like SNMP, has a short list of protocol commands; however, the behavior of agent and manager in executing these commands is complex. CMIP acts on GDMO defined objects, which inherit characteristics from more generalized objects as a very powerful tool for designing an NMS. The CMIP information model is therefore much richer in features and better organized than that provided by SNMP. CMIP operates in a management environment where the semantics of the objects are well defined.

The CMIP SMI uses templates to specify the behavior of the manager and the agent, when issuing and responding to these commands. These templates qualify the characteristics of object classes; that is, these define how managed objects relate and respond to management requests. ACTION and NOTIFICATION templates program an object in regard to the respective CMIP commands, described next. Generally, CMIP commands are always confirmed; that is, the manager and agent are aware that any specific command is sent, received, and executed. Therefore, CMIP commands have some of the important features of transactions. These are the CMIP command primitives:

- ▼ M-CREATE creates an instance of a managed object, which initializes the specific logical image of an element that management tracks and controls.
- M-DELETE deletes object instances, including all instances inherited from the deleted object.
- M-GET retrieves the values, that is, attributes, of objects specified in the command content.
- M-CANCEL-GET stops an ongoing M-GET command.

- M-SET modifies values of attributes in objects specified in the command.
- M-ACTION invokes the performance of an administrative or operational function, as defined in a corresponding ACTION template.
- M-REMOVE removes data from attributes.
- ▲ M-EVENT-REPORT communicates an event (e.g., an alarm) to the manager as defined in a corresponding NOTIFICATION package. Its delivery is not confirmed.

Actual behavior of a command is determined by the content carried with the command and the semantics of the objects on which it works. The preceding commands are initiated by the manager on the agent, with the exception of M-EVENT-REPORT, which is generated by the agent.

The development of the GDMO interface library is extremely important for CMIP to function to expectations. Vendors are now providing CMIP agents, and off-the-shelf CMIP managers are available in the market. However, do not expect the plug-and-play behavior of SNMP manager/agent MIBs. Plan to expend considerable effort to turn up a functioning management system. Unless both manager and agent are provided by the same vendor, have the switch vendor and the manager demonstrate intercommunication in a management trial.

Proprietary Network Management Protocols

Often one hears that open interfaces are good and proprietary interfaces are bad. The normal argument is that open interfaces lead to interoperable implementations, while proprietary implementations can never interoperate. In fact, business drivers at one time promote open interfaces but at other times drive users to select proprietary solutions. In actual experience, any specific stack of OSS applications deployed in a large network will contain both standard and proprietary network management protocols.

Some proprietary protocols become open interfaces. If an owner publishes and licenses proprietary interfaces, they may, by dint of widespread implementation by different organizations, become *de facto* standards. For example, a widely used *de facto* standard is the API to HP's Network Node Manager (NNM), an SNMP-based application. Early on, HP published a private API and provided a toolkit allowing equipment vendors to use HP NNM. This approach was commercially successful and has been mimicked by other network management vendors.

Why do vendors invent proprietary interfaces instead of following published standard? Sometimes a particular feature is not defined in the standards, and a proprietary MIB is necessary to describe proprietary features, or valuable new capabilities, whose management is not yet standardized. Another reason is that sometimes an important function is awkward to implement using a standard protocol. For example, provisioning, which requires the notion of a reversible transaction, is awkward to implement in SNMP. At other times, vendors invent proprietary interfaces because it is cheaper or easier to do so. For example, the fact that the ITU-T CMIP specification was very large and complex and proved very costly in initial implementations by big service providers motivated the

development of simpler, proprietary solutions. Also, proprietary protocols are sometimes perpetuated as competitive differentiation.

Considerations on Choice of Network Management Protocol

There is general agreement that SNMP is best for CPE and private networks, while CMIP/CMISE is more appropriate for carrier applications. The addition of security in SNMPv3 makes it more appropriate for service provider networks. Although there is still active debate on the usefulness of SNMP for large carrier environments, currently more of these networks using standard interfaces employ SNMP than those that either use or plan to use CMIP. In fact, many early ATM switches implemented SNMP because it was simpler and easier to achieve interoperability than with the CMIP protocols.

Some vendors support hybrid implementations where some functions, statistics capture, for instance, are performed by SNMP, and other functions rely on proprietary interfaces. Where CMIP is deployed, it is being rolled out incrementally; for instance, SNMP TRAPs handle alarms, while CMIP supports complex provisioning actions because it supports the transaction model. Another example employed by many MPLS LSRs is use of SNMP for read-only operation along with reliance on secure file transfer or terminal sessions for provisioning actions. In general, more traditional carriers have made more use of CMIP and less of SNMP.

Management Information Bases (MIBs), first widely used in local area network and Internet environments, have achieved a great degree of interoperability using the SNMP protocol. Therefore, it is likely that SNMP-based management will be the de facto standard for the network and higher layers, particularly for IP-derived technologies like MPLS. Note that compiling and loading a MIB into a manager and using that MIB are different problems. Only part of that information is in the semantics of the MIB object definitions, with more information often given in the introductory sections of the MIB than in the comments associated with each object definition.

Even the most successful vendors of proprietary network management protocols and nonstandard OSS applications are under pressure to standardize, since service providers often have equipment from multiple vendors. Proprietary applications can use gateways to other applications with standard interfaces, which provide a standards-based interface to the outside world. Today, wide acceptance of toolkits and off-the-shelf management application stacks by special tool vendors both for SNMP and for TMN have made the time to deploy standards as quick and as cost efficient as proprietary methods once were. Keeping old proprietary management code can be expensive. For these reasons, expect to see more vendors support standard open interfaces.

ATM MANAGEMENT INFORMATION BASES (MIBS)

This section summarizes standard MIBs as examples of the types of information that can be accessed and manipulated in ATM interfaces, end systems, switches, and networks.

ATM Forum Integrated Local Management Interface (ILMI)

When the ATM Forum created the Interim Local Management Interface (ILMI) in 1992, it anticipated that ITU-T standards would create a final interface management solution. Four years later, the Forum changed the initial “I” in the acronym to Integrated, since the Integrated Local Management Interface (ILMI) [AF ILMI 4.0] now performs the following critical functions:

- ▼ Basic configuration information
- PVC status indication in FR/ATM service interworking (see Chapter 17)
- ILMI connectivity detection and auto neighbor discovery
- Address registration for SVCs and PNNI (see Chapter 13)
- ABR attribute setting for PVCs
- ▲ Auto-configuration of a LAN Emulation Client (LEC) (see Chapter 18)

ILMI Configuration

Figure 27-2 illustrates the reference configuration for the ILMI. ATM Interface Management Entities (IMEs) communicate using the ILMI protocol, which is based upon SNMP operating over AAL5, each in turn over physical or virtual links. IMEs may operate in either a user, network, or symmetric mode. Each ATM End System (ES), and every network that implements a Private Network UNI or Public Network UNI, has an ILMI Management Entity (IME) responsible for maintaining the information and responding to SNMP commands received over the ATM UNI. The information in the ILMI MIB can be actually contained on a separate private or public Network Management System (NMS) or may be accessed over another physical interface. NMSs may also be connected to networks or end systems by other network management interfaces.

The code used to write SNMP managers and agents was familiar to the authors of ILMI, but the ILMI departs in several key ways from the SNMP model. SNMP’s manager

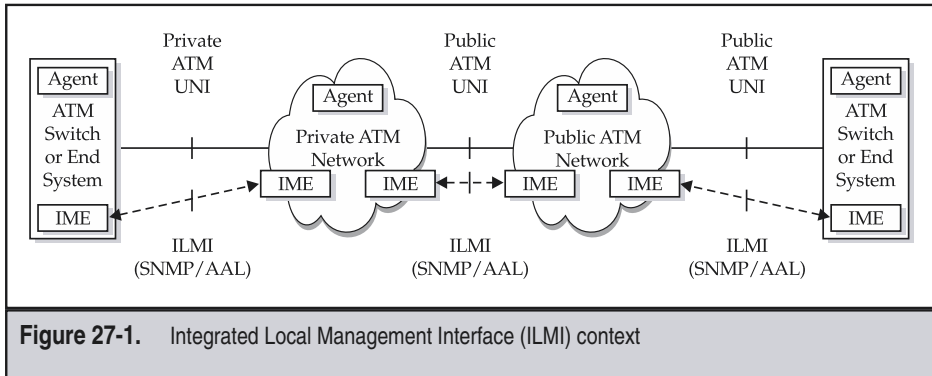


Figure 27-1. Integrated Local Management Interface (ILMI) context

agent administrative model is replaced in ILMI by a symmetric association between the user side and the network side, each of which can SET variables or GET (query) variables in the other's MIB. For instance, the network side SETs address prefixes in the user side and the user side SETs addresses in the network side address registration table as described in Chapter 13. Originally, ILMI agents could also double as SNMP agents and also talk to managers. However, some confusion resulted in initial implementations that delayed widespread support for ILMI. These confusions were identified as problems in the early implementations and were resolved in the ATM Forum standards with the ATM Forum ILMI 4.0 specification. A partition of the ILMI MIBs from the SNMP agent MIBs was clearly required. Today a manager who wishes to find out information in the ILMI MIB must use a "shadow" ILMI MIB implemented in the SNMP agent space of the ATM element. Early experience also made clear that implementations of the ILMI would benefit from using a set of linked state machines at the user and network sides to carry the true semantics of interaction of command exchanges.

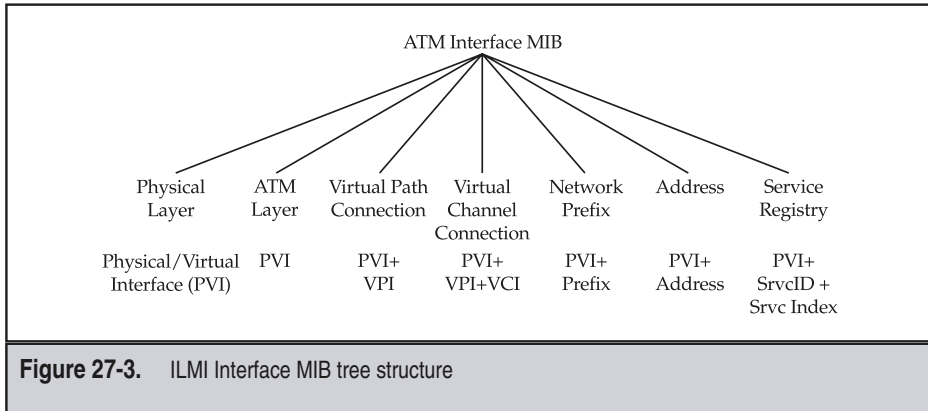
For the ATM layer management interface, a default value of VPI=0, or VCI=16 for the ILMI, was chosen because CCITT/ITU reserved VCIs 0 through 15 (i.e., the first 16) for future standardization. Alternatively, another VPI/VCI value can be manually configured identically on each side of the UNI for ILMI use. Use of this method is undesirable, since it is not automatic and is one more configuration parameter that can be incorrectly set. The ILMI operates over AAL3/4 or AAL5 as a configuration option, with support for a UDP/IP configurable option. Therefore, in order for IMEs to interoperate, the AAL (either 3/4 or 5) and the higher-layer protocol (either UDP/IP or Null) must be chosen.

ILMI Management Information Base

Figure 27-3 illustrates the ILMI 4.0 Interface Management Information Base (MIB) tree structure and its index structure. Three versions of the ILMI MIB have been specified by ATM Forum UNI specification version 2.0 [AF UNI 2.0], version 3.1 [AF UNI 3.1], and version 4.0 [AF ILMI 4.0]. The version 3.1 ILMI MIB is backward compatible with the version 2.0 MIB, while the version 4.0 MIB deprecated (i.e., deleted) many objects from the version 3.1 MIB, since other standards now define these objects. This section summarizes the content of the version 4.0 MIB. The IME indexes each branch of the MIB tree via a Physical/Virtual Interface (PVI) as indicated in the figure. The following paragraphs summarize the basic content and function of each of these MIB groups for version 4.0 of the ILMI.

The Physical Layer MIB information contains only an interface index and adjacency information, with historical information deprecated, since it is now contained in other MIBs.

The ATM Layer MIB information contains a number of objects related to the use of VPI and VCI values, including the maximum number of active VPI and VCI bits, the maximum number of allowed VPCs and VCCs, the current number of configured VPCs and VCCs, the maximum VPI for an SVPC, and the maximum SVCC VPI and minimum SVCC VCI values. It also contains a public/private interface type indicator, a device type (either user or node), the ILMI version, the UNI signaling version, and the NNI signaling version.



The Virtual Path Connection (VPC) MIB information contains, for each VPC: the VPI value, the operational status (either up, down, or unknown), the transmit and receive traffic descriptors, a best effort indicator, the ATM Forum service category, and ABR operational parameters.

In a similar manner, the Virtual Channel Connection (VCC) MIB information contains, for each VCC: the VPI and VCI value, the operational status (either up, down or unknown), the transmit and receive traffic descriptors, a best effort indicator, the ATM Forum service category, and ABR operational parameters. It also contains an indication for the transmit and receive directions of whether the EPD/PPD frame discard mechanism described in Chapter 22 is implemented.

As described in Chapter 13, address registration using ILMI is a key component of automatic configuration of Private Network-Network Interface (PNNI) reachability information in the ATM switched virtual connection (SVC) capability. Basically, address registration allows the network to communicate to the user which address prefixes are valid on the User-Network Interface (UNI). The user can then register the valid remaining portions of the address(es) present locally. It also provides source authentication for virtual private networks, since the originating switch may screen the calling party information element in the SETUP message against the set of registered addressed prefixes.

In support of these functions, the Network Prefix MIB information contains a network prefix and a network prefix status. Also, the Address MIB information contains the ATM address, the ATM Address Status, and the ATM Address Organizational Scope Indication.

The Service Registry MIB information portion of the ILMI provides a general-purpose service registry for locating ATM network services, such as the LAN Emulation Configuration Server (LECS) and the ATM Name Server (ANS).

The ILMI also uses the standard systems group by reference, which supports things such as identification of the system name, and the time that the system has been up. The

systems group also provides standard TRAPs, such as when a system is restarted or an interface failure is detected.

IETF ATOM MIBs

The IETF initially standardized support for ATM in the ATOM MIB working group in RFC 1695 in 1994. Five years later, RFCs 2512 through 2515 replaced this standard and defined additional MIB components that aligned with ATM Forum ILMI specifications, defined new capabilities, and built on implementation experience. In the remainder of this section, we summarize the content of these MIBs and how they can be used to manage an ATM device or network.

The ATOM MIBs use the terminology described in Chapter 11 of an ATM VP or VC connection (VPC or VCC) being made up of one or more VP or VC links (VPL or VCL). It also covers the way that a VPC or VCC is established, as either a permanent, switched, or semipermanent connection (i.e., PVC, SVC, or SPVC) as described in Part 3. It also contains traffic and QoS parameters at the UNI and NNI, as described in Chapter 20. The scope of the ATOM MIBs covers objects used to manage ATM interfaces; ATM virtual links; ATM cross-connects; AAL5 entities; and AAL5 connections supported by ATM hosts, switches, and networks. The MIB arranges the managed ATM objects into a number of tables, mentioned in the following along with a brief description of their contents and potential uses. As described in Table 1 of RFC 2515, there is some overlap with ILMI 4.0, but the intent is to achieve identical semantics and syntax.

The ATM interface configuration table in RFC 2515 contains configuration information for ATM cell layer information on a local ATM interface. This includes active VPI/VCI fields, maximum number of VPC/VCCs, ILMI VPI/VCI values, neighbor system information, the maximum number of VPI/VCI bits, and the ATM address of the interface. This is information in addition to that contained in the standard ifTable interfaces group, which contains administrative and operational status, the number of packets and octets sent and received, and counts of errors detected.

The ATM interface DS3 PLCP and TC sublayer tables in RFC 2515 provide performance statistics of the DS3 PLCP and TC sublayer (see Chapter 11) of local interfaces. These include the alarm state along with counts of out-of-cell delineation events and errors.

The AAL5 connection performance statistics table in RFC 2515 contains information for each AAL5 virtual connection supported by an AAL5 entity in an ATM switch or host. AAL5 entities have an ifTable MIB similar to that defined for ATM interfaces, with objects specifically defined for AAL5. These include the MTU size, AAL5 CPCS CRC-32 errors, and reassembly timeout errors.

The ATM interface virtual path and channel link (VPL/VCL) and VP/VC cross-connect configuration tables in RFC 2515 model bidirectional ATM virtual links and cross-connects. ATM hosts, switches, and networks implement the VPL/VCL tables, while only ATM switches and networks implement the VP/VC cross-connect tables for service provider and customer network management purposes. The ATM virtual link tables are used to create, delete, or modify ATM virtual links and, when used in conjunction with PVC cross-connect tables, can be used to configure an ATM PVC. For an ATM SVC or SPVC, the VPL and VCL tables are used in conjunction with an SVC cross-connect

table in a separate MIB that contains the configuration established by the ATM signaling protocol. The tables contain the administrative and operational state of VP/VC links and cross-connects, and in the case of VP/VC links, they also contain information about the AAL type and AAL parameters.

RFC 2514 details the content of the ATM traffic descriptor table of RFC 2515 in terms of objects that detail the ATM traffic parameters and service category (see Chapter 20) as referenced by a particular VP/VC link table entry. This data structure allows more than one VP/VC link to use the same combination of ATM traffic and service category parameters.

The accounting MIBs of RFC 2512 and 2513 define managed objects for controlling the collection and storage of accounting information for connection-oriented networks, such as ATM and Frame Relay. In particular, they provide an SNMP administrative interface to control the bulk transfer, generally TFTP or FTP, of accounting statistics collected by an ATM switch to a management repository. This MIB can be used to gather data for billing and/or reporting systems.

The AToM MIB and Trunk MIB working groups published, in RFC 2493, a generic MIB module for using performance history based on 15-minute intervals. This MIB provides an RMON-like history table for performance measurement statistics commonly used in connection-oriented transport networks such as TDM and ATM. Typically, there is a requirement for a network element to store 24 hours worth of data in 96 15-minute bins. The statistics for an individual bin could be collected by the GET command, or all statistics for a 24-hour period may be collected via a single GETBULK command.

Other ATM MIBs

The use of SNMP means that all of the IETF-defined MIBs for physical interfaces (e.g., RFC 2558 for SONET/SDH) can be used in ATM without change. The ATM Forum has defined a number of additional MIBs in support of specific functions as summarized in Table 27-1. A network management system using SNMP can utilize these MIBs to manage devices performing these functions.

The ATM Forum network management working group also published a number of requirements documents on M4 security and logical MIB (af-nm-0103.000), automatic configuration of PVCs (af-nm-0122.000), and management of path and connection trace (af-nm-0153.000), as well as usage measurement (af-nm-0154.000).

MPLS MANAGEMENT INFORMATION BASES (MIBS)

Since MPLS is IP-based [RFC 3031], the principal focus of MIBs has been using SNMP. The use of SNMP means that all of the IETF-defined MIBs for physical interfaces (e.g., RFC 2558 for SONET/SDH) can be used for MPLS without change. At the time of writing, the IETF was defining MIBs for MPLS-related functions in several other areas, namely, management of a label switch router (LSR) and its components, traffic engineering, VPNs, and pseudo-wire emulation. We briefly summarize important areas of these MIBs and how they can be used to manage MPLS-based networks. For up-to-date information, the reader should consult the IETF working group pages referenced herein.

Function(s) Supported by MIB	Reference
ATM Data Exchange Interface (DXI) v1.0	af-dxi-0014.000
Private Network-Network Interface (PNNI) v1.0	af-pnni-0055.000
PNNI V1.0 SPVC MIB Addendum	af-pnni-0066.000
Inverse Multiplexing over ATM (IMA) v1.1	af-phy-0086.001
Circuit Emulation Service (CES) 2.0 MIB	af-vtoa-0078.000
ATM Remote Monitoring SNMP MIB	af-nm-test-0080.000
CMIP Specification for the M4 Network Element Interface v2	af-nm-0027.001
M4 Network View CMIP MIB Spec v1.0	af-nm-0073.000
SNMP M4 Network Element View MIB	af-nm-0095.001

Table 27-1. Other ATM-Related MIBs

Label Switch Router (LSR) and Related MIBs

Recall from the introduction to SNMP at the beginning of this chapter that the management of a device itself is of paramount importance. Toward this end, the IETF MPLS working group was working on a MIB for an MPLS label switch router (LSR), which is the basis for many of the MPLS MIBs [Nadeau 01]. This LSR MIB [Srinivasan 02a] models a label switched path (LSP) as a connection consisting of one or more incoming segments (in-segments) cross-connected to one or more outgoing segments (out-segments) as specified in a cross-connect table. It supports both manually configured LSPs as well as LSPs established by any MPLS signaling protocol, enabling/disabling MPLS on a per-interface basis, allocation of label space on a per-platform or per-interface basis, and the capability to configure label push and pop actions. With manual configuration, it supports point-to-multipoint and multipoint-to-point connections at an LSR. It provides for specification of LSP traffic parameters; such as mean and maximum rates as well as a maximum burst size. The MIB also contains performance counters (e.g., in and out octet, packet counts, as well as fragmentation, discard, and error counts) for the in- and out-segments, as well as per-interface counters.

Several other, related MIBs are being defined in the IETF MPLS working group. The Label Distribution Protocol (LDP) MIB [Cucchiara 01] allows a user to configure potential LDP sessions as well as monitor the status of all LDP sessions on an LSR. It also supports configuration for LDP using IP, ATM, or Frame Relay networks. The FTN MIB [Nadeau 02a] describes managed objects for specifying FEC (Forwarding Equivalence Class) to NHLFE (Next-Hop Label Forwarding Entry), abbreviated as FTN, mappings and corre-

sponding actions for MPLS. The MIB consists of three tables: the first defines the rules for matching incoming packets and the actions taken, the second associates these rules to specific interfaces, and the third provides performance counters for every active FTN entry on a per-interface basis.

Traffic Engineering (TE) MIBs

Recall from Chapter 10 that the original motivation for MPLS was to provide better traffic engineering support in Internet service provider backbones than could be achieved by IP routers overlaid on an ATM network. Toward this end, the IETF had several MIB efforts in this area at the time of writing.

The MPLS working group was working on a traffic engineering MIB [Srinivasan 02b] with a complementary focus. This MIB is designed to support configuration of point-to-point unidirectional MPLS tunnels, which could be configured as an interface. These MPLS tunnels could be manually configured on a hop-by-hop basis or set up via an MPLS signaling protocol, with loose or strict source routed hops. This MIB is intended for use in conjunction with an LSR MIB described earlier to manually configure tunnel segments. The LSR MIB is needed to determine performance of these tunnels and tunnel segments. The MIB objects in this MIB are a tunnel table for setting up MPLS tunnels in conjunction with the LSR MIB; a resource table; a table for tunnel specified, actual, and computed hop tables for strict and loose source-routed MPLS tunnel hops; and a table for specifying resource objects for tunnels signaled using CRLDP.

The traffic engineering working group has also been working on a Traffic Engineering (TE) MIB [Kompella 02]. It allows configuration of logical traffic tunnels, which are composed of one or more label switched paths. It allows assignment of one or more LSPs to such a tunnel, contains the per-hop information regarding each path, and monitors operational aspects of the tunnel, such as their operational and administrative states and counters indicating the number of octets and packets carried by the traffic tunnel. This MIB is complementary to the MPLS-TE MIB in that a logical traffic tunnel is important in operational networks to spread a large traffic aggregate across multiple LSPs using load balancing so that traffic can be more tightly packed onto trunks.

Multiservice PPVPN and PWE3 MIBs

At the time of writing, the IETF provider-provisioned VPN (PPVPN) and Pseudo Wire Edge to Edge emulation (PWE3) working groups had begun work on requirements and framework on MIBs for these applications that could be carried over MPLS or IP tunnels. The intent was that for MPLS-based tunnels, these applications would leverage the MIBs being defined in the MPLS working group [Nadeau 01]. We provide a brief summary on this work and refer readers interested in current information and more detail to the Web pages of these IETF working groups.

At publication time, there were drafts for MIBs for the BGP/MPLS aggregated routing and virtual router (VR) types of PPVPN (see Chapter 19). The BGP/MPLS VPN MIB [Nadeau 02b] assumes that MPLS is configured and operational and that LDP is used

between provider edge (PE) LSRs. It allows configuration of a virtual routing and forwarding (VRF) instance for each VPN, provides for assignment of physical or logical interfaces to a VRF, supports performance counters for a VRF, and also allows configuration and monitoring of VRF routing. On the other hand, a virtual router MIB takes a different approach, since in this style of PPVPN a PE is made up of multiple logically separate virtual routers. Thus, multiple logically separate management domains are necessary so that different management entities (e.g., customers) can manage the virtual routers and associated services. SNMPv2 community strings or SNMPv3 context names are used for this purpose. Each context, then, has separate access to logically separate MIBs for the routing protocol (e.g., OSPF, BGP), interfaces, alarms, statistics, and other items. The MIB also provides the means for a service provider to create and delete VR instances, as well as assign interfaces to a specific VR.

IP-BASED MANAGEMENT TOOLS FOR MPLS

As described in Chapter 10, the origins of MPLS are in IP, and the architecture assumes the presence of IP routing and signaling protocols [RFC 3031] in each LSR. Although not standardized yet by the IETF, at publication time operational IP over MPLS networks use the Internet Control Message Protocol (ICMP) and extensions as a form of vendor proprietary de facto standard. We summarize the IP-based ancestors of this technology, describe the currently implemented support for MPLS, and summarize the direction being pursued in the IETF.

ICMP PING and Traceroute

As introduced in Chapter 8, ICMP and ARP are an integral part of the Internet Protocol (IP) layer. ICMP messages are encapsulated in IP datagrams and are identified by protocol type and IP source and destination address (SA, DA). An ICMP message [RFC 792, RFC 950] has a four-octet header, with one-octet type, one-octet code, and two-octet checksum fields prior to any data specific to a particular ICMP message type.

The widely used packet Internet groper (i.e., “ping” in many operating systems) command uses two ICMP message types for echo and echo reply to allow a sender identified by the IP SA to determine whether a receiver identified by the IP DA is active. The sender populates the echo message (type 8) with a two-octet identifier and two-octet sequence number, along with some data to pad the datagram to a specified size, and sends it using IP to the destination. If the destination receives the echo message, it then returns the datagram to the sender by reversing the SA and DA fields in the IP packet header, and sets the ICMP message type to zero. Typically, the sender measures the difference in time between sending the echo message and receiving the echo reply with corresponding identifier and sequence number in a ping report.

The ICMP time exceeded message (type 11) is also quite important. As described in Chapter 8, each IP device must decrement the Time to Live (TTL) field in the IP header before forwarding the packet on the next hop interface. Furthermore, RFC 792 requires

that any node that receives a packet with a TTL of one discard the packet and generate an ICMP time exceeded message back toward the sender along with the IP header and 64 bits of the original datagram. Note that the source address of the ICMP message is the node where the TTL value expired. Traceroute is another widely used utility in management of IP networks that uses the ICMP time exceeded message. A source traces out the route to a destination by sending IP datagrams with successively higher TTL values, intentionally causing intermediate nodes to generate an ICMP time exceeded message with an SA field that indicates the IP address of the node along the path where the TTL expired. Figure 27-4a illustrates a simple example of the traceroute command without the MPLS extensions and the resulting output issued from router CE1. Most implementations report on the maximum, minimum, and average delay in units of ms, as shown in this example. Note how the MPLS hops through routers 2 and 4 are not revealed.

Vendor-Proprietary ICMP Extensions for MPLS

An initial approach considered in the IETF MPLS working group was to define extensions to ICMP in support of MPLS [Bonica 00]. Although this approach was not adopted by the IETF, for the reason that its use could be more general, as discussed in the next section, several major LSR vendors implemented this relatively simple extension to ICMP. First, recall from Chapter 11 that RFC 3032 requires that an LSR decrement the TTL field in the MPLS header, discarding the packet if the TTL reaches zero. RFC 3032 also requires that an LSR return an ICMP time exceeded message when the TTL in the MPLS header reaches zero. The basic idea here was to use this TTL processing along with an agreed-to format for carrying information about the MPLS label stack instead of just the first 64-bits of the contained datagram per RFC 792, or as much as possible as recommended in RFC 1812. The ICMP extension for MPLS requires that 128 bytes of the contained IP

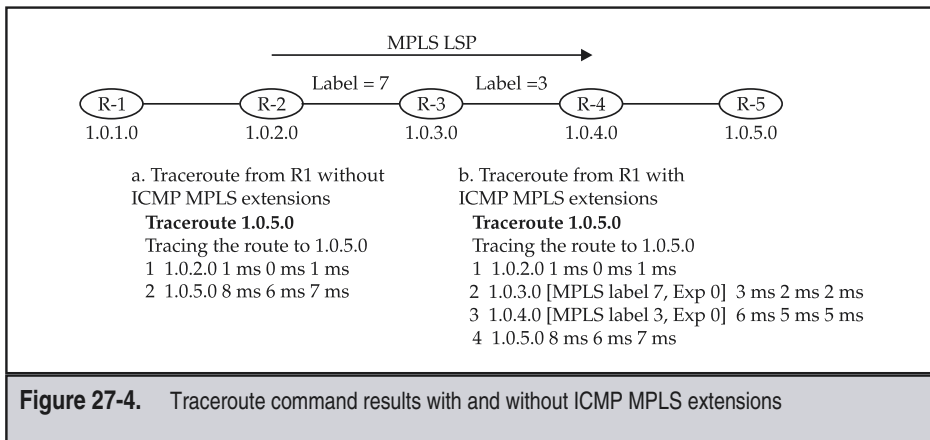


Figure 27-4. Traceroute command results with and without ICMP MPLS extensions

header and datagram be included (padded with zeros if necessary), followed by a new object. This new object contains either the entire MPLS label stack or else more bytes from the contained datagram, as indicated by a revised ICMP common header and a new object header.

With this revised ICMP command implemented in LSRs in a network and ICMP configured to recognize this option, operators are able to trace the label switch path through an MPLS network via an operator command. As in IP networks, route tracing is a proven operational tool for troubleshooting and optimizing networks. Figure 27-4b shows the result of a traceroute with these MPLS extensions to ICMP for the same network using an IP ICMP-based traceroute. Observe how the augmented trace shows the path through the MPLS LSRs over which the LSP passes, indicating the MPLS label at each hop, as well as the value of the MPLS header experimental (EXP) bits.

IETF Direction for IP-Based MPLS Management

The extensions to ICMP for the MPLS equivalent of traceroute was not adopted by the MPLS working group. Instead, it was felt that a more general approach for tracing the path of a number of different tunnels types could be superior. Therefore, at publication time, the effort has been to define general tunnel trace requirements in the IETF common control and management (CCAMP) working group, which is responsible for defining generalized MPLS (GMPLS). An important aspect of the generic tunnel trace requirements [Bonica 2001] is that the management protocol should work for not only MPLS tunnels, but IP-based, L2TP, GRE, and IPsec tunnels as well. The anticipated use of the protocol is to trace the actual tunnel path and support fault diagnosis. The trace may be done in either the control plane or the forwarding plane, through one or more layers of label stacking across potentially heterogeneous technologies.

At the time of writing, there were IETF efforts to define a protocol similar to ICMP echo request and echo reply specifically for the purposes of detecting and isolating faults in MPLS LSPs [Ping 02]. In this approach, the ICMP echo request packets follow the same path as that of the forwarding plane, and they can verify connectivity consistent with that signaled by a control protocol. The basic approach is to test that packets belonging to a specific Forwarding Equivalence Class (FEC) sent over an MPLS LP exit on an LSR that is an egress for that FEC. Such mechanisms could be used to ping an LSP to detect a number of failure modes, and then use the ICMP echo request with increasingly larger values of TTL to diagnose the location of the fault(s) that prevent forwarding. Since IP is assumed present at every node, the ICMP echo reply or TTL exceeded messages need not have an MPLS LSP in the return direction.

A competing approach being developed in ITU-T study group 13 is similar to the ATM-based OAM approach, discussed in the next chapter. Unfortunately, multiple standards approaches for operation and management of MPLS could place more of a burden on vendors and further complicate the already challenging problem of network management, but this may in fact be the result. As we will see in the next chapter, the ITU-T approach inherits much of the operational experience from decades of experience with TDM, while as discussed in this section, the IP-based approach inherits from the opera-

tional experience from decades of experience with packet switching. This in fact may lead to the circumstance in which tools with differing ancestry may be more applicable to different problems, and hence one will win acceptance over the other, depending on the context. It could also lead to service providers using MPLS choosing different management protocols, which would complicate the adoption of interprovider MPLS-based services.

REVIEW

This chapter summarized and compared the competing network management protocols developed by the IETF and the ITU-T. We first covered the IETF-defined Simple Network Management Protocol (SNMP) and Management Interface Base (MIB), which has achieved a high degree of interoperability in the industry. The text then moved on to the ITU-T's Common Management Interface Protocol (CMIP) designed to manage transmission, voice, and data networks. The chapter also discussed the role of proprietary network management protocols as it relates to deployments of SNMP and CMIP. We then summarized the ATM Forum's SNMP-based derived Integrated Local Management Interface (ILMI) for the ATM UNI; the IETF's AToM MIB for management of interfaces; end systems, switches, and networks; and other MIBs defined for ATM. We then summarized the efforts in progress at publication time regarding standardization of MPLS MIBs and their potential usage. This included traffic engineering, LSR management, LDP management, and support for manual MPLS cross-connection of LSPs, as well as support for pseudo-wire and VPN applications. We also summarized vendor proprietary, yet de facto standards for the IP-derived PING and traceroute commands and how they have been applied to MPLS. The chapter concluded with a discussion on the IETF direction for MPLS level management and how this relates to the ITU-T approach described in the next chapter.



CHAPTER 28

ATM and MPLS Management and Performance Measurement

Because the standards and deployment of ATM are more mature than those for MPLS in the Operations and Maintenance (OAM) area, the focus here is on ATM. The chapter begins by introducing the integrated physical and ATM layer OAM information flow architecture, details the ATM OAM cell formats, and provides a description of fault management procedures. Next, the text defines OAM cell formats and procedures used to activate and deactivate the performance measurement and continuity check functions. We then summarize ATM protection switching and how it uses ATM OAM functions.

We then define the general concept of Network Performance (NP) and user Quality of Service (QoS). QoS is user perception, while NP finds use in network management, OAM, and network design. Descriptions and examples then illustrate the functions performed by performance measurement OAM cells. The text then gives detailed examples of how the OAM performance measurement cells and procedures estimate each of the QoS parameters from the traffic contract described in Chapter 20. The chapter concludes with a discussion of the state of applying OAM principles to MPLS networks.

ATM OAM FLOW REFERENCE ARCHITECTURE

Currently, ATM OAM flows are defined only for point-to-point connections. A fundamental part of the infrastructure for network management is that of OAM information. Figure 28-1 shows the reference architecture that illustrates how ATM OAM flows relate to SONET/SDH management flows [ITU I.610]. The F1 flows are for the regenerator section level (called the Section level in SONET), F2 flows are for the digital section level (called the Line level in SONET), and F3 flows are for the transmission path (call the Path level in SONET). ATM adds F4 flows for Virtual Paths (VPs) and F5 flows for Virtual Channels (VCs). Recall from Chapter 11 that a single VP carries multiple VCs. Each flow either traverses an intermediate subnetwork connection or terminates at a termination connection point, as shown in the figure. Each subnetwork connection has a pair of connection points, which could be a single device, or a network of devices.

Each of the F4/F5 flows may be either end-to-end or segment-oriented. An *end-to-end* flow is from one termination connection point to another at the same level. Only devices that terminate ATM connections receive end-to-end OAM flows.

A *segment* flow is a concatenation of VP (or VC) links from one connection point to another connection point. Only network nodes receive segment OAM flows. Indeed, network nodes must remove segment flows before they ever reach devices that terminate an ATM (VP or VC) connection. Segment flows cannot overlap.

ITU-T Recommendation I.610 indicates that OAM flows apply to permanent, semipermanent, and switched virtual ATM connections. The standard does state that procedures for switched connections are for further study. For example, as studied in Chapter 13, the normal response to a trunk or switch failure is to tear down a switched virtual connection (SVC), instead of generating OAM alarms.

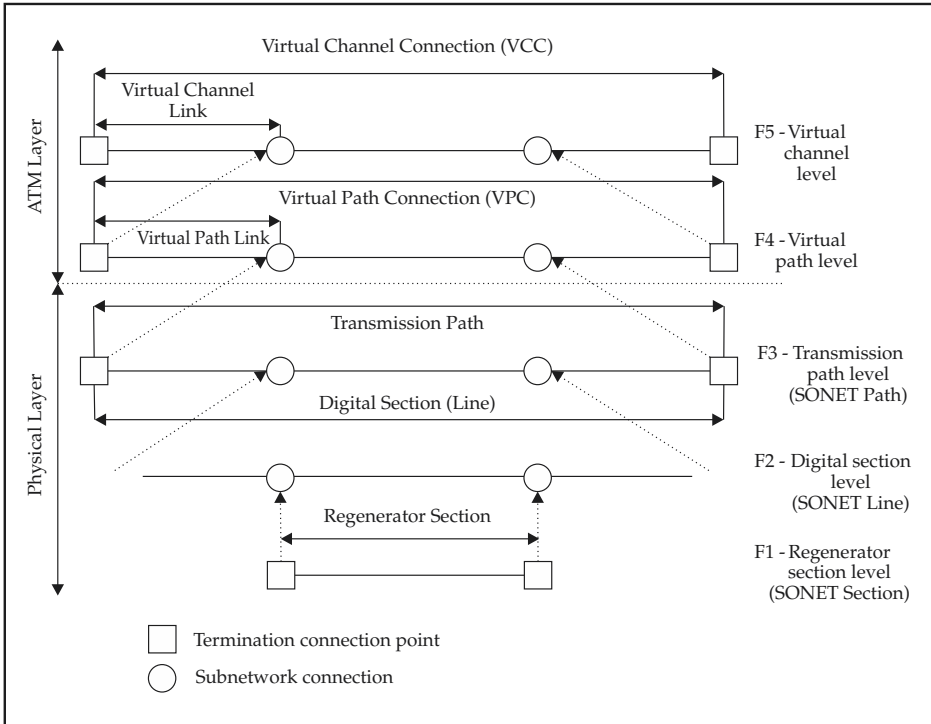
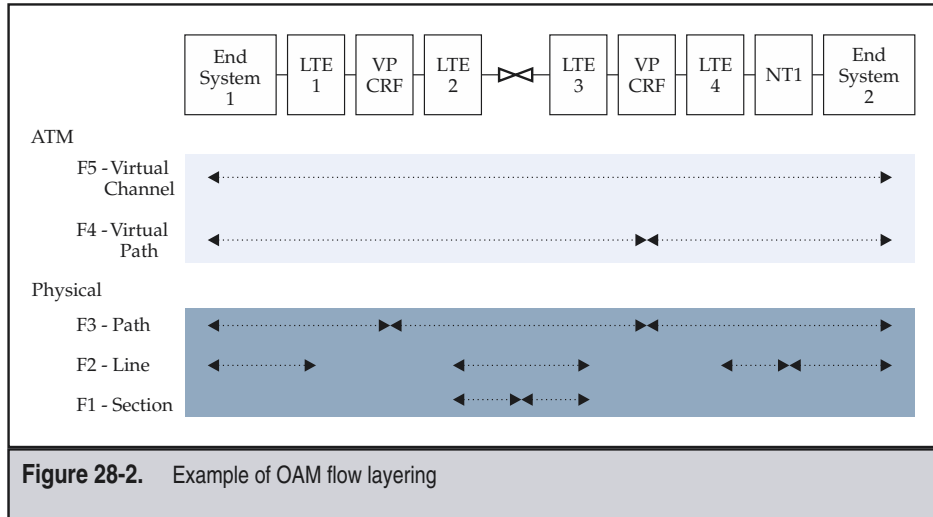


Figure 28-1. ATM and SDH management plane reference architecture

Figure 28-2 shows a real-world example of end-to-end OAM flows for an end-to-end ATM VCC connecting two end systems. Starting from the left-hand side, end system 1 connects to Lightwave Terminal Equipment (LTE) 1, which terminates the digital section OAM flow (F2). The transmission path flow (F3) terminates on the VP Cell Relaying Function (CRF). The VP flow (F4) passes through the VP CRF, since it is only a connection point; that is, only the VPI value changes in cells that pass through that specific VP; the VCI value is unchanged. Next, the example traverses a typical transmission path across the wide area network from LTE 2 to LTE 3 through a repeater (indicated by the “bow tie” symbol in the figure). The regenerator section flow (F1) operates between LTEs 2 and 3 and the repeater, as well as between repeaters. The OAM flow between LTE 2 and LTE 3 is an example of a digital section flow (F2). The transmission path (F3) flow terminates on the VC CRF. The VP flow (F4) also terminates on the VC CRF because in its relaying function it can change the VCI as well as the VPI. A separate digital section OAM flow (F2)

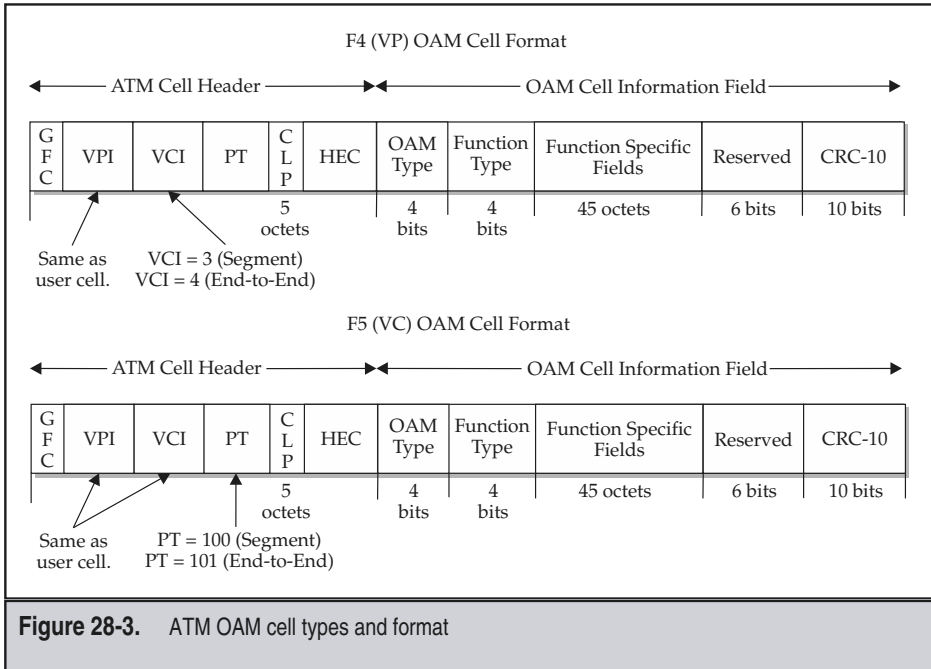


then extends from LTE4 to a CPE device (NT 1) as another line flow (F2). The OAM flow to end system 2 from NT 1 is also a digital section level flow (F2). The transmission path flow (F3) extends from VC CRF to end system 2, as does the VP flow (F4), since the VPI cannot change in this portion of the connection. Finally, note that the VC flow (F5) is preserved from end system 1 to end system 2.

The 1999 revision of I.610 uses some concepts and terminology from the generic transport architecture of G.805 (see Chapter 26) as described in ITU-T Recommendation I.326. The mapping between the models is that an I.610 VPC (VCC) segment endpoint performs G.805 VPC (VCC) segment trail termination and a VPC (VCC) endpoint performs end-to-end VP (VC) trail termination.

ATM OAM CELL FORMATS

The I.610 ATM-layer management standard defines the format of OAM cells for VP flows (F4) and VC flows (F5) on either an end-to-end or a switch-to-switch (i.e., segment) basis. Figure 28-3 depicts the format of these F4 and F5 OAM cells, illustrating the specific coding used to distinguish end-to-end and segment flows within a virtual path or a virtual connection. Note that this use of VCIs within a virtual path and use of Payload Type (PT) within a virtual channel forces VP OAM cells to implicitly follow the same sequence of switches as user cells. Of course, VC OAM cells follow the same path as user cells. This direct relationship of OAM cell and user cell switching traversing the same path is the foundation of many ATM OAM functions.



As described in Chapter 11, VP flows (F4) utilize different VCIs to identify whether the flow is either end-to-end (VCI=4) or segment (VCI=3). For a VC flow (F5), a specific VCI cannot be used because all VCIs are available to users in the VCC service. Therefore, the Payload Type (PT) differentiates between the end-to-end (PT=101) and segment (PT=100) flows in a VCC.

Table 28-1 summarizes the OAM type and function type fields in the OAM cells from Figure 28-3 as defined in I.610. The OAM types are fault management, performance management, automatic protection switching (APS) coordination, activation/deactivation, and system management. Each OAM type has further function types with codepoints as identified in the right-hand side of the table. The activation and deactivation functions support the other OAM types, as indicated in the table. The system management OAM type is defined in the ATM Forum security specification [AF SECURITY] for use in dynamic cryptographic key exchange. When encryption is used, OAM cells are always left unencrypted so that ATM OAM functions can be performed in a secured network. The other function-specific fields of ATM OAM cells are described in subsequent sections.

OAM Type	Function Type		
Fault Management	0001	Alarm Indication Signal (AIS)	0000
	0001	Remote Defect Indication (RDI)	0001
	0001	Continuity Check (CC)	0100
	0001	Loopback (LB)	1000
Performance Management	0010	Forward Performance Monitoring (FPM)	0000
	0010	Backward Reporting (BR)	0001
Activation/Deactivation	1000	FPM and associated BR	0000
	1000	Continuity Check (CC)	0001
	1000	FPM	0010
System Management	1111	Security—non-real time	0001
	1111	Security—real time	0010

Table 28-1. OAM Types and OAM Function Types

ATM OAM FAULT MANAGEMENT

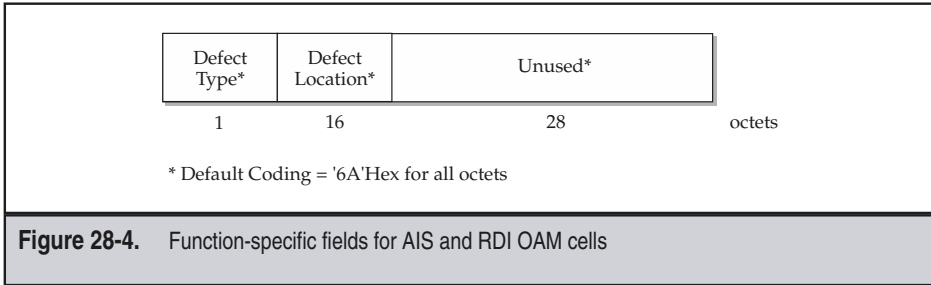
The ATM layer uses an approach based upon the SONET/SDH paradigm. Fault management determines when there is a failure, notifying other elements of the connection regarding the failure, and providing the means to diagnose and isolate the failure.

AIS and RDI Theory and Operation

Figure 28-4 illustrates the ATM OAM cell AIS and RDI function-specific fields. The meaning of each field is described here:

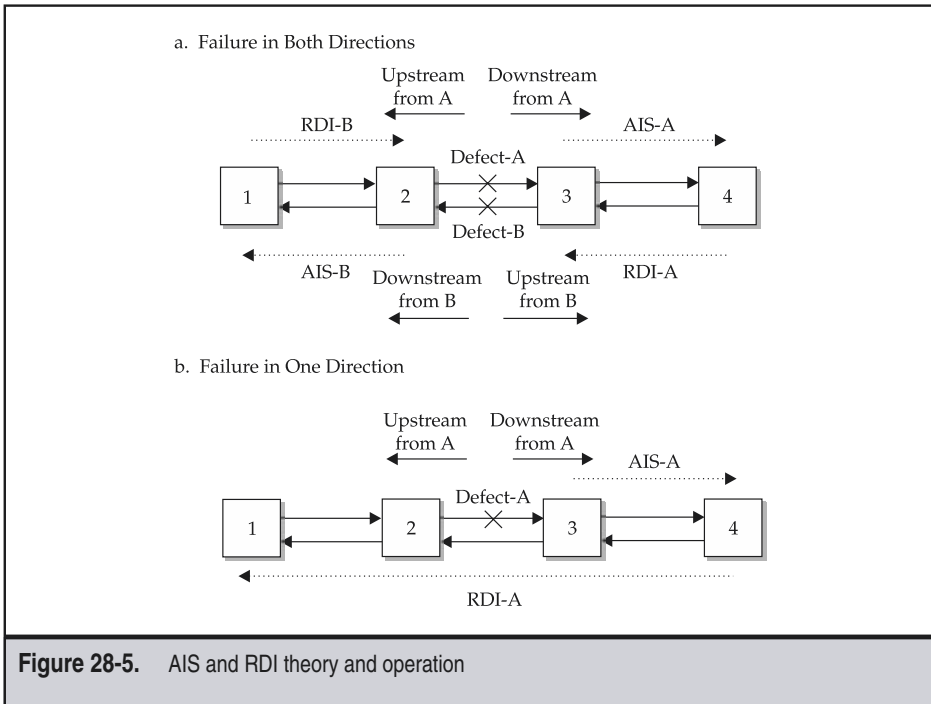
- ▼ **Defect Type** indicates the type of failure as either: unspecified, a VP/VC layer defect, or a lower-layer defect.
- ▲ **Defect Location** indicates where the failure occurred. This is an optional field, which I.610 defines in terms of E.164 or NSAP addresses (see Chapter 13). If present, the RDI cell contains the same information as the corresponding AIS cell.

The following example illustrates the basic principle of AIS and RDI as it applies to any of the OAM functional layers shown in Figure 28-1. We cover two cases: (a) the defect occurs in both directions simultaneously, and (b) the defect occurs in only one direction. Figure 28-5a illustrates the typical case where a defect occurs simultaneously in both directions between nodes 2 and 3, shown as “Defect-A” and “Defect-B” in the figure, such that the resulting AIS and RDI cells can be traced to the defect location. A node adjacent



to the defect generates an AIS signal in the downstream direction to indicate that an upstream defect has occurred, as indicated in the figure. As can be seen from example a, both ends of the connection (nodes 1 and 4) are aware of the defect because of the AIS alarm that they receive. By convention, each generates an RDI signal.

Figure 28-5b illustrates the purpose of the RDI signal. In most networks, the connection should be considered failed even if it fails in only one direction. This is especially true



in data communications where each packet often requires acknowledgment in the reverse direction. Example b illustrates the case of a defect that affects only one direction of a full-duplex connection between nodes 2 and 3. Node 3, which is downstream from the defect, generates an AIS alarm, which propagates to the connection end (node 4), which in turn generates the RDI signal. The RDI signal propagates to the other connection end (node 1), which is now aware that one direction of the connection has failed. Without the RDI signal, node 1 would not be aware that there was a defect in the connection between nodes 2 and 3. This method will also detect any combination of single-direction defects. Note that the node(s) that generate the AIS signals know exactly where the defect is and could report this to a centralized network management system, or the network could make a distributed rerouting response.

A general principle articulated in G.805 is used in ATM fault management. Namely, that a network element should not generate a large number of OAM cells in response to a single root cause event. Specifically, an AIS or RDI cell is sent only if precisely defined conditions for detection of a defect are met. A defect at the VP level (F4) is a physical layer transmission path (F3) defect (e.g., AIS or loss of signal [LOS]), loss of cell delineation (LCD), or loss of continuity (LOC) for the VPC. A defect at the VC level (F5) is either an end-to-end VP level (F4) defect as just defined or LOC for the VCC. I.610 defines rules for the generation of end-to-end and segment levels (an option within an operator domain) for VP and VC AIS and RDI OAM cells. From the VP and VC definitions of defects, it is important to note that any defect that causes end-to-end VP-AIS to be generated would cause VC-AIS to be generated only at the VP endpoints, and not at an intermediate VP connection point. This layering and correlation of fault management cells helps to reduce OAM traffic volume and help diagnose and localize the root cause of a defect.

Once a fault condition is detected, the node detecting such a defect sends an OAM cell periodically until the defect condition clears. In order to limit the number of OAM fault management cells, the period for generating OAM cells is on the order of a second. Furthermore, the ATM Forum UNI does not require VC-AIS and VC-RDI. These restrictions constrain the amount of OAM cells that are generated (and processed) such that the function of AIS and RDI is delivered in an efficient manner.

Loopback Operation and Diagnostic Usage

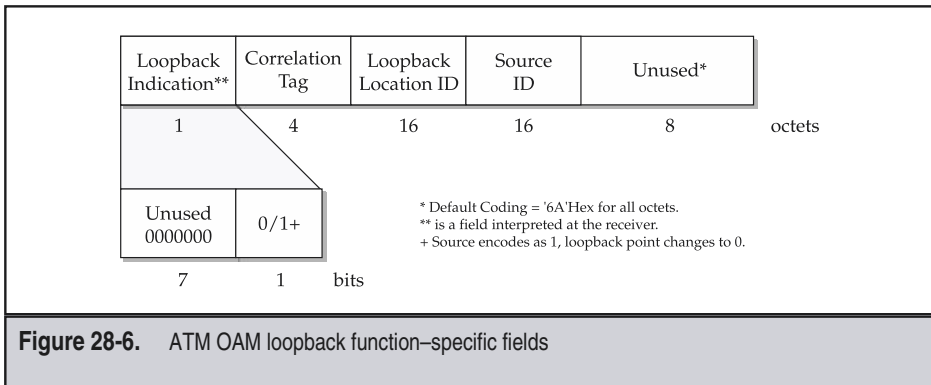
Figure 28-6 illustrates the ATM OAM cell loopback function-specific fields. A summary of the ATM OAM cell loopback function-specific fields is as follows:

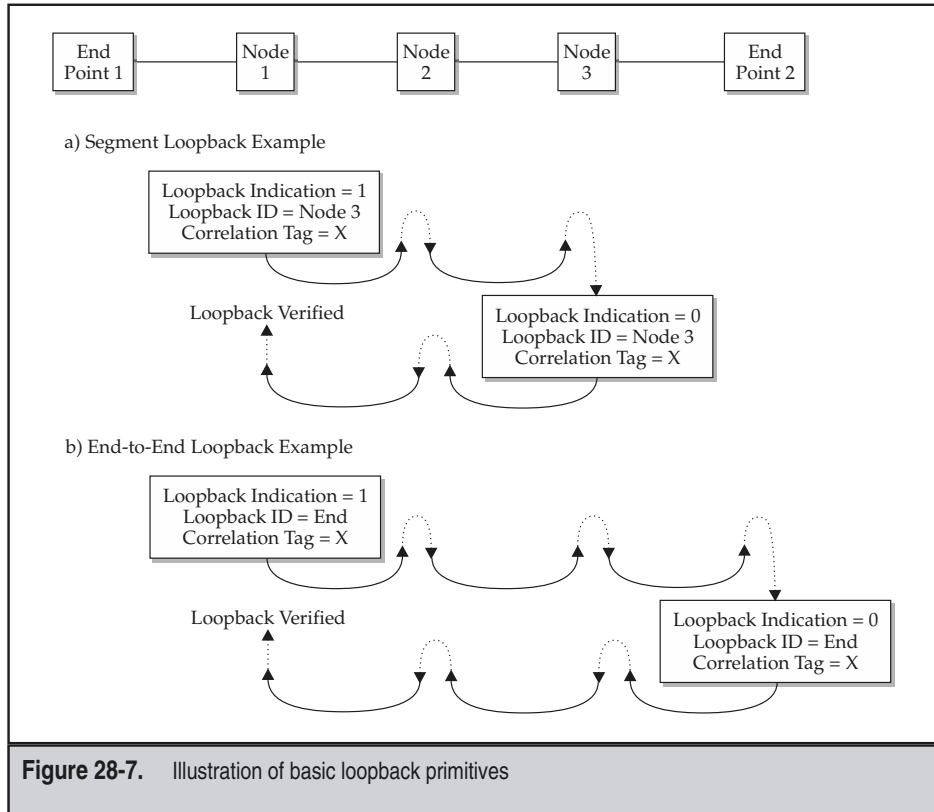
- ▼ **Loopback Indication** is a field that contains '0000 0001' when originated. The loopback point changes it to a value of '0000 0000'; this action confirms that the ATM layer performed the loopback and prevents the cell from looping indefinitely.
- **Correlation Tag** is a field defined for use by the originator because multiple OAM loopback cells may be in transit on a particular VPC or VCC. This field allows the sender to correlate its loopback command with a response.

- **Loopback Location ID** is an optional field provided to the sender and the receiver for use in segment loopbacks to identify the connecting point where the loopback should occur. The value of all 1s indicates that the loopback should occur at the end point.
- ▲ **Source ID** is an optional field provided to identify the loopback source in the OAM cell using E.164 or NSAP addressing (see Chapter 13).

As seen in the preceding section, AIS and RDI are quite useful in communicating to connection endpoints that a defect has occurred. However, some defects are not so easily detected, or an operator may just want to verify continuity without taking the connection out of service. I.610 sets the maximum loopback response time at five seconds. An example of a defect that does not generate AIS/RDI is a misconfiguration of VPI and/or VCI translations at a connection point, such that cells do not reach the destination endpoint or are sent to the wrong destination. Loopback OAM cells help diagnose and localize these types of problems.

As described earlier, the ATM header determines whether an OAM cell is either segment or end-to-end. Figure 28-7a illustrates the operation of a segment loopback normally used within a network. The originator of the segment OAM loopback cell at node 1 sets the Loopback Indication field to 1, provides a loopback ID indicating the last connecting point of the segment (node 3), and adds a correlation tag of "X" so that it can match this field in a response. The loopback destination (node 3) extracts the loopback cell, changes the Loopback Indication field to 0, and transmits the loopback cell in the opposite direction. Note that every node along the path may extract and processes every OAM cell, which results in node 2 transparently conveying the segment loopback cell received from node 1 onto node 3. Node 3 matches the Loopback ID, sets the Loopback Indication field to 1, and sends an OAM cell in the opposite direction of the virtual connection. Eventually, node 1



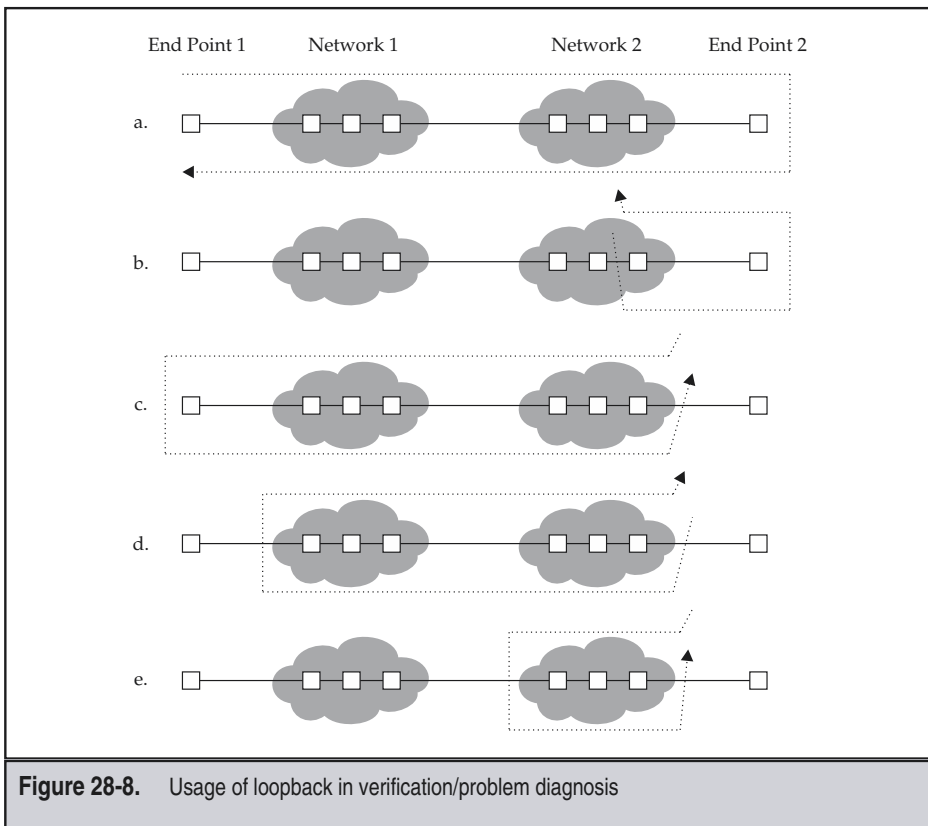


extracts the loopback cell, confirms that the loopback indication changed to 0, and matches the correlation tag. This procedure verifies that ATM continuity exists between the segment of VP (or VC) links extending from node 1 to node 3. If a physical loopback were in place between nodes 2 and 3, then node 1 would have been able to determine this because the loopback indication in the OAM loopback cell would still have a value of 1.

Figure 28-7b illustrates an end-to-end loopback that could be used by a node to verify connectivity with an endpoint, or by an endpoint to verify connectivity with the distant endpoint. In the example, node 1 performs an end-to-end loopback to endpoint 2. Node 1 inserts an end-to-end OAM loopback cell that has a Loopback Indication of 1, a Loopback ID indicating the endpoint (i.e., all ones), and a correlation tag of "X" that it uses to match against the response. Intermediate nodes may extract and process every OAM cell, but since the Loopback ID indicates the endpoint, they pass it on until the destination

endpoint 2 extracts the loopback cell. The destination endpoint changes the Loopback Indication field to 0 and transmits the loopback cell in the opposite direction. Eventually, node 1 extracts the loopback cell and matches the correlation tag, thus verifying the continuity of VP (or VC) links from node 1 to endpoint 2.

Figure 28-8 illustrates how the segment and end-to-end loopback cells can be used to diagnose a defect that AIS and RDI cannot. An example of such a defect would be a misconfigured VPC or VCC. The example shows two endpoints and two intervening networks, each with three nodes. Part a shows the verification of end-to-end continuity via an end-to-end loopback to endpoint 1. If this were to fail, then network 2 could diagnose the problem as follows. Part b shows verification of connectivity between a node in network 2 and endpoint 2 via an end-to-end loopback. If this fails, then the problem is between network 2 and endpoint 2. Part c shows verification of connectivity from network 2



to endpoint 1 via an end-to-end loopback. If this fails, there is a problem in the link between endpoint 1 and network 1, a problem in network 1, or a problem in the link between networks 1 and 2. Part d shows verification of connectivity across networks 1 and 2 via a segment loopback. If this succeeds, then the problem is the access line from endpoint 1 to network 1. If this fails, then part e shows verification of connectivity from entry to exit in network 2. If this succeeds, then the problem is in network 1. Localization of the problem to a specific node within network 1 can then be done using the segment loopback. See I.610 for more details on the use of the loopback function for diagnosis and localization of other types of faults.

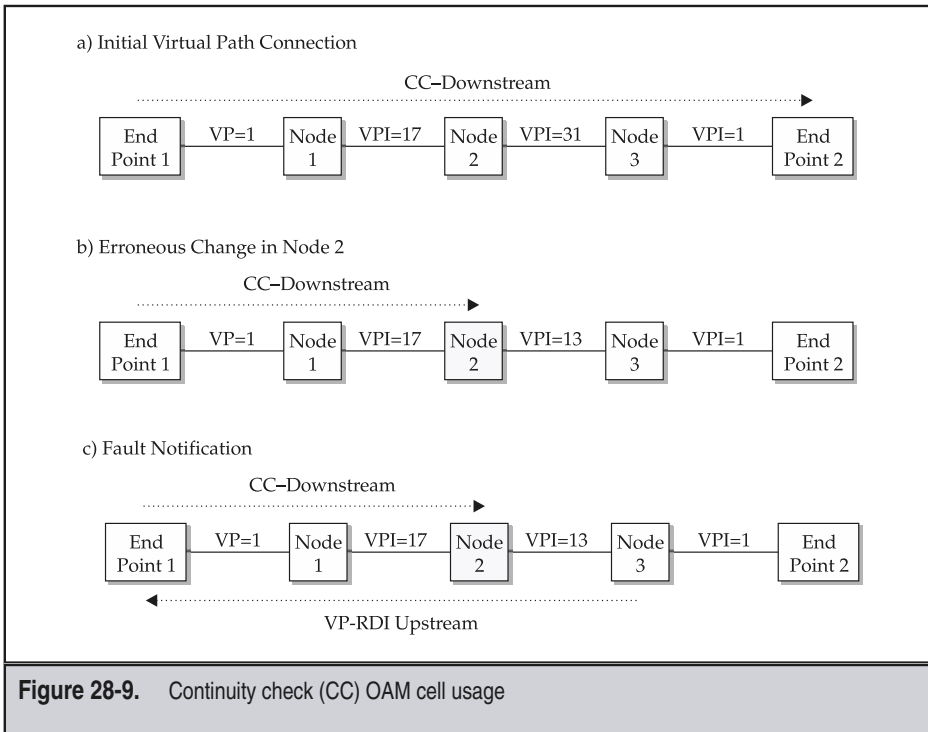
Continuity Check (CC)

The idea behind the continuity check is straightforward. An endpoint sends a cell periodically (nominally once per second) if no other traffic is sent on the connection so that the connecting points and the other endpoint can distinguish between a connection that is idle and one that has a defect that does not generate AIS/RDI. Continuity checking can be activated and deactivated by procedures described later or through use of the optional OAM traffic descriptor in Q.2931 signaling messages for an SVC or SPVC that indicates that the origin will transmit CC cells once per second. This allows the destination to expect receipt of CC cells to confirm that the user plane of the connection is functioning correctly in response to a dynamically established connection. The CC cell currently has no standardized function-specific fields. The continuity check function may be invoked for a subset of VPC or VCC connection points on an end-to-end or segment basis. The ATM Forum does not require support for continuity checking at the User Network Interface (UNI).

The continuity check detects defects that AIS cannot, such as an erroneous VP cross-connect change as illustrated in Figure 28-9. Part a shows a VP connection traversing three VP cross-connect nodes with VPI mappings (configured to be symmetric) shown in the figure carrying only continuity check (CC) cell traffic downstream, interleaved with the VP user cells. In part b, an erroneous cross-connect is made at node 2, shown shaded in the figure, interrupting the flow of CC cells. In part c, node 3 detects this continuity defect and generates a VP-RDI OAM cell in the opposite (i.e., upstream) direction of the VPC.

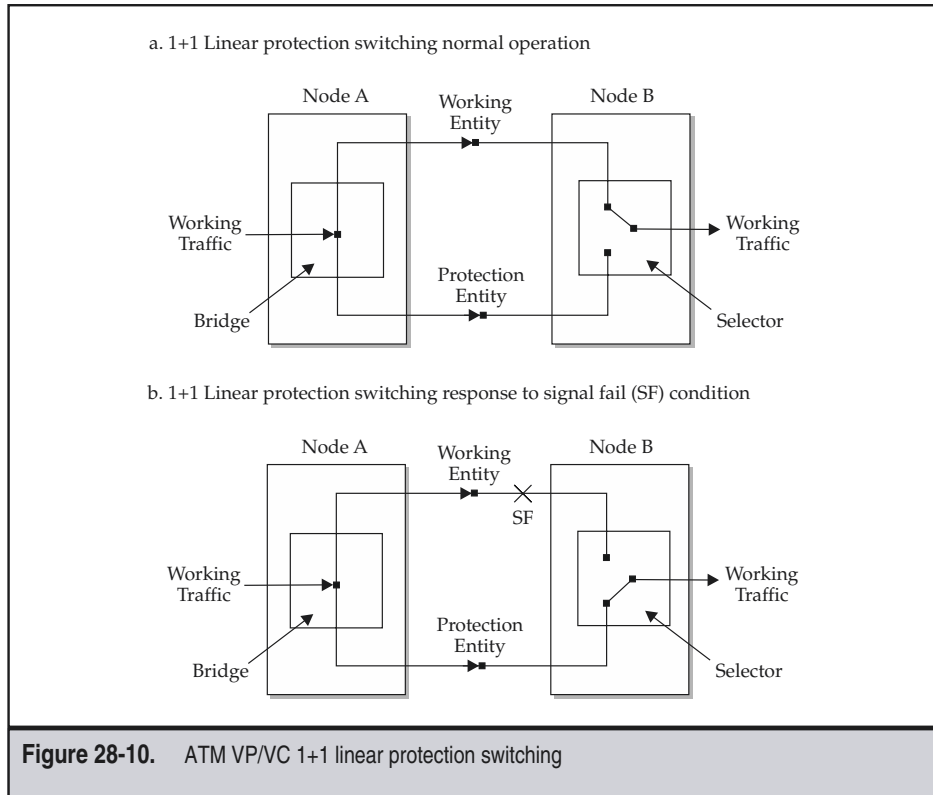
ATM PROTECTION SWITCHING

ITU-T Recommendation I.630 applies the concept of protection switching from TDM transmission networks to protection for only one or a group of ATM VP or VC links or connections. An example use is where there is no SONET/SDH ring protection for the transmission circuits connecting ATM switches. It also protects against ATM switch or port failures along the VP or VC path, failure modes that transmission protection switching does not address. When used with transmission protection schemes, the ATM protection must wait until the lower-layer networks have first had a chance to restore to avoid contention and race conditions. The protection switching modes defined include support



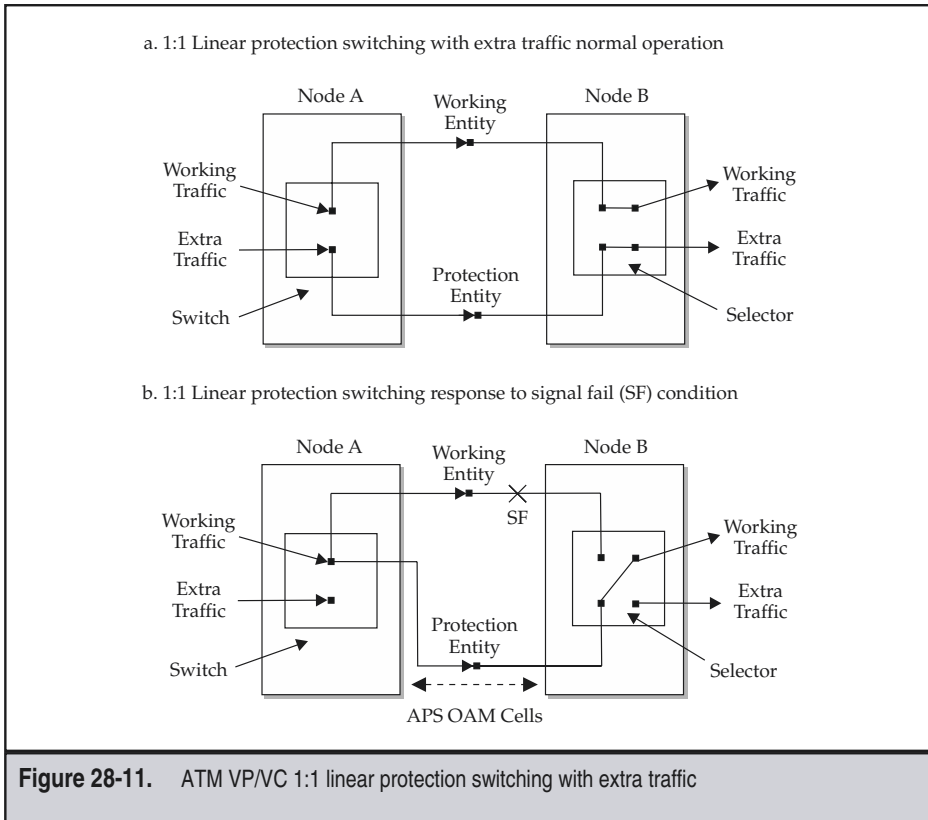
for linear, ring, and mesh network topologies. We summarize two popular linear protection-switching modes as an example.

Figure 28-10 illustrates the operation of 1+1 linear protection switching for a unidirectional connection across the protection domain from node A to node B. Protection switching concepts are applicable to bidirectional connections using the ATM OAM APS cell type. Figure 28-10a shows the typical configuration where transmitter node A bridges traffic onto both the working and protection entities (i.e., a single VP [VC] or a group of them). A selector function that operates on VPI/VCI values and physical interfaces at receiver node B makes the decision about whether to send the cell stream from the working or protect entity to the working traffic output according to whether a signal fail condition is detected. A signal fail condition occurs when end-to-end or segment AIS, as applicable to the protection domain, is present for longer than a provisionable hold-off time, which is specified to be in the range of 0 to 10 seconds with a granularity of 500 ms. Figure 28-10b illustrates the case where the receiver node B has detected signal failure and uses the



selector function to now send the protection entity cell stream to the working traffic output in accordance with incoming VPI/VCI value(s) and incoming port.

Figure 28-11 illustrates the operation of 1:1 linear protection switching for a unidirectional connection across the protection domain from node A to node B. The 1:1 protection technique can support preemptable extra traffic instead of dedicating protection capacity. Therefore, it can reduce the amount of transmission capacity required to make an ATM network resilient. However, the 1:1 technique is inherently slower than the 1+1 approach because communication must occur between both ends prior to a switching operation. Figure 28-11a shows transmitter node A sending working traffic on the working entity and sending extra traffic on the protection entity. Figure 28-11b shows what happens when receiver node B detects a signal failure condition (e.g., AIS for greater than the hold-off time). It configures its selector for VPI/VCI value(s) on the protection entity for switching to the working traffic output, disconnects the extra traffic output, and sends



AIS on the extra traffic output. In order to effect a protection switch, it must signal transmitter node A using ATM OAM APS cells that a signal fail condition exists and that a protection switch is required. In response to this request, transmitter node A then disconnects the extra traffic input from the protection entity and connects the working traffic input to the protection entity. Note that there may a transient period where the transmitter and receiver node switch and selector functions are not synchronized, but there is no possibility for sending working traffic over the extra traffic connection.

The standard also defines a mode that allows an operator to manually command a protection switch, for example, in preparation for a maintenance activity. Once the failure has been repaired, there is an option for the nodes to automatically revert back to the normal configuration, or else remain in the current state until manually commanded to change.

ATM PERFORMANCE SPECIFICATION AND MEASUREMENT

QoS is specified and measured either for each individual VPC or VCC, or measured over the aggregate of many VCCs or VPCs. For the individual case, a device measures QoS by inserting OAM cells on each connection, increasing cell traffic and introducing additional complexity (and cost) for processing these OAM cells. On the other hand, measurement on the aggregate assumes that the performance of all connections for a network are identical, and that only a sample is necessary, which significantly reduces the number of measurement cells that must be transmitted and processed. The cost and complexity of individual measurement is justified when it is critical to ensure that the performance of an individual virtual connection is being achieved, for example, if a service level agreement (SLA) is in force. Normally, measurement on the aggregate is adequate to ensure that the QoS of a group of virtual connections is being met, and hence the QoS of the individual virtual connection is met on a statistical basis. Let's now look at some basic definitions and OAM cells and procedures involved in performance measurement.

Network Performance and Quality of Service

Although ITU-T Recommendation I.350 was originally written to cover only ISDN, some of the concepts are quite relevant to ATM and MPLS networks. Quality of Service (QoS) is observed by the user on an end-to-end basis, in particular, as determined by parameters that can be directly observed and measured at the point at which the service is accessed by the user. On the other hand, Network Performance (NP) is measured in terms of parameters that are meaningful to the network provider for use in system design, configuration, operation, and maintenance.

ITU-T Recommendation I.350 makes a distinction between QoS and NP along the following lines: Although user-oriented QoS parameters are a valuable framework for network design, they may not be sufficient to state performance requirements for individual connections. And even though NP parameters ultimately determine user-observed QoS, they may not describe quality in a way meaningful to users. There is a need to quantitatively relate individual and accumulated NP parameters in order for a network to meet specific QoS objectives.

ATM Performance Measurement (PM)

This section describes how to activate and deactivate PM and CC procedures. We then describe the performance measurement process. A key objective is to measure or estimate these parameters *in-service*; that is, the customer traffic is not impacted by the measurement process.

Activation/ Deactivation Procedures

Figure 28-12 depicts the ATM OAM cell Activation/Deactivation function-specific fields. The following text defines the meaning of each field.

- ▼ **Message ID** defines the type of command or response as follows:
 - '000001' Activate Command
 - '000010' Activation Confirmed Response
 - '000011' Activation Request Denied Response
 - '000101' Deactivate Command
 - '000110' Deactivation Confirmed Response
 - '000111' Deactivation Request Denied Response
- **Direction of Activation** coding indicates the A-B direction ('10'), the B-A direction ('01'), or a two-way action ('11').
- **Correlation Tag** enables nodes to correlate commands and responses.
- **PM Block Size A-B** PM Block Size A-B identifies the Performance Measurement (PM) block size supported from A to B for sizes ranging from 128 to 32,768 in increments that are successive powers of two. The default value of all zeros is used for continuity checking.
- ▲ **PM Block Size B-A** identifies the block sizes supported from B to A using the same convention as in the preceding field. Note that the block sizes may differ in each direction.

Figure 28-13 illustrates the activation/deactivation procedure for performance monitoring or continuity checking. In the example, connection/segment endpoint A generates a De(Activate) command toward B requesting action on the A-to-B, B-to-A, or both directions.

Message ID	Directions of Action	Correlation Tag	PM Block Sizes A-B**	PM Block Sizes B-A**	Reserved for future use*	
6	2	8	4	4	42x8	bits

* Default coding = '6A'Hex ** Default coding = '0000'

Figure 28-12. ATM Activation/Deactivation OAM cell function-specific fields

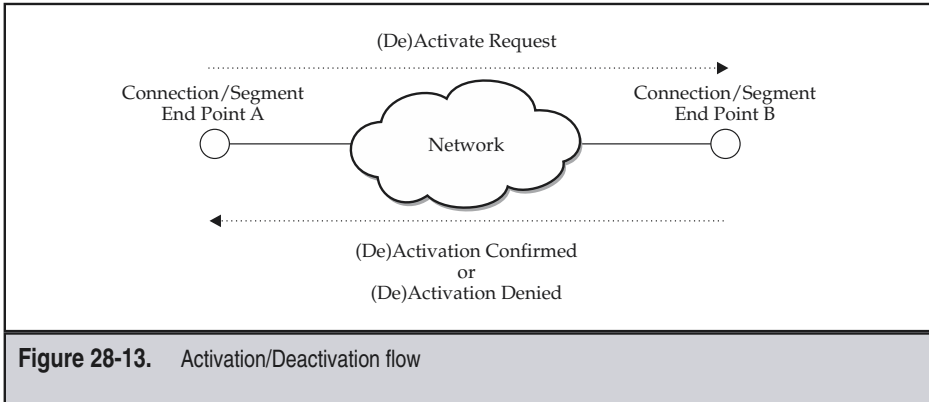


Figure 28-13. Activation/Deactivation flow

If B can comply with all of the requests, then B responds with a (De)Activation Confirmed message. If B cannot comply with the command(s), then it returns a (De)Activation Request Denied response, for example, if the endpoint cannot support performance management.

Once a performance measurement flow is activated, the procedure described in the following section is performed. Activation and Deactivation allow the performance measurement to be performed on selected VPCs and VCCs. This keeps the total processing load required for performance measurements manageable.

Performance Measurement Procedure

Figure 28-14 depicts the Forward Performance Monitoring (FPM) ATM OAM function-specific fields. The following text defines the meaning of each field:

- ▼ Monitoring Cell Sequence Number (MCSN) is the PM cell number, modulo 256.
- Total User Cell (TUC) is the total number of CLP=0+1 cells (TUC-0+1) or CLP=0 cells (TUC-0) containing user data sent since the last PM cell.
- BEDC-0+1 is a block error detection code computed over all of the CLP=0+1 user cells since the last PM cell. PM procedures employ it for error rate estimation.
- ▲ Time Stamp (TSTP) is an optional field indicating the time at which the cell was inserted.

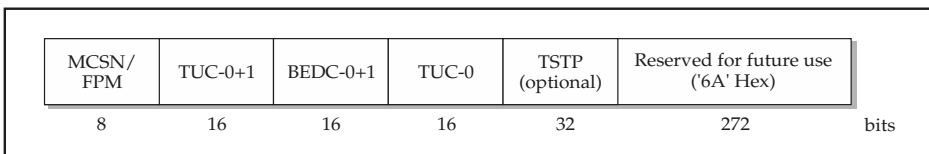


Figure 28-14. Forward performance monitoring (FPM) OAM cell function-specific fields

Figure 28-15 depicts the Backward reporting (BR) ATM OAM function-specific fields. The following text defines the meaning of each field not already defined for FPM.

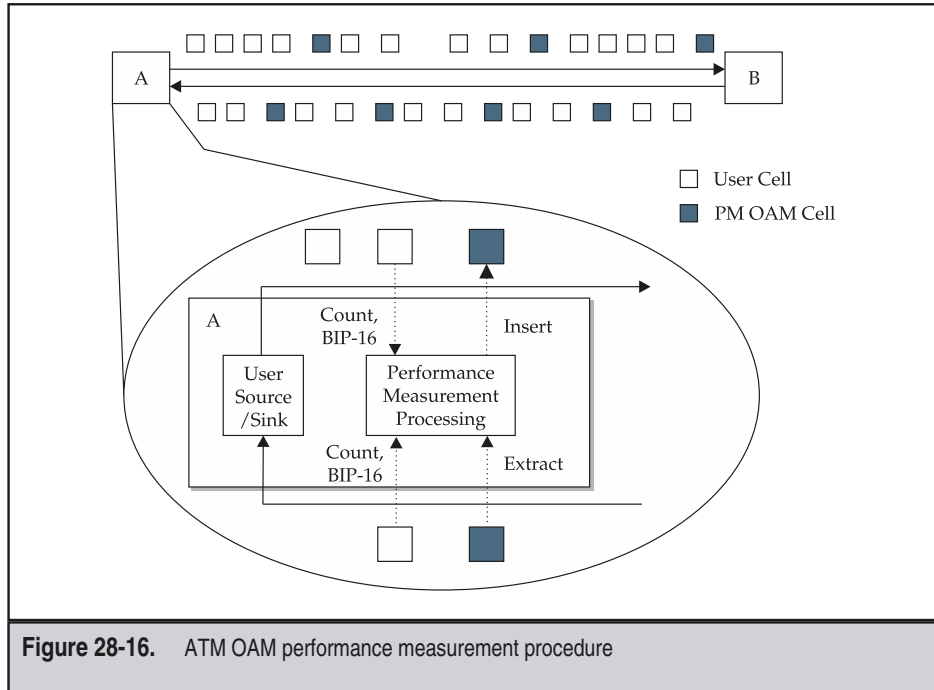
- ▼ **Reported Monitoring Cell Sequence Number (RMCSN)** is the value copied from the MCSN/FPM value from a paired FPM cell.
- **Severely Errored Cell Block Count (SEBC)** is a running counter, modulo 256, of the number of severely errored cell blocks detected.
- **Block Error Result (BLER-0+1)** is a count of the number of errored parity bits detected across all user CLP=0+1 cells in the last block.
- ▲ **Total Received Cell Count (TRCC)** is the number of received CLP=0+1 cells (TRCC-0+1) or received CLP=0 cells (TRCC-0).

Figure 28-16 illustrates the operation of the FPM and BR OAM cell insertion and processing. The connection or segment endpoints A and B that are involved are determined by the activation/deactivation procedure. In this example, the FPM cell flow in each direction has different block sizes, every four cells from left to right, and every two cells from right to left. The functions involved with FPM on the transmit side are insertion of OAM cells, counting user cells, and computing the 16-bit parity. At the destination, the receiver extracts FPM OAM cells and may return BR cells. The receiver makes the same counts and recomputes the 16-bit parity for comparison with the value received in the monitoring cell computed by the transmitter. Note that the monitoring cell contains the results for the cells in the preceding block.

Performance monitoring OAM cells detect the following types of impairments on ATM connections: missing or lost cells, many bit error patterns, extra or misinserted cells, delay, and delay variation. Higher-level network management systems can then utilize this data to determine if the desired ATM-level QoS is being delivered. Calculations based upon the measurements just described readily estimate ATM QoS and NP parameters, as described in the next section. Another means to estimate QoS and NP is to connect ATM test equipment over connections in the same service category that traverse the same switches and trunks that user cells traverse.

MCSN/ BR	TUC-0+1	Reserved (‘6A’ Hex)	TUC-0	TSTP (optional)	Reserved (‘6A’ Hex)	RMCSN	SECBC	TRCC-0	BLER -0+1	TRCC -0+1
8	16	16	16	32	216	8	8	16	8	16
										bits

Figure 28-15. Backward reporting (BR) OAM cell function-specific fields



NP/QoS Parameter Estimation

This section defines the QoS and NP parameters in terms of basic cell transfer outcomes. The text also describes a method to estimate these QoS parameters from OAM cells.

ATM Cell Transfer Outcomes

ITU-T Recommendation I.356 defines QoS in terms of specific outcomes related to cell entry events at one end of a connection and exit events at the other end, as illustrated in Figure 28-17, as follows:

- ▼ A *cell exit event* occurs when the first bit of an ATM cell completes transmission out of a device to an ATM network element across source Measurement Point 1.
- ▲ A *cell entry event* occurs when the last bit of an ATM cell completes transmission into a device from an ATM network element across destination Measurement Point 2.

As illustrated in Figure 28-17, a cell that arrives at the exit point without errors and is not too late is considered successfully transferred. A successfully transferred cell may

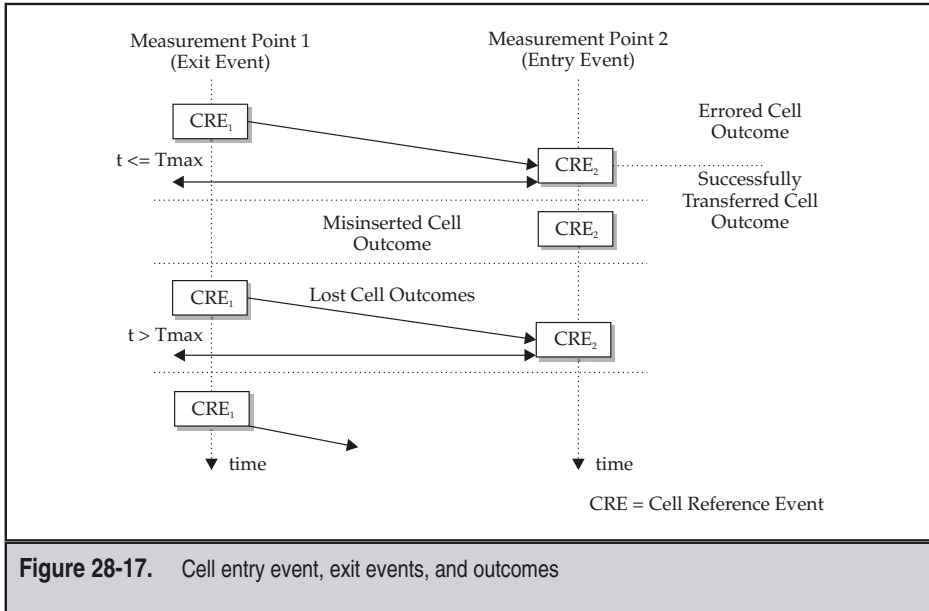


Figure 28-17. Cell entry event, exit events, and outcomes

arrive with the CLP bit set to zero, or one. A successfully transferred cell with the CLP bit set to one is called a *tagged cell outcome* if it entered the network with the CLP bit set to zero. If a cell arrives within a certain delay (T_{max}) but has errors, then it is considered an *errored outcome*. If a cell arrives too late, or never arrives at all, then it is considered lost. There is also a possibility of a misinserted cell, which is defined as the case when a cell arrives at the exit point for which there was no corresponding input cell at the entry point, a condition that can occur due to undetected cell header errors.

The following possible cell transfer outcomes between measurement points for transmitted cells are defined in ITU-T Recommendation I.356.

- ▼ **Successful Cell Transfer outcome** The cell corresponding to the transmitted cell is received within a specified time T_{max} . The content of the received cell conforms exactly to the corresponding cell payload and has a valid header field.
- **Tagged Cell Transfer outcome** A successful cell transfer outcome: the CLP was bit 0 at measurement point 1 but had a value of 1 at measurement point 2.
- **Errored Cell outcome** A cell is received corresponding to the transmitted cell within a specified time T_{max} . Either the content of the received cell payload differs from that of the corresponding transmitted cell, or the cell has an invalid header field.

- **Lost Cell outcome** No cell is received corresponding to the transmitted cell within a specified time T_{max} (examples include “never arrived” or “arrived too late”).
- **Misinserted Cell outcome** A cell is received for which there is no corresponding transmitted cell.
- ▲ **Severely Errored Cell Block outcome** This is when M or more Lost Cell outcomes, Misinserted Cell outcomes, or Errored Cell outcomes are observed in a received cell block of N cells transmitted consecutively on a given connection.

ATM Performance Parameters

This section summarizes ATM cell transfer performance parameters defined in I.356 as described in Chapter 20 using the cell transfer outcomes defined previously. It is important to note that QoS (or NP) can be measured only when a connection is in the available state, as defined in I.357. These definitions apply only to cells conforming to the traffic contract. Nonconforming cells must be excluded from the cell transfer outcomes.

Cell Error Ratio

The *cell error ratio* is defined as follows for one or more connection(s):

$$\text{Cell Error Ratio} = \frac{\text{Errored Cells}}{\text{Successfully Transferred Cells} + \text{Errored Cells}}$$

Successfully Transferred Cells and Errored Cells contained in cell blocks counted as Severely Errored Cell Blocks should be excluded from the population used in calculating the cell error ratio. Errored Cells can only be estimated by counting the number of up to M ($2 \leq M \leq 16$, with a default of 4) parity errors in the BEDC-0+1 code for the block. The successfully transferred cell count is the Total User Cell number (TUC-0+1) from the PMOAM cell.

Severely Errored Cell Block Ratio

The *severely errored cell block ratio* for one or more connection(s) is defined as:

$$\text{Severely Errored Cell Block Ratio} = \frac{\text{Severely Errored Cell Blocks}}{\text{Total Transmitted Cell Blocks}}$$

A cell block is a sequence of N cells transmitted consecutively on a given connection. A severely errored cell block outcome occurs when more than a specified number of errored cells, lost cells, or misinserted cells are observed in a received cell block. An Errored Cell Block (ECB) contains one or more BIP-16 errors, lost cells, or misinserted cells. A Severely Errored Cell Block (SECB) is a cell block with more than M ($2 \leq M \leq 16$, with a default of 4) BIP-16 errors, or more than K ($2 \leq K \leq M$, with a default of 2) lost or misinserted cells.

Cell Loss Ratio

The *cell loss ratio* is defined for one or more connection(s) as:

$$\text{Cell Loss Ratio} = \frac{\text{Lost Cells}}{\text{Total Transmitted Cells}}$$

Lost and transmitted cells counted in severely errored cell blocks should be excluded from the cell population in computing cell loss ratio. The number of lost cells can be estimated as the difference in the past two Total User Counts (TUC) received in the FPM OAM cells from the distant end minus the number of cells actually received in a cell block. If this result is negative, then the estimate is that no cells were lost, and cells were misinserted as defined next. Note that this estimation method would report zero loss and misinsertion if there are an equal number of cell loss and misinsertion outcomes in a cell block.

Cell Misinsertion Rate

The *cell misinsertion rate* for one or more connection(s) is defined as:

$$\text{Cell Misinsertion Rate} = \frac{\text{Misinserted Cells}}{\text{Time Interval}}$$

Severely Errored Cell Blocks should be excluded from the population when calculating the cell misinsertion rate. Cell misinsertion on a particular connection is most often caused by an undetected error in the header of a cell being transmitted on a different connection. This performance parameter is defined as a rate (rather than the ratio) because the mechanism producing misinserted cells is independent of the number of transmitted cells received on the corresponding connection. The number of misinserted cells can be estimated as the number of cells actually received in a cell block minus the difference in the past two Total User Counts (TUC) received in the PM OAM cells from the distant end. If this result is negative, then cell loss has occurred, and the number of misinserted cells is estimated as zero.

Measuring Cell Transfer Delay and Cell Delay Variation

The *cell transfer delay (CTD)* is defined as the elapsed time between a cell exit event at measurement point 1 (e.g., at the source UNI) and a corresponding cell entry event at measurement point 2 (e.g., the destination UNI) for a particular connection. The cell transfer delay between two measurement points is the sum of many things that contribute to delay (see Chapter 20) between these measurement points. The methods proposed to measure delay are optional and utilize either the time stamp function-specific field or a well-defined test cell stream.

For the time stamp method, synchronized time-of-day clocks (e.g., derived from the Global Positioning System [GPS]) at the sender and receiver are essential to the measurement of absolute delay, cell delay variation (CDV) can be estimated by taking differences

in time stamps. Any estimation of delay also assumes that the FPM OAM cells are processed exclusively in hardware, and not in software as other OAM cell types could be. Figure 28-18 illustrates how absolute delay and differential delay can be measured using the time stamp method. The source periodically sends OAM FPM cells with a time stamp. These cells traverse an ATM network and experience variable delays. At the destination, the time stamp is extracted from the OAM FPM cell and several operations are performed on it. First, the absolute delay is calculated as the (nonnegative) difference between the local time stamp clock and the time stamp received in the OAM FPM cell. Next, the value in a memory is subtracted from the absolute delay to yield a differential delay. Finally, the current absolute delay calculation is stored in the memory for use in calculation of the next differential delay.

The *mean cell transfer delay* is the average of a specified number of absolute cell transfer delay estimates for one or more connections. The 2-point cell delay variation (CDV) defined in I.356 can be estimated from the differential delays. A histogram of the differential delay can be computed, as well the mean, the variance, or other statistics.

ITU-T Recommendation I.356 also defines the 1-point CDV as the variability in the pattern of cell arrival events observed at a single measurement point with reference to the negotiated peak rate $1/T$ as defined in the traffic contract. A method to implement this measurement is for a constant rate source to emit a cell once every T seconds (note this implies that T is a multiple of the cell slot time in the TDM transmission convergence sublayer). This cell stream, perfectly spaced, is transmitted across an ATM network that introduces variations in delay that we wish to measure. The receiver knows the spacing interval T and can compute the interarrival times of successive cells and subtract the time T to result in a 1-point CDV estimate. Positive values of the 1-point CDV estimate correspond to cell clumping, while negative values of the 1-point CDV estimate correspond to gaps, or dispersion, in the cell stream. At the receiver, cell clumping occurs for cells that

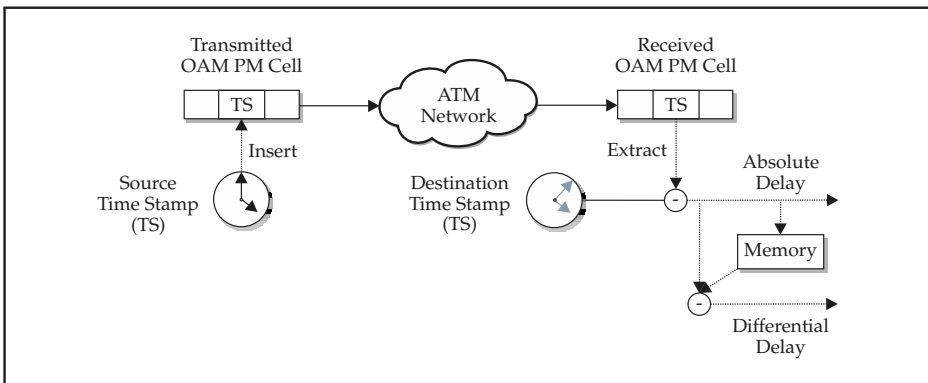


Figure 28-18. Time stamp method delay estimation

are closer together than T seconds, while dispersion occurs for cells spaced further than T seconds apart. This is important for determining the likelihood of overrun and underrun for CBR services as described in Chapter 16.

MPLS OAM STATUS AND DIRECTION

This section provides a brief summary and references for the emerging area of OAM concepts as they could apply to MPLS. Since MPLS had its origins in IP, the state of MPLS network management at publication time was primarily IP-based, as summarized in the preceding chapter. However, there may be some applicability of concepts from TDM and ATM network management to certain modes of MPLS.

As studied in Chapter 14, only the mode of MPLS label distribution with fully enumerated explicit routing and ordered control is strictly speaking a connection-oriented protocol like ATM. There is also the case where an MPLS LSP is set up on a hop-by-hop basis, for example, using the MIBs described in Chapter 27. For these cases, the ITU-T is developing requirements and an OAM protocol for MPLS based upon the ATM OAM approach.

Overview of ITU Direction for MPLS OAM

The requirements for MPLS OAM are stated in ITU-T Recommendation Y.1710. As you would expect, MPLS OAM expresses everything in terms of packets instead of cells. The concepts in Y.1710 are based on the G.805 “transport” model summarized in Chapter 26 as applied to the MPLS user plane, by analogy with the B-ISDN/ATM user plane. A central notion is that the OAM of the MPLS G.805 layer network be independent of the server network that carries it or the client protocol operating over it. A number of the requirements are quite generic, for example, requiring rapid detection of defects, diagnosis, localization, and reporting. Other requirements are quite specific to MPLS, including required support for detection of mismerged or unintended replication (e.g., looping) packets. At publication time, some revisions to Y.1710 were being discussed, but most ITU-T effort in this area was focused on standardizing a solution.

At the time of writing, draft Recommendation Y.1711 was under review as a specific solution to these requirements. Derived from the ATM OAM AIS/RDI and CC cells described earlier, there are many similarities with I.610, but several important differences. Regarding continuity checking, the MPLS OAM thinking is to generate continuity verification (CV) packets once per second, with a receiver declaring a defect if no CV packets are received for more than three seconds. The CV packets are always sent regardless of whether the user has sent any packets, in an approach that differs from ATM, where CC cells are sent only if there is no user traffic. Furthermore, unlike the ATM OAM CC cell, which has effectively no content, the proposed MPLS OAM CV packet contains an IP-based source identifier, which helps to diagnose misinserted and mismerged packets.

Regarding alarm reporting, the Y.1711 draft continues the unfortunate tradition of renaming signals, with ATM AIS and RDI now called MPLS forward and backward defect indications (FDI and BDI). Other than the change of name, the addition of the source

identifier defined for the CV packet, and a proposed use of IP-based addressing for the defect location, the operation of MPLS FDI/BDI is very similar to that of ATM AIS/RDI. At the time of writing, the MPLS OAM details for loopback and path trace functions were still under development.

MPLS Protection Switching and Fast Rerouting

Much discussion has occurred in the IETF MPLS and traffic engineering working groups regarding fast rerouting and protection switching [Sharma 02, Lai 02]. The protection switching approach was described earlier in this chapter for ATM as described in I.630. At the time of writing, the ITU-T was pursuing a similar approach for MPLS in draft Recommendation Y.1720. In parallel, the IETF was also defining similar capabilities.

Fast rerouting is more feature rich and therefore a more complex form of restoration. Chapter 14 covered the case of signaled LSP establishment and the way that a label distribution protocol responded to failure conditions by distributing labels to move the traffic to another path around the failure. The basic idea behind fast reroute is to establish a secondary, backup path in addition to the primary path for part or all of an LSP. When a failure is detected, the traffic can then be more quickly restored by rerouting to the already established backup path. The simplest case is to have a backup path for every segment of an LSP. However, a more interesting case is sharing of a single backup path among many primary paths. Another important consideration in such designs is ensuring that the devices and transmission circuits on the primary and backup paths are diverse, so that a single failure does not cause both the primary and secondary paths to become unavailable at the same time. Usually, ensuring diversity requires careful tracking of the transmission and LSR deployments and then configuration of this information into the MPLS routers. Although the concept is straightforward, a fair amount of complexity is involved in advertising the diversity, automatically selecting a shared backup path, determining when precisely to reroute, and deciding when the failure has been repaired to revert to the normal mode of operation.

REVIEW

This chapter described the reference configuration for ATM Operations and Maintenance (OAM) flows at the physical layer and the ATM Virtual Path (VP) and Virtual Channel (VC) levels. The reference model defines connection points for VPs and VCs at the ATM layer. Coverage moved to definition of the OAM cell format and how VPs and VCs use it on either an end-to-end or segment basis. The discussion then proceeded to the topic of fault management, including the definition of the Alarm Indication Signal (AIS) and Remote Defect Indication (RDI). Next, the chapter described the use of loopback and continuity check functions in determining and diagnosing faults where AIS and RDI does not. We then summarized ATM protection switching and its use of ATM OAM functions.

The text then defined the concepts of Network Performance (NP) and Quality of Service (QoS) measurement. QoS is what the user perceives, while NP is what the network uses to make design, operational, or capacity decisions. Next, we described the ATM OAM cell activation/deactivation format and procedure for continuity check and performance measurement. The chapter then described and explained how ATM performance measurement OAM cells can be used to measure ATM NP/QoS parameters in a live network. The chapter concluded with a discussion on directions involving OAM concepts as applied to MPLS. We summarized how the ITU-T is applying the concepts learned from ATM to MPLS in this area. The text also summarizes the emerging area of protection switching and fast rerouting as applied to MPLS.



PART VIII



Design Considerations and Future Directions Involving ATM and MPLS

The telecommunications world continues to change at a mind-numbing pace. As Einstein predicted in his famous theory of special relativity in 1905, strange things happen when traveling at high speeds. Strange things are indeed happening in the world of communications networking that will change the way we live and work together in the

twenty-first century. At the end of the 1990s, it became clear that ATM would never become all things to all people. Instead, Internet-based applications appeared poised to reinvent the business world. But now it appears that bubble may also burst, or at least no longer will it grow at hundreds of percent per year.

While the first seven parts of the book were largely objective, this last part moves into the more subjective areas of comparison and prediction. Most of the text does not present the author's opinions; rather, it categorizes other opinions published in the literature or discussed in open forums within a particular structure. To the extent possible, we attempt to separate hype and hope from hard facts and reality.

Chapter 29 presents several important design considerations involved with ATM, MPLS, and IP as infrastructure for multi-service networking (or a native service) along the major categories of efficiency, scalability, and complexity. The efficiency analysis compares how well ATM, MPLS, and IP support multiple services. The scalability analysis centers on the trade-off between connection-oriented and connectionless networking. Interestingly, ATM and IP are at the extremes, with MPLS taking a middle ground. Next, we discuss more subjective measures of scalability and complexity, highlighting historical advantages and disadvantages, and future trends. The chapter continues with a discussion of other design considerations and how these apply to ATM and MPLS networks.

Chapter 30 gazes into the crystal ball for a glimpse at potential future directions for MPLS and ATM. ATM has less potential for the future but still has some important applications. We summarize some important lessons learned from ATM by MPLS and IP. On the other hand, the future of MPLS- and IP-based multi-service networking has more potential, but it is not without issues. The coverage also includes generalized MPLS (GMPLS), control of optical networking, and the potential separation of control and switching. Finally, the book concludes with a discussion on the possible future of MPLS and ATM multi-service networking.

CHAPTER 29



Design Considerations for ATM and MPLS Networks

The chapter first presents an objective efficiency analysis for the support of circuit emulation, voice, packet data, and video over ATM, MPLS, and IP. We then present a scorecard on how these technologies rank in terms of efficiency in supporting these multiple services. We then discuss scalability considerations for these technologies. The coverage then moves to a comparison of complexity and a subjective assessment of what pitfalls ATM encountered, which MPLS and IP must avoid to be successful in support of a multi-service network. Finally, we conclude with a discussion of other design considerations, including reliability, availability, stability, supportability, operability, and security.

EFFICIENCY ANALYSIS

Despite what some advocates claim, bandwidth is not free, and although the cost per unit of capacity is declining, accelerating demand may increase the overall cost of used capacity for the foreseeable future. The most expensive transmission facilities usually occur in the access or backbone network. Access circuits connect large and small business locations, as well as residential subscribers, to nodes at the edge of a network, while backbone trunks connect these nodes. All protocols have overhead, and the relative efficiency of a particular protocol determines how much information an access or backbone circuit can carry of a specific service type. Currently, access charges are a significant component of public WAN data services, and, therefore, protocols that effectively support multiple services on the same access facility are attractive for this reason. Furthermore, capacity in some areas of the backbone, such as transoceanic cables or transmission facilities to remote areas, is inherently more expensive than capacity in other parts of a network, for example, those parts connected by fiber optic cables owned by a service provider. The motivation is strongest in these expensive parts of the network to utilize the most efficient protocol possible to carry the mix of services traversing that link.

Circuit Emulation Efficiency

All protocols have overhead, but the amount of overhead SONET uses to carry lower-speed circuits may surprise you. The VT1.5 mapping in SONET uses 27 octets every 125 μ s frame inside an STS-1 payload to carry 24 octets of payload (see Chapter 6) in a DS1. Furthermore, the STS-1 section, line, and path overhead uses 36 octets along with 18 stuff octets per frame out of the total $9 \times 90 = 810$ octets in an STS-1. Thus, only 28 VT1.5s (or 672 DS0s) fit into an STS-1, resulting in an efficiency computed as follows:

$$\text{SONET VT1.5 Efficiency} = \frac{24 \times 28}{9 \times 90} = 83\%$$

To carry the same VT1.5 payload, ATM's AAL1 unstructured mode (see Chapter 12), uses one extra octet out of the 48-octet payload in the 53-octet cell. Thus, the ATM AAL1 efficiency in carrying the equivalent user payload in the STS-3c path payload of 9×260 octets (see Chapter 12) is the following.

$$\text{ATM Unstructured AAL1 Efficiency} = \frac{47}{53} \frac{260}{270} = 85.4\%$$

Furthermore, for a decrease in efficiency of less than 0.3 percent, choosing AAL1 structured mode allows connections of arbitrary multiples of 64 Kbps. Additionally, ATM can efficiently pack these structured and unstructured connections at utilization in excess of 90 percent on any link and achieve acceptable delay variation, as analyzed in Chapter 24. Networks must set up connections using the AAL1 circuit emulation protocol and the CBR service category, which requires that devices implement some form of prioritized queuing.

Also, use of the structured mode means that no intermediate cross-connect stages are required in ATM networks implementing circuit emulation when compared with the classical TDM digital cross-connect hierarchy, which reduces cost and complexity. Even if individual ports on an ATM device performing circuit emulation are more expensive than those on the separate TDM cross-connects, the overall cost may be less depending upon the number of ports required to interconnect the levels of the TDM hierarchy.

Although not finalized at the time of writing, circuit emulation over MPLS should be capable of achieving relatively high efficiency as well, especially if packets larger than an ATM cell are used. On the other hand, circuit emulation over IP will be much less efficient due to the 20-octet overhead of an IPv4 header. Efficiency of circuit emulation over IP will be further reduced if 20 octets of UDP and RTP overhead are used for sequence numbering and loss detection, as is being discussed in the IETF Pseudo-Wire Edge-to-Edge Emulation (PWE3) working group.

Additionally, multi-service packet-switching support for circuit emulation can be considerably more flexible than systems built only for circuit switching. This occurs because traditional TDM solutions assign a connection to a particular time slot. If a free time slot of sufficient size is not available, a new connection cannot be set up, even though free bandwidth may be available that is greater than that needed by the new connection. For example, if DS1s and DS3s are being set up on an OC12, a new DS3 connection request may be refused because at least one DS1 is in use in each of the twelve DS3s in the OC12s. What has to be done in circuit networks is to rearrange DS1s to free up a DS3, potentially impacting paying customers. Since packet-switched circuit emulation is a stream of packets at a constant rate, this packing issue does not occur. The major issue is phase alignment between the streams, causing increased delay variation at high loads, as analyzed in Chapter 24.

All right, packet-based circuit emulation is more efficient than STM in terms of basic encoding, and it has better blocking performance and enables statistical multiplexing. Does it have any disadvantages? Yes, it does. As studied in Chapter 16, circuit emulation adds a packetization delay and a playback buffer delay. For lower-speed circuits, packetization delay can be a significant factor. Also, as discussed in Chapter 12, timing recovery or transfer of timing can become quite complex. Finally, TDM circuits have essentially no loss or errors. If the underlying packet-switched network does not have stringent QoS, then the performance of the emulated circuit may not be acceptable. As discussed in Chapter 12, ATM AAL1 defines a standard means to compensate for some

performance impairments in the underlying ATM network through interleaving and error correction coding.

Packetized Voice Efficiency

This section compares the efficiency of the various packetized voice techniques described previously. Table 29-1 shows the overhead (in octets) incurred by each approach, the packetized voice payload size, and the resulting efficiency.

As discussed in Chapter 12, ATM defines two methods for supporting voice: AAL1 and AAL2. The analysis considers cells partially filled to 20 octets for direct comparison with IP and MPLS, as well as completely filled cells. As discussed in Chapter 16, AAL2, voice over IP, and voice over MPLS all support a range of coding techniques, and packet sizes can differ markedly for these techniques. Of course, silence suppression applies in a comparable manner to all of these approaches, but it is not included in this analysis. RFC 2508 defines a means to compress the 40 octets of RTP/UDP/IP headers on a point-to-point access line down to 3–5 octets, as indicated in the table. Indeed, without this header compression,

Overhead	Voice over AAL1	Voice over AAL2	Voice over IP Access	Voice over IP Backbone	Voice over MPLS
AAL	1–2	3–4			
ATM	5	5			
IP			Compressed	20	
UDP			to 3–5 for	8	
RTP			RTP/UDP/IP	12	
HDLC (PPP)			6–8	6–8	6–8
MPLS					9–12
Total Overhead	6–7	8–9	9–13	46–48	15–20
Packet Size	20–47	20–44	20–50	20–50	20–50
Efficiency	38%–89%	38%–83%	59%–82%	29%–52%	50%–76%
Voice Bit Rate (Kbps)	6–32	6–32	6–32	6–32	6–32
Packet Bit Rate (Kbps)	16–36	16–39	14–42	20–65	11–45

Table 29-1. Voice over Packet Efficiency Analysis

IP telephony loses some of its attraction on lower-speed access lines. On an IP backbone network, such header compression cannot be done and efficiency suffers. As described in Chapter 16, the MPLS Forum's voice over MPLS trunking specification adds between 5 and 8 octets of overhead, in addition to the normal PPP/HDLC and MPLS overhead. Voice over MPLS ends up being somewhat more efficient than voice over IP in an Internet backbone. In conclusion, with packet size matched to AAL size, ATM is most efficient for transporting voice on xDSL or DS1/E1 access lines from an efficiency point of view, while IP with header compression is attractive on dial-up lines or lower-speed access circuits. However, on lower-speed access lines, long data frames can impose problems in achieving QoS if not handled by link-level segmentation, as described in Chapter 10, which adds some additional overhead and complexity.

Efficiency of Cells Versus Frames for Packet Switching

While overall efficiency isn't so important in the LAN—after all, 40 percent utilization on a shared Ethernet is considered acceptable—high utilization of expensive WAN facilities is often an important consideration. Recall from Chapter 10 that the choice of the 53-octet ATM cell size was a compromise between low voice delay and efficient data encapsulation. This choice means that while ATM is relatively efficient for voice, it is relatively inefficient for encapsulation of packet data. This section presents an accurate accounting of the ATM cell tax (including AAL5) compared with MPLS and Ethernet to make the point that any encapsulation protocol has an overhead tax of its own. In the end, the relative efficiency of one protocol compared with another is what a network designer should analyze.

HDLC and Ethernet use variable-length packets with overhead of O octets per packet as described in Chapters 6 and 9, respectively. Therefore, the efficiency for these protocols in the encapsulation of a P octet packet is

$$\text{Efficiency(Frame Based)} = \frac{P}{(1 + S)(P + O)}$$

where S is the stuffing overhead and O is the protocol overhead. For MPLS/PPP/HDLC, S is described in the text that follows and O is 10 octets. For Gigabit Ethernet, there is no stuffing overhead, but the overhead O is 42 octets.

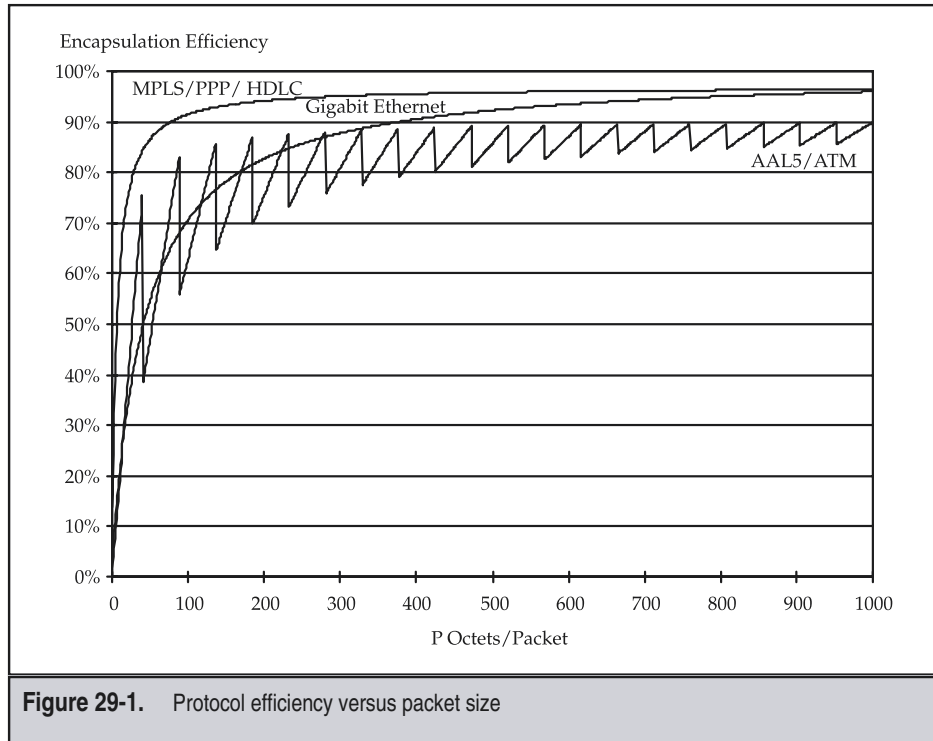
Many analyses covering the efficiency of packet transport over HDLC ignore the increased line bit rate created by the zero stuffing overhead percentage S in the preceding formula. For random data, this results in a 3 percent increase, since HDLC stuffs a zero bit for any sequence of five consecutive ones in the data stream to avoid replicating the six ones in an HDLC flag ('01111110'). For random bit patterns, each bit is a one or a zero half the time. Hence, the likelihood of encountering five consecutive ones is one out of thirty-two, or approximately 3.1 percent. For nonrandom data (for example, all binary ones), the overhead of HDLC zero stuffing never exceeds 20 percent, but it may be less than 3 percent. IP over SONET as defined in RFC 2615 uses octet stuffing for the flag character, which results in an efficiency loss of approximately 1 percent (i.e., 1/128) [Manchester 98].

As described in Chapter 12, the 8 octets in the AAL5 trailer combined with the 5 octets of ATM cell header overhead and the rounding up to an integer number of cells using the pad field yield an encapsulation efficiency of:

$$\text{Efficiency(AAL5)} = \frac{P}{53} \left\lceil \frac{P+8}{48} \right\rceil^{-1}$$

where $\lceil x \rceil$ is the smallest integer such greater than or equal to x .

Figure 29-1 plots the efficiency of encapsulating a P octet packet over MPLS, ATM AAL5, and Gigabit Ethernet. The reason that the AAL5 curve is jagged is that the efficiency calculation rounds up the overhead to a full cell as described by the preceding formula. As can be seen from the chart, MPLS/PPP/HDLC is the most efficient protocol for carriage of packet data. AAL5 and Gigabit Ethernet are quite close in their packet transport tax for smaller packet sizes, but Ethernet is more efficient for longer packets. The average packet size in the Internet is approximately 300–400 octets, and in this range the efficiency



of AAL5 is close to that of Gigabit Ethernet. As we will see in the next section, a more complex calculation that takes into account the distribution of packet sizes is necessary to precisely compute encapsulation efficiency.

Efficiency is not the only consideration in choice of an encapsulation protocol. There are also some differences in features between these encapsulation protocols. Ethernet is limited to a maximum packet size of approximately 1500 octets, while AAL5 and MPLS both provide support for much longer packets. ATM and MPLS both have support for traffic engineering and QoS, while Ethernet supports only a simple form of prioritization. ATM has OAM support that is compatible with Frame Relay, and a similar function is being defined for MPLS. Ethernet may have OAM support defined in the future.

Which protocol is best for your application? If you need a feature that only a less efficient encapsulation supports, then the choice is clear; you can't use a more efficient encapsulation if it doesn't support a critical feature. If raw efficiency is key, then MPLS is a good choice. If interface cost is paramount, then Ethernet may be the best choice. Frame Relay with IP encapsulated in AAL5/ATM is still deployed in many parts of service provider backbones, and there may not be a compelling reason to replace it unless the transmission facilities are expensive and highly utilized. However, in Internet backbones, MPLS is now the technology of choice, at least for new growth.

IP/ATM, IP/MPLS, and IP/SONET Efficiency

The preceding analysis assumed that each packet had the same size. Real traffic, however, has packets of many different sizes as reported in studies of IP network traffic [Thompson 97]. Figure 29-2 plots a typical measurement of the relative frequency of occurrence of packets of various kinds on an IP backbone versus the IP packet size. For this sample, the average packet size (with overhead) was 360 octets. Approximately 30 percent to 40 percent of the packets are TCP/IP acknowledgment packets, which are exactly 40 octets long. The majority of packets are less than or equal to the maximum Ethernet frame size of 1500 octets. Less than one in 10 million packets have lengths greater than 1500 octets, reflecting the predominant use of Ethernet in end systems. Recent measurements of Internet backbone packet sizes indicate a similar distribution.

Table 29-2 illustrates the protocol overhead required when using IP over various frame-based and ATM-based protocol stacks. As pointed out in Chapter 18, the LLC/SNAP multiprotocol encapsulation defined in IETF RFC 2684 adds an extra 8 octets of overhead when carrying multiple protocols over ATM. Although this standard targeted multiprotocol applications between end systems, many backbone routers use this same form of encapsulation, for example, when IS-IS is the interior routing protocol. This means that the 40-octet IP packets with 8 octets of AAL5 Common Part (CP) overhead plus the 8 octets of LLC/SNAP overhead require two cells instead of the single cell that would be required if VC multiplexing were used instead (see Chapter 18). Note that IP operating directly over HDLC adds up to another 2 octets of PPP overhead per RFC 1662 and that RFC 2615 recommends a 32-bit HDLC CRC for IP operating over SONET. We also include PPP running over Frame-based UNI (FUNI, see Chapter 17) in the comparison. The table also includes MPLS running over Gigabit Ethernet (GbE) and MPLS/PPP/HDLC.

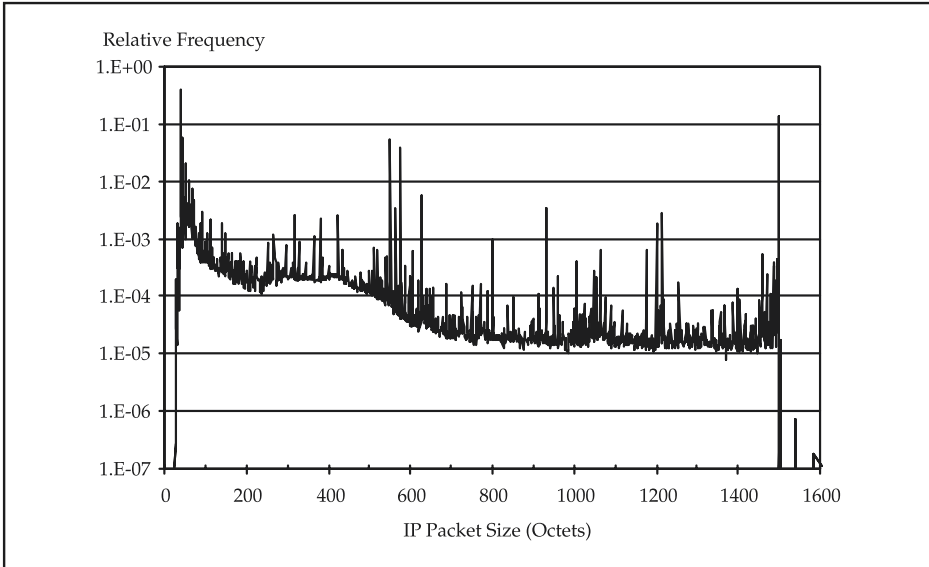


Figure 29-2. Relative frequency of IP packets versus packet size

Overhead Field	PPP/SONET	LLC/SNAP over AAL5	VC Mux/AAL5	PPP/FUNI	MPLS/GbE	MPLS/PPP
HDLC	8 octets			2 octets		8 octets
FUNI				8 octets		
Zero Stuffing	1%			1%		1%
Ethernet					34 octets	
53-Octet Cell	No	Yes	Yes	No	No	No
AAL5 CP		8 octets	8 octets			
LLC/SNAP		8 octets			8 octets	
PPP	2 octets			2 octets		2 octets
MPLS label					4 octets	4 octets

Table 29-2. Overhead for Various Protocols Carrying IP Packets

The average encapsulation efficiency is then a function of weighting the efficiency of each packet length accounting for the overhead in Table 29-2 by the relative frequency of each packet size (i.e., the probability $\text{Pr}[P]$) determined from the histogram of observed packets sizes as follows:

$$\text{Efficiency} = \sum_{P=1}^{\text{Max}} \text{Pr}[p] \text{Efficiency}(P)$$

where the $\text{Efficiency}(P)$ represents the formula for protocol P ; for example, as given by the equations for a frame-based protocol or AAL5 as described in the preceding section.

Computing the relative efficiency for each of these protocols using the measurements reported in [Thompson 97] results in the efficiencies shown in Table 29-3. Other studies [Armitage 95] using actual packet distributions also measured approximately 80 percent raw efficiency for LLC/SNAP over ATM AAL5. Unfortunately, other studies cite the ATM overhead as the relative difference and make the erroneous assumption that other encapsulations have no overhead. Since by definition encapsulation means adding overhead, clearly, native IP operating over PPP over SONET (POS) is the most efficient, at almost 97 percent efficiency. It is approximately 16 percent more efficient than ATM using AAL5 and LLC/SNAP multiprotocol encapsulation. Adding MPLS to POS reduces efficiency by another 2 percent, so that the difference between MPLS and LLC/SNAP/AAL5 narrows to 14 percent. It is interesting to note that the efficiency of IP over gigabit Ethernet for this set of data is identical to that of IP over AAL5 with LLC/SNAP. If you hate ATM cell tax, then note that Gigabit Ethernet levies the same penalty. Note that a network using the Frame-based UNI (FUNI) protocol achieves efficiency comparable to native IP over SONET. If LLC/SNAP encapsulation is removed, then the overhead of running IP over ATM is reduced by approximately 4 percent.

Some studies have shown that the resulting efficiency is relatively independent of the choice of cell size. Therefore, the fundamental difference exists between the choice of a

IP Operating over the Protocol(s)	Encapsulation Efficiency
PPP over SONET (POS)	96.6%
ATM AAL5 using LLC/SNAP Multiprotocol Encapsulation	80.8%
ATM AAL5 using VC Multiplexing	85.1%
PPP over ATM Frame-based UNI (FUNI)	96.1%
MPLS over PPP over SONET (POS)	94.6%
MPLS over Gigabit Ethernet	80.8%

Table 29-3. Efficiency of IP Transport over Various Encapsulation Protocols

fixed packet size (e.g., a cell) and a variable-length packet (e.g., an HDLC frame). Note that when operating over SONET and SDH links, ATM or any other protocol has the same synchronous payload envelope rate listed in Chapter 11. For example, the actual rate available to any protocol operating over a 155.52 Mbps link is only 149.76 Mbps. Therefore, SONET itself introduces a 3.7 percent overhead at the OC3/STM-1 rate. At higher speeds, the SONET overhead decreases slightly (see Chapter 12). Nothing is certain but death and taxes. As seen from this analysis, the ATM cell tax can be reduced by operating IP over other protocols, but not eliminated. And some technologies that, like Ethernet, may have a lower port cost are actually just as inefficient as ATM in the transport of Internet packets.

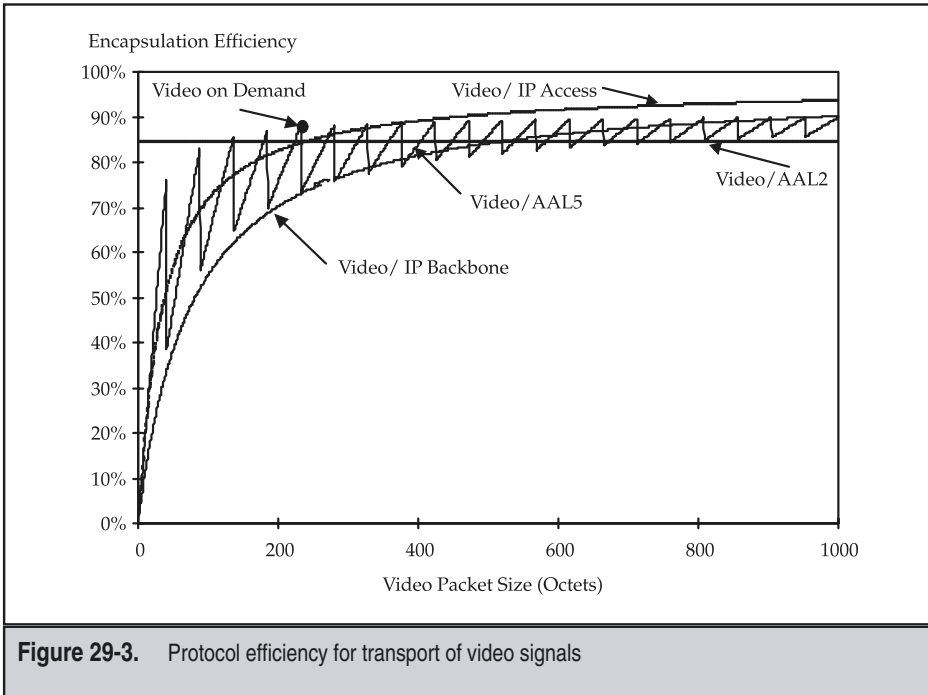
Packet Video Efficiency

This section rounds out the efficiency comparisons in this chapter by briefly comparing the efficiency of carrying packet video over ATM and IP. Recall from Chapter 16 that a number of video coding standards generate packetized video at rates ranging from 100 Kbps to over 40 Mbps; for example, MPEG2. An IP network must carry the entire header field across the backbone, while an ATM network can convey addressing information during call establishment and not send information in the header field once the connection is established. The analysis assumes an IPv6 packet header to support the extended addressing capabilities instead of dynamic address assignment as is done today to allow assignment of an address to every subscriber. The 20-octet ATM End System Addresses (AESAs) could meet this requirement, as studied in Chapter 13. As covered later in this chapter, a connection-oriented protocol generally provides a more economical solution for long-lived information flows, such as video information transfer. Also, in some cases a video packet is more efficiently encapsulated in AAL5.

Figure 29-3 illustrates the protocol efficiency of video carried by AAL5, AAL2, and IP protocols on access and backbone circuits for packet sizes up to 1000 octets. The analysis assumed that AAL5 overhead was only 8 octets because all other parameters were conveyed in the setup message. The AAL2 overhead assumes a constant 8 octets of overhead per cell, and since it runs in a streaming mode completely filling each cell, it results in a constant efficiency of approximately 85 percent. The analysis assumes that on backbone circuits, video over IP uses RTP running over UDP using IPv6 over PPP with HDLC framing for a total of 77 octets of overhead per packet. On access circuits, the figure depicts the resulting efficiency assuming that IPv6 and UDP header compression reduces the overhead by 42 octets.

Video coders typically send packets on the order of several hundred octets; for example, MPEG2 uses a packet size of 188 octets. If the video-to-ATM mapping accounts for the round-off efficiently (as described in Chapter 16 for the ATM Forum's video on demand specification [AF VOD 1.0]), then video over ATM is more efficient than IP on the backbone by 8 percent and slightly better than the performance achieved using header compression by video over IP on access circuits.

Making the packet size larger to improve video over IP efficiency creates several problems. First, packet lengths of 1000 octets create a significant packetization and



transmission delay at xDSL speeds (i.e., 1 Mbps). This amount of delay could reduce the interactive nature of other services such as voice and interactive network game applications on a lower-speed access line. Recall that voice over packet service needs to minimize delay, so creating more delay by using long video packets exacerbates this issue. Second, video coders send frames at least 30–60 times a second (for example, for basic updates and detailed information). Furthermore, the video session also involves transmission of side information such as audio, closed caption, and other video-related data. Therefore, a video coder sends packets at least one hundred times a second, which results in a packet size of approximately 1000 octets for a transmission rate of 1 Mbps. Furthermore, many video coders include some redundancy in case video packets are lost or errored. Finally, many video coding algorithms require some form of timing recovery to synchronize the video, audio, and ancillary information streams for playback purposes. Large gaps or variations in packet arrival times complicate timing recovery at the receiver. Therefore, video coders generally use smaller packets.

Multiservice Efficiency Comparison

All right, now that we've looked at the efficiency comparisons for TDM circuits, voice, packet data, IP, and video, how do they stack up in the big picture? Table 29-4 shows the

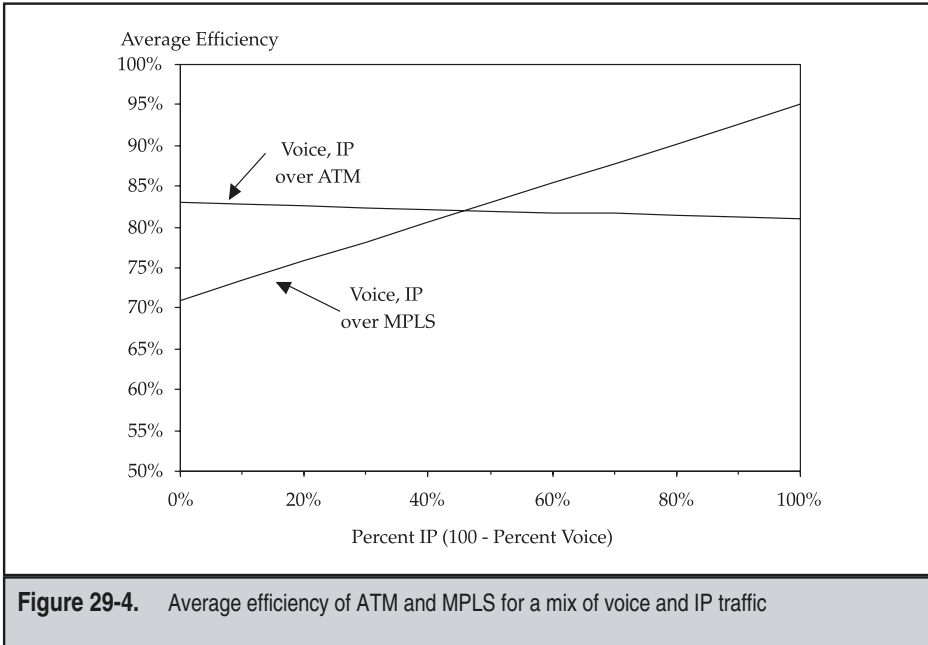
Service	TDM Circuit	Voice	Packet Data	IP	Video
ATM	1st	1st	2nd	3rd	1st
MPLS	2nd	2nd	1st	2nd	2nd
IP	3rd	3rd	3rd	1st	3rd

Table 29-4. Relative Protocol Efficiency Comparison Scorecard

scorecard of the efficiency analyses from the previous sections. The rows are placed in golf score order, with the lowest total being that for ATM, followed closely by MPLS. However, since ATM is relatively inefficient in support of IP, the scores do not have equal weight in many applications. The overall measure of efficiency, of course, depends upon the relative volume of each service, as discussed next. However, this table shows some of the rationale behind the decision by some carriers to deploy ATM, since it handles multiple traffic types in a relatively efficient manner using a common infrastructure. However, MPLS has relatively good efficiency for every service and only adds a nominal amount of overhead while supporting IP well and adding the tremendous benefit of traffic engineering. So, from an efficiency point of view, MPLS appears to be a promising multi-service infrastructure.

If more than one service is multiplexed over a transmission link, then the ratio of the services determines what is the overall most efficient solution. Let's look at a simple example of the efficiency of carrying different mixes of voice and IP packets over a shared transmission link using the ATM or MPLS encapsulation efficiencies computed earlier. That is, the efficiencies are 83 percent for voice over ATM AAL2, 71 percent for voice over MPLS, 81 percent for IP over AAL5 using LLC/SNAP, and 95 percent for IP over MPLS over POS. Figure 29-4 shows the result of this comparison by plotting the average efficiency of a mix of IP and voice traffic indicated on the horizontal axis. For 100 percent voice traffic, ATM is the most efficient. But as the fraction of IP traffic increases above 40 percent, MPLS becomes more efficient for a mix of traffic that contains more IP than voice. As mentioned before, overall IP and data traffic exceeded the level of voice traffic at some point in the late 1990s. Therefore, even if MPLS is not the most efficient means to carry toll-grade voice, it may often be the most efficient way to carry a mix of traffic that is predominately IP.

Of course, in order for this to work in a network with multiple vendors or service providers, interoperable standards for multiservice networking over MPLS must first be completed. However, efficiency alone does not dictate choice of a particular protocol. In addition to standards and feature requirements, other design considerations can be more important in certain networking circumstances.



SCALABILITY ANALYSIS

This section moves away from the objective approach of efficiency computations into the more subjective area of relative scalability.

Addressing and Hierarchy

Addressing may be geographically oriented like the E.164 telephone numbering plan or like the Internet, where blocks of IP addresses are assigned to service providers or large enterprises. The ATM NSAP address model also assigns addresses at an enterprise level. Addresses have limited utility unless an entity associated with an address can reach other entities in that address space. Agreement on a technical standard for addressing is usually an easy task compared with resolving the political, economic, business, and social issues that arise in achieving this reachability.

The IPv4 address is 32 bits, while the IPv6 address is 16 octets. Careful design and allocation of the IP address space has resulted in controlled growth of the number of forwarding table entries in the global Internet, which at the time of writing was on the order of 100,000, as described later. The E.164 addressing plan is 15 Binary Coded Decimal

(BCD) digits and currently aggregates well on a country code or service provider prefix basis. The ATM Forum PNNI specification mandates the hierarchical organization of the 20-octet NSAP-based address. ATM SVCs utilize either the E.164 or 20-octet NSAP-based addresses.

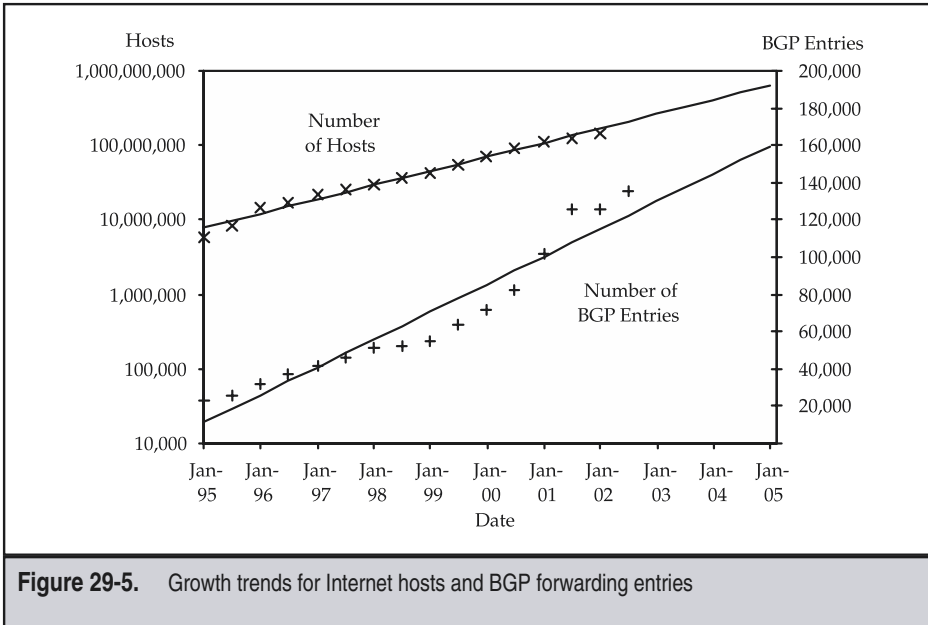
Coordinated assignment of the address space is a key enabler to global scalability for both connection-oriented and connectionless protocols, since it determines the size of the routing table in switches and routers. Both IP and ATM NSAP-based addresses support the notions of hierarchy and summarization essential to large networks. ATM PNNI has more done on the conceptual design of hierarchy as it relates to scalability than has IP or MPLS; however, IP and MPLS have scaled quite well despite a lack of this style of hierarchy.

Supported User and Routing Table Growth

In order to truly appreciate how well the connectionless routing and forwarding paradigm of the Internet has scaled historically, Figure 29-5 plots the number of hosts [Hobbes 02] and the number of BGP forwarding table entries [Telstra 02], from 1995 through 2002. Note that these values are plotted on different scales: the logarithmic scale on the left is for the number of hosts, while the linear scale on the right is for the number of BGP forwarding entries. The lines project these trends out another few years. Note that a straight line on a semilog graph for the number of hosts indicates exponential growth for the number of hosts at a 55 percent cumulative annual growth rate. On the other hand, the number of BGP forwarding table entries grew in a linear manner, increasing on average by almost 15,000 entries per year.

The time during the late 1990s when the slope of the BGP entries increased markedly was of great concern to Internet service providers and the IETF, with predictions of the impending collapse of the Internet a real concern if the table capacity of legacy routers was exceeded. However, since the growth rate of BGP forwarding entries is linear, Moore's law of electronic storage capacity tells us that new routers will be able to easily meet this scaling challenge. Therefore, the concept that MPLS reduces the need for forwarding table capacity in the core of a network is not addressing a real scalability problem, since the core is where the newest, highest-performance routers are traditionally deployed. Interestingly, since the Internet bubble burst, the BGP entries' growth rate seems to have slowed down again. Since BGP message processing is proportional to route entries, and not hosts, the connectionless paradigm of the Internet with its service provider-based assignment of address prefixes has an inherently superior scalability when compared with per-destination flow switching, like that done for ATM VCs or VPs or extremely granular MPLS FECs.

The way that routing protocols exchange topology information presents another scalability challenge. The flooding of link-state updates in response to certain events (e.g., a failure) can produce significant transients of messages and associated processing. The end result is that the slowest processor effectively constrains the number of nodes that can exist within a single interior routing domain. The current wisdom on the way to address this issue is to partition such a domain into multiple domains, which may have a hierarchical relationship. As discussed in Chapter 14, OSPF and IS-IS interior routing protocols



support two levels of hierarchy. However, networks containing a very large number of nodes may require more levels of hierarchy [McDysan 00a]. As discussed in Chapter 15, the ATM PNNI protocol [ATMF PNNI 1.0] supports more levels of hierarchy but still requires careful network design to avoid processor overload during waves of connection establishment attempts, for example, in response to a failure [Felstaine 99]. At the time of writing, the IETF was still determining whether to use the experience from PNNI hierarchical routing for traffic engineering or to continue the search for a better constraint-based routing solution for MPLS.

Packet Forwarding and Moore's Law

Although the routing infrastructure of the Internet appears to be scalable, an assessment of the forwarding challenge is difficult because there is no good published data regarding the rate of Internet traffic growth. The best we can do here is go with an industry consensus that the long-term historical growth rate is approximately 100 percent, although during the Internet bubble, claims of annual growth rates that were hundreds of percent were not uncommon. A 100 percent annual growth rate of traffic is still a daunting challenge, though, as discussed in Chapter 2, Moore's law corresponds to approximately only a 60 percent annual increase in the speed of the electronics used to build ATM switches or MPLS LSRs. In order to respond to this challenge, network designers must make changes

to the architecture, since the speed of a particular technology cannot keep up with such a large sustained rate of growth. The move to hardware-based ATM switches away from software-based routers at the core of an IP network was the first such architectural change made in the mid-1990s. The late 1990s ushered in the next phase of architectural change with MPLS LSRs that had higher-speed interfaces because the hardware was simpler than that required for ATM. For example, ports did not need to implement AAL5 segmentation and reassembly (SAR) and ATM mapping to the physical layer. And the next architectural phase in pursuit of continued forwarding scalability may well be IP-controlled wavelength or optical switching, as discussed in the next chapter.

Connection-Oriented Versus Connectionless Paradigms

Another fundamental difference between ATM or MPLS and IP is the basic routing paradigm. As studied in Chapter 5, ATM (like MPLS) uses a connection-oriented paradigm like that employed in the telephone network, whereas IP uses a connectionless paradigm where devices use the header within each packet to determine the next hop toward the destination. The connection-oriented paradigm must maintain state for every flow, on the order of 100,000 connections per OC3-worth of capacity for today's Internet traffic. On the other hand, connectionless routing protocols require forwarding tables on the order of 100,000 entries for an entire router. Each port uses the same (logical) forwarding table. Hence, the connectionless design scales better than the connection-oriented paradigm for Internet traffic. The key difference is aggregation of addresses in the connectionless design. Note that aggregating flows using hierarchy as described in Chapter 21 would improve scaling of connection-oriented ATM or MPLS networking, since only the distributed edge switches or routers would need to perform per-flow processing. Virtual paths for ATM or label-stacked LSPs for MPLS could be used to implement such a hierarchy.

Flow-oriented switching does not scale to support the current best-effort Web traffic observed on the Internet. Supporting 100,000 simultaneous flows on an OC3, each lasting an average of 30 seconds [Thompson 97], would require signaling for the initiation and termination of over 3000 flows per second. Current ATM switches and MPLS LSRs process at best hundreds of such requests per second. If we consider only the 20 percent of the flows that comprise 50 percent of the traffic [Thompson 97], then the per-flow signaling rate is still 600 per second per OC3, still a large number. MPLS handles this for IP networks by assigning many individual IP flows, for example, those with destination address prefixes associated with a particular egress router to a single FEC and automatically setting up an LSP to that egress router. In IP over ATM networks, the ATM virtual connection performed a similar function by providing a manually configured logical trunk to the egress router for the FEC. In essence, history has shown that the paradigm that is scalable is to route once and then switch thereafter for packets from many flows with a common destination.

Support for a Wide Range of Interfaces and Speeds

Scalability in terms of interface speed and nodal capacity not only has meaning in terms of ability to support high speeds and large capacities, but also has an often overlooked aspect of being able to support lower speeds at small nodal capacities in a cost-effective manner as well. Let's look at the frequently publicized history of high speeds and large-capacity switches and routers first.

Scalability at the upper end is focused on how to build large networks with high-speed interfaces and large capacity nodes. In 1995, ATM switches had an edge over IP routers because ATM was the only technology capable of operating at 150 Mbps, with 600 Mbps achieved in 1997. However, the development of hardware-based routers and layer 3 switches driven by the tremendous demand for Internet services changed all this [Keshav 98, Kumar 98]. In the latter part of the 1990s, IP routers reached OC48 speeds before ATM switches did, and they continued on to support OC192 (10 Gbps) speeds in 2000. At the time of writing, the latest generation of hardware-based routers were targeting support for 40 Gbps interface speeds. However, the fastest interface that ATM switches supported in 2002 was OC48 (2.5 Gbps). Furthermore, a number of router manufacturers tout plans to support routers with a total interface capacity in the terabit-per-second range, while the largest ATM switches available supported only 500 Gbps. Therefore, if you need the highest-speed interface and the largest capacity available, then an MPLS label switching router (LSR) is the only choice. However, if your needs are more modest or you already own ATM switches that support multiple services, then continuing to grow an existing ATM network may make sense. Additionally, as discussed in Chapters 11 and 17, you may soon be able to get standards-based ATM over MPLS implementations that could trunk parts of a legacy ATM network.

Scalability concerns not just the fastest interface, largest node, or maximum network size, but also geographic reach and cost-effective implementation of smaller nodes with lower-speed interfaces. Data communications speeds range from those for analog dial-up access up through TDM access over twisted pair or coax. Here, often technologies other than MPLS or ATM are prevalent, such as Frame Relay or PPP. However, ATM was the first to standardize an nxDS1 or nxE1 type of inverse multiplexing, but now comparable standards are defined for Frame Relay and PPP here as well. Usually, a service provider network has a device that aggregates traffic from a number of such lower-speed interfaces and multiplexes them onto a higher-speed trunk. The high-speed trunk side of such a multiplexer may well use ATM or MPLS, with traffic engineering and QoS capabilities as a means of feeding a core network. This division of responsibility (even for devices located in the same building) often gives a network designer the freedom to deploy the most cost-effective solution at the capacity level required. Historically, this divide and conquer approach has worked better than monolithic solutions targeted at being both a core and edge device.

Capacity Bottlenecks

The wide-scale adoption of 10/100 Mbps Ethernet to the desktop and gigabit Ethernet in the LAN backbone is ushering in a new era of almost limitless bandwidth in the local networking area. But with all things comes a price. While these technologies allow LAN environments to scale into the tens of gigabits per second range, WAN access bandwidth of even many megabits per second presents a significant bottleneck if even a fraction of the traffic must go someplace else. This is a scalability challenge that results from the limited physical reach of fiber networks and the relatively high cost of extending the fiber network through new construction. Unless a service provider can connect a number of customers to a newly installed fiber, the price that must be charged is higher than most customers are willing to pay.

The most successful response to this scalability challenge has been to try to increase the speed of already installed network plant, such use of DSL over carefully selected twisted pairs originally installed for telephony, or use of cable modems over plant originally designed for distribution of television programming. As service providers extend their distribution networks to feed DSL and cable modem feeder networks, the fiber optic cables sometimes reach other customers. In the end, the only way that the scalability challenge of geographic reach will ever be addressed is through installation of fiber to those customers who will pay for high-speed access. This has already occurred in some Scandinavian countries as driven by government fiat. Whether this will happen as a result of competitive forces in other parts of the world remains to be seen.

COMPLEXITY ANALYSIS

We now move on to another consideration whose very name implies difficulty, namely the trade-off between complexity and function that has been the challenge of communications from the very beginning.

To Switch or Not to Switch? An Answer to This Question

The signaling for establishing or releasing a connection adds complexity to an application and the network. For this reason, computer applications that can use a connectionless infrastructure are less complex and more widely deployed. However, there are cases where the complexity of establishing a connection is warranted. Connection-oriented service does incur request-processing complexity, but it then sets up forwarding tables only once and then the forwarding of all subsequent packets is greatly simplified. On the other hand, routed connectionless services incur a greater degree of processing complexity for each packet but avoid connection processing complexity altogether. If connection processing adds an important feature, like that of admission control to guarantee QoS for a specified level of traffic, then the complexity of signaling may be justified—if the flow is of long enough duration. Voice or video on demand services that traverse parts of a network where congestion may occur may justify such complexity, and although the IP

RSVP and ATM SVC protocols provide this function, they have not seen wide-scale deployment. However, the IETF has not given up on this paradigm, and the next steps in signaling (NSIS) working group is working on a solution to this very problem. A fundamental tenet, though, is that the solution must be simpler than RSVP, which suffered from too much complexity in trying to solve the challenging problem of multicast, when most applications are either unicast or broadcast.

Keep It Simple to Succeed

The history of data communications presented in this book is replete with examples of the simpler protocol winning out over the more complex one. In the computer room, the complex mainframe was king until the simpler and less capable minicomputer came along, and then the even simpler PC took this a step further. In the LAN, FDDI and Token Ring were too complex, losing out to the simpler Ethernet technology, which effectively scaled one thousandfold from 10 Mbps to 10 Gbps in a period of less than 20 years. This scaling of Ethernet eclipsed the brief entry of ATM LAN emulation, largely due to the plug-and-play simplicity of Ethernet. In the WAN, the ITU-T vision of first trying to put every telephony feature and interoperability workaround into narrowband ISDN and provide multi-service support and then trying to do essentially the same thing again with broadband ISDN failed due to excessive complexity.

What lessons can be learned from this experience such that MPLS has a better chance of reaching the holy grail of multi-service networking? One promising area is that of proper application of protocol layering and a divide-and-conquer mentality being pursued by the IETF PWE3 and PPVPN working groups. Instead of trying to build many features into MPLS, the approach is to assume MPLS or IP as a simple tunnel and then build other protocols on top of this infrastructure in support of other services. This is the approach that historically worked for IP, which is a simple protocol that has been hugely successful.

Hardware Is Hard, but Software Is Harder

All modern communications systems have both hardware and software. Most people experienced in the business of networking would agree that the design, development, and operation of hardware is hard, but those aspects that involve software are often even harder. Keeping equipment and operating costs reasonable presents ongoing challenges for network designers and operators that involve trade-offs in hardware and software complexity. In general, manufacturers do not commit a design to hardware until the problem is defined in great detail, preferably standardized so that the hardware components can be reused in multiple products to amortize hardware design, development, and tooling costs over as many units as possible. An exception to this maxim is the increasingly cost-effective capacity of a field programmable gate array (FPGA), which is a chip full of gates that can be reconfigured if necessary. However, the best price performance for hardware results from a purpose-built application-specific integrated circuit (ASIC) optimized to do a particular function. However, if a vendor gets an ASIC wrong, redoing

this work can be an expensive proposition. Other aspects of ATM and MPLS system hardware design leverage commercially available processor, memory, transmission adaptation, and communication interface chips produced by component vendors. MPLS has one significant advantage over ATM in terms of hardware complexity, as studied in Chapter 11. MPLS can run over a number of link layer protocols (e.g., Ethernet, POS, or ATM), while an ATM switch must have specific hardware that implements the ATM and AAL functions.

Software is often the province of things that are relatively new, are very complex, are subject to change, or are an area of the system where flexibility is necessary. An ATM or MPLS software system may start with components purchased from suppliers, such as an operating system, communication protocol stacks, routing protocols, management protocol stacks, and device drivers. But then a vendor of a switch or router must integrate these purchased components together and add other functions unique to their system. Because so many functions are implemented in software, the overall set of programs is quite a complex collection of interacting, moving parts. And in a communications network environment, software in a node often interacts with software in many other nodes, increasing the networked complexity further still. Human beings inevitably make errors when developing this complex software system, and new versions of the software must be produced to fix the most significant bugs in a timely manner. Add to this the continual change in communication protocol standards and the drive for vendors to add new features and enhancements, and the result is periodic major releases of new software versions, which always introduce some errors, and the bug fix cycle begins anew. As discussed later, the frequency of change required for bug fixes and feature upgrades is a major factor in the reliability and stability of a network. Here, the ideal trade-off would be to enable just enough features (complexity) by installing just enough software releases that have been tested and found to be free of major errors (stability).

Operating a large network of multi-service ATM switches and/or MPLS LSRs, each a complex system in its own right, presents yet another order of operational complexity. In response to this need, network management software solutions are available commercially from vendors as well as third parties. Furthermore, in many cases, service providers and enterprises will develop software to support parts of their operations. Most modern data networks employ routers to some extent. Another important consideration in the overall cost of networking is the amount of investment required in such network management and operational support software and the people required to run the network. Here again, there is another difficult trade-off in implementing the right amount of support software complexity versus employing so many technicians to manually configure and run the network that they end up getting in each other's way.

Are QoS and Bandwidth Reservation Really Necessary?

QoS and bandwidth reservation introduce complexity. Are they necessary? If so, what is the right amount of complexity? This has been a topic debated for many years. There are several points of view: they are not needed if you install enough capacity, they are not needed because applications will adapt, they are needed to make a network cost-effective,

and they are needed in support of applications that have differing quality requirements. You will still see the assertion that all that is needed is enough capacity that congestion rarely occurs. This is true in a networking environment where transmission capacity and router/switch ports are inexpensive (e.g., LANs) or are overbuilt (e.g., the Internet capacity glut of the early twenty-first century). However, there are other networking contexts where there is strong economic motivation to better balance the load and invoke automatic procedures to ensure that congestion rarely occurs.

Philosophically, QoS and bandwidth reservation create inequality [Antonov 96]. Basically, user requests that are admitted by ATM or MPLS signaling protocols get the QoS and bandwidth they require, while those who pay less, or are latecomers, do not. An alternative to such a blocking system could offer graceful degradation in the event of overflows via adaptive voice and video coding. Some voice and video codecs have an avalanche performance curve in terms of degradation in the face of increasing loss and delay variation: they degrade gracefully until a certain point where they become unusable. However, as described in Chapter 22, other codec designs are available that use discard probability to distinguish between the essential information for image replication and the nonessential fine detail. Both ATM and MPLS support selective discard: ATM uses the cell loss priority (CLP) bit, while IP and MPLS can use the discard preference field of DiffServ Assured Forwarding (AF). During congested conditions, the network discards only the fine detail information, and hence the video and audio are still usable. When the network is not congested, then the user receives higher quality.

ATM designers made the decision that yes, QoS and bandwidth reservation are required and developed a set of very detailed specifications on how to implement these functions. History will probably show that they added too much complexity. RSVP was another great effort to implement QoS and bandwidth reservation, and as discussed in Chapter 13, it was never widely used for end-system use in signaling reservations for which it was designed. History will likely also show that it too was more complex than was necessary. However, MPLS designers have learned from these mistakes, and as described alongside their ATM counterparts in Part 5, at least the initial MPLS standards for QoS and bandwidth-related functions are quite minimalist, and hence less complex than ATM or RSVP. Internet service providers seem to be doing well with these MPLS implementations in their infrastructure, so this is an example of adding just enough complexity to achieve the desired traffic engineering benefit. However, it remains to be seen what will be necessary for IP and MPLS to fully support multi-service networking.

The following set of questions and answers illustrates the fundamental inequality of QoS and reserved bandwidth and whether applications and users will require and pay for it. When you connect to a Web site, are you willing to experience only part of its potential audio, visual, video, and interactive capabilities? Possibly. What if the quality were so poor that you had to go back and visit that site at some later time? Sure, everybody has to do that sometimes. What if you were paying for the experience, as you do when you watch a movie, receive educational training, or receive consultation? If the quality were poor, would you still pay? No, of course not, or at least not without complaining.

RELIABILITY, AVAILABILITY, AND STABILITY

Reliability is a measure of how frequently a thing fails. For example, the mean time between failures (MTBF) of a specific component is a measure of reliability. Availability is a measure of what fraction of time a system is available to provide a specific service. For example, if a component requires a mean time to repair (MTTR), then the availability of that single component to provide service is:

$$\text{Ability} = A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

If a single component does not meet the availability requirement of a specific service, then a common design approach is to deploy a backup for that component in parallel. The availability of such a parallel system then increases markedly, as given by the following formula:

$$\text{Availability(Parallel)} = 1 - (1-A)^2$$

where A is the availability of a single component given by the preceding formula.

High reliability and availability starts in the design phase but must be carried throughout the system life cycle to meet user needs. Not only must the design be analytically validated but also simulations and tests should be performed before deployment. Once a technology is deployed, measurements should be collected on failure rates and repair times, with significant differences from the design goals fed back to developers. Also, since customers increasingly rely on services supported by MPLS and ATM network infrastructures, there may be a service-level agreement (SLA) offered by a service provider that often includes measures of reliability and availability.

ATM inherits a penchant for reliable and highly available operation from over a century of experience on the telephone network, since many manufacturers building ATM equipment use telephony design philosophies. On the other hand, Internet technology is only a few decades old, and it was commercialized only within the past decade. It was designed for highly resilient operation even if some components may not be reliable. Overall availability in the face of frequent failures is reduced because the repair (i.e., recovery) time of routing protocols is sometimes on the order of minutes. The current direction of MPLS is to reduce this repair time to seconds, or even less than a second, which should improve availability and reduce the duration of application-impacting service interruptions.

Change creates the potential for instability. Installing a new software version with an undiscovered bug in an MPLS or ATM network can cause an outage. A misconfiguration of a single switch or router can also cause an outage or cause a network routing protocol to become unstable, impacting service quality. Also, frequent state changes in switches or routers, their ports, and the links that interconnect them can also cause a routing protocol to become unstable. Whenever a state change occurs, the routing protocol must distribute the updated topology information, and packets may begin following another path. A change in routed path can result in jittered, lost, mis-sequenced, or, in some cases, duplicated packets. Although usually infrequent, these events can cause annoying effects in

certain applications, such as voice or video. Customers may not be as tolerant of such impairments when they are paying for higher quality as compared with traditional Internet best-effort service.

SUPPORTABILITY AND OPERABILITY

The experience of the Internet bodes well for the prospects of having a set of skilled developers, operators, and administrators for MPLS- and IP-based networks. On the other hand, an ATM expert is becoming an ever more specialized breed. The talent pool of designers, developers, and operators for a particular technology is an important consideration of supportability. Some technologies are now being retired because there are no longer sufficient people skilled in these areas, or people who have the requisite skills would prefer to work on something else. Such has been the fate of X.25, FDDI, and SMDS, to name a few.

The underlying complexity of a technology has a strong influence on what it costs to operate a network. The degree of automation afforded by the network and/or the management system greatly determines the overall operational cost. A complex network protocol that automates many functions may require highly skilled, expensive operators but fewer of them. On the other hand, complex protocols requiring detailed configuration and intelligent monitoring require not only operators with greater skill but more of them and hence greater cost. Simpler protocols can use operators with less skill at lower cost, but if the processes do not have automation support then you may need a small army of operators to run the network. This is in general not a good idea, because often only a few people can access any switch or router at the same time, and after a point, too many people working on related tasks tend to interfere with each other.

SECURITY

We would be remiss if we did not cover security. Security has several interpretations. A traditional interpretation is the degree of confidentiality, integrity, and authenticity of information. Some experts believe that cryptographic methods are necessary to meet these requirements. There is another interpretation of security that relates to how well a network ensures that only those parties who are configured to exchange certain information can in fact do so. Both ATM and MPLS rely on this security by configuration paradigm. For example, when properly configured, only the source and destination interfaces in a provider network for an ATM virtual connection can receive cells sent by customer devices sent on these interfaces. Within a service provider network, it may be possible to tap into an information stream, and even to modify it. Here, MPLS and ATM rely on the physical site security of a provider or enterprise to make such actions difficult, if not impossible. Since configuration and physical site security have the potential for human error and compromise, this is why security experts recommend cryptographic protection for sensitive information. However, often the bulk of an enterprise's information does not

justify the expense of cryptographic equipment and administration, and therefore the configuration and site security are often deemed sufficient.

When used as an infrastructure for support of other services, MPLS and ATM have an additional level of configuration security. This results from the fact that at the edge of such a network there is additional configuration of the adaptation of the service to the ATM or MPLS infrastructure. In essence, there are more things that must be misconfigured to compromise security. This is a fundamental driver for the difference of function of a user-to-network interface versus a network-to-network one (i.e., UNI vs. NNI), where the NNI has both routing and signaling support, whereas a UNI can have at most the signaling function. By not opening up the routing function, the network is protected from external sources of instability. In a permanent connection service, excluding signaling adds further protection as well.

Finally, there is an aspect of security regarding how resilient a network and its management system(s) are to attacks by those who would strive to disrupt or deny service. This begins with configuration security, and restriction of routing, and in some cases signaling, interfaces to only those network elements that are trusted. However, in an Internet, anyone can send a packet to any IP address. An area of significant concern is when attackers send packets to routers with the intent to disrupt service. This is an area where IP-based solutions, like MPLS, have some significant security weaknesses; at the time of writing, this was an area where major efforts were in progress to make routers less vulnerable to such attacks. Another area that must be secured is that of management access to the network elements themselves. This is important in all networks, and it applies to ATM as well, where the control protocols in the switches are not accessible to would-be attackers. However, management systems for ATM as well as MPLS-based networks are often TCP/IP based, and hence the problem of securing the network in these cases centers on securing access to the management network and not the network elements themselves.

REVIEW

This chapter looked at MPLS and ATM from economic, performance, and business points of view. The principal focus was an objective analysis of the efficiency versus performance questions involved in transferring TDM circuit, voice, packet data, and video information. As described in the text, the principal economic impact of efficiency occurs in access and expensive backbone circuits and ports. The fundamental difference in efficiency between different approaches is a consequence of the use of fixed-length cells in ATM versus use of variable-length packets in all other approaches. However, different protocols have different amounts of overhead, and we showed that the inefficiency tax of ATM is comparable to that of gigabit Ethernet. When efficiency is paramount, MPLS is the clear winner for a multi-service network where the majority of traffic is IP. On the other hand, when high-quality support for voice and circuit emulation is a major driver, ATM is currently the best solution available, but standardization and development are in process that may make MPLS the winner for this need as well.

The coverage then moved on to other design considerations. Scalability begins with the design and administration of addressing and routing. We discussed how connectionless routing scales better than connection-oriented protocols for the typical traffic profile. However, there is a fundamental scalability issue in that Internet traffic is growing faster than that supported by the sustained improvements in electronics predicted by Moore's law. In response to this challenge, new designs and architectures are necessary, as discussed in the next chapter.

We then discussed some aspects of the trade-off between complexity and function. This includes considerations regarding application requirements where the complexity of connection-oriented protocols with QoS and bandwidth reservation is justified, along with a discussion of some competing proposals. ATM and MPLS switches and routers have complex hardware, but even more complex software, and this presents a significant challenge to vendors and network operators in trading off complexity versus function versus reliability and availability. In order to achieve higher availability, redundancy is often applied, and this, too, is an area of future extensions being designed for MPLS, as discussed in the next chapter.

Next, we briefly discussed other design considerations related to stability resulting from equipment errors, human error, or complex interactions of routing protocols in large networks. The discussion continued with a review of important support and operational considerations. Finally, the chapter mentioned several important security design considerations, and what needs to be done in MPLS and ATM networks to address security issues. Now, let's gaze into the potential future of MPLS and ATM in the final chapter.



CHAPTER 30

Future Directions and Applications Involving MPLS and ATM

This final chapter presents some speculation regarding the future as a basis for a discussion regarding the potential application of selected technology directions involving MPLS and ATM. First, the future for ATM appears to be narrowing due to increased adoption of MPLS and IP. The principal future applications for ATM will likely be continued use in multiservice provider backbones and use in integrated access scenarios, such as digital subscriber line (DSL). The potential future uses of MPLS appear to be much broader. For certain, it will continue to be used as traffic engineering in Internet backbone networks, which will likely evolve with the extension to generalized MPLS (GMPLS) to support control of technologies other than packet switching (e.g., optical switches and multiplexers) to continue to scale core networks well into the future. Also, future network architectures may make the separation of control and forwarding a physical one instead of the current logical interpretation as part of achieving this vision. Finally, we ask you to suspend disbelief and invite you to think about what an MPLS- and IP-based multiservice network that carried all data, voice, video, and multimedia traffic would look like ten years from now.

FUTURE DIRECTIONS AND APPLICATIONS OF ATM

This section discusses the multiservice backbone and access infrastructure applications of ATM that are expected to see continued use in service provider networks.

Multiservice Backbone Network Infrastructure

Sometimes it is more expensive to migrate a network that is no longer rapidly growing to a new technology than it is to continue incrementally growing an existing network using legacy technology. There are several reasons why this may be true. First, moving to a new infrastructure reduces the useful life of previously purchased equipment. Second, the cost of the new equipment looms, as do installation, migration, and decommissioning costs. Also, there is often a need to build parallel networks in such a migration and there is more risk of destabilizing the network whenever change of such a scale is made. Finally, the operations staff must be trained on the new technology.

ATM will likely play some role in many service provider networks as infrastructure for Frame Relay service as well as a means to provide native ATM service [Nolle 00, Krapf 01]. Some service providers have used AAL1-based circuit emulation and/or AAL2 for toll-grade voice quality trunking over ATM. For the reasons stated, these service providers may have incentives to continue with this infrastructure. However, many service providers will migrate to MPLS for their Internet backbone as their network scales to a number of nodes where the full mesh of ATM PVCs could impact the stability of the IP interior routing protocol.

Several service providers have offered standards-based ATM switched virtual connection (SVC) services. As studied in Part 4, several important higher-layer applications running over ATM require SVCs to avoid the complexity of manual configuration and

perform optimally. These include classical IP over ATM and LAN Emulation (LANE). Another important use of SVCs enables a user to “dial up” bandwidth on demand, effectively provisioning a virtual private line in seconds.

Convergence and Integrated Access

In the latter part of the 1990s, ATM showed great potential for integrating voice, data, and video on a common access line for residential and even commercial customers. However, to date there has not been a wide-scale rollout of ATM for convergence of voice and data on access. One part of the reason that this has not happened is business, and the other is regulatory. In order for a business case to make sense, the equipment must be inexpensive and easy to maintain. Here, there is a chicken and egg situation—equipment is cheaper if many units are sold, but providers and customers will buy equipment only if it is cheap. In late 2000, the cost of such an integrated access device for a DS1 was over \$5000 [Mier 00]. As further evidence of high cost, Sprint pulled the plug on its version of converged access using ATM in late 2001 due to excessive costs [Cope 01]. Another inhibitor is regulatory. If the incumbent service provider that owns the access network must sell unbundled services to competitors, then there is less motivation to sell bundled services.

However, ATM is still very much alive in the DSL space as a means to provide converged voice and Internet access services [Krapf 01]. As discussed in Chapter 11, DSL competes with cable modems for broadband access to the residence or small business. Both technologies are optimized for high-speed transmission rates from the service provider toward the user. And consequently, the bandwidth from the user to the network can be quite limited and must be carefully managed. The packet voice standards summarized in Chapter 16 enable voice over ATM or IP to share an expensive access facility with Internet data. When ATM is used as the multiplexing technology, it can provide guaranteed QoS and reserved bandwidth for the voice cells. On lower-speed DSL uplinks, ATM is still a solid solution for meeting these needs, if the business and regulatory issues are addressed.

LESSONS LEARNED FROM ATM FOR MULTISERVICE NETWORKING

In large part, MPLS and IP have adopted ATM’s best features [McQuillan 97], eliminating the need for ATM in end systems. As mentioned at the beginning of this part, this will be the third attempt at networking to the Utopian goal of support for multiple services, the first two being narrowband ISDN and then ATM. This section briefly highlights some lessons learned from the past being applied in MPLS- and IP-based multiservice protocol and network design.

Don’t Operate at the Per-Flow Level

As discussed in Chapters 18 and 19, ATM SVCs strove to bridge the gap between the fundamentally different paradigms of IP and LANs versus ATM. In summary, IP and LANs

use connectionless packet-by-packet processing as opposed to ATM's connection-oriented protocol that switches once for an entire flow of packets communicated between endpoints. When switching was more cost-effective than packet forwarding, there was some motivation to do this. However, the ability of routers to cost-effectively implement packet forwarding at high speeds for native IP datagrams reduced the motivation for switching. Additionally, keeping state about each flow does not scale well at all due to the messaging required.

Furthermore, the ATM SVC protocol is quite complex and, in addition to supporting new features, also carries along with it the evolutionary baggage of interworking with telephony protocols conceived over a century ago. Also, emulating connectionless protocols with various forms of registration, address resolution, flow recognition, and fast connection setup is quite complex and in the end has not been economically justified. A similar lesson was learned from the commercial failure of RSVP for per-flow IP reservations. The result was that IP standards focused on QoS and bandwidth reservation at an aggregate level using diffserv, which is the approach used by MPLS, as discussed next.

Use Basic QoS and Traffic Management on Aggregates

Achieving useful QoS and traffic management for packet-switched communications has been a challenging research and standards activity for almost thirty years. There is a long string of failures, beginning with IP precedence in the late 1970s, which was too simple, continuing with RSVP in the 1990s, which turned out to be too complex, and most recently the differentiated services (diffserv) standards. On a parallel track, X.25 used a rather complex flow control for traffic control, and then Frame Relay introduced the notion of rate-based policing, followed by ATM, which precisely defined QoS and the notion of a traffic contract. ATM also defined a number of traffic and QoS functions that have not yet seen wide commercial use. Because MPLS was the latest approach, MPLS designers had the benefit of hindsight and chose the best subset from the concepts pioneered in ATM, RSVP, and diffserv. This approach may still not have all the functions necessary to serve a multiservice network, but it is often easier to add a feature than to remove one from a protocol.

Use Bandwidth Reservation for Constraint-Based Routing

The traffic engineering functions of ATM were embraced and refined in MPLS [RFC 2702]. Constraint-based routing affords an operator much more control over his or her network than a single routing metric can. And one of the most important constraints is that of bandwidth. Here, MPLS chose only a subset of the complex set of functions defined for ATM, but it is one that may be just enough to get the job done.

Assume a Heterogeneous Underlying Network

ATM assumed a homogeneous network environment, but IP works well in a heterogeneous environment, since it was designed that way from the outset. MPLS made the assumption that the underlying network could be made up of any link layer technology, even ATM. And as we shall see, starting with this principle from the outset may enable

extension of MPLS to operate over protocols running below the link layer as well. MPLS does not make the assumption that it is ubiquitous; however, at the time of writing, it does make the assumption that IP is ubiquitous [RFC 3031].

FUTURE APPLICATIONS AND DIRECTIONS OF MPLS (AND IP)

This section discusses some important applications and future directions in the area of supporting multiple services over MPLS (or IP).

Next Generation Multiservice Network Infrastructure

IP and MPLS may well realize the potential of the multiservice vision that ATM sought but failed to attain because it was not optimized for the dominant Internet application traffic profile. As described in Part 4, MPLS is optimized for IP and is going down a trail similar to that blazed by the ATM AAL of defining adaptation protocols to support multiple services in a well-defined layered fashion. There we described the plans that standards bodies have for MPLS and IP to support essentially the same set of services that were envisioned in the multiservice over ATM vision embarked upon by the ITU-T in the late 1980s. Although some areas of this work are further along than others, there are many parallels but the MPLS and IP work seems at least to be starting out trying to solve a simpler problem in a more layered approach, as contrasted with the more monolithic B-ISDN approach, where everything must run over ATM.

What may be different here for MPLS and IP as compared with ATM is that there is a larger talent pool of knowledgeable engineers, operators, and managers. Also, there was something of a religious war between IP and ATM that diffused energy that could have been otherwise applied to solving problems. Perhaps if the industry can focus on a common standard, the goal of universal multiservice networking may be achieved over MPLS and IP.

In addition to supporting the same set of services targeted by ATM adaptation layers and application-oriented protocols, MPLS is being defined as an enabler for new and different services. We also touched on some of these in Part 4. In particular, in Chapter 18 we discussed standards efforts and vendor proposals to implement virtual private LAN services over MPLS tunnels in metropolitan or wide area networks. In Chapter 19, we described the emerging set of network-based layer 3 VPN services that are based upon MPLS. These services may well supplant Frame Relay and ATM VPNs that are carrying only IP traffic. Finally, there are proposals for using MPLS technology to support optical VPNs.

Optical Networking for Scalability

As described in the last chapter, the price-performance history of electronics predicted by Moore's law cannot keep up with the growth rate of Internet traffic. There are several

parts to the answer. Optical networking is essential not only to achieve scalability, but also to reduce cost because although Internet traffic is growing at a rapid rate, Internet revenues are not growing nearly as fast [Varma 01, RHK 02]. What can be done to meet such a tidal wave of demand and reduce the cost of providing service at the same time? The first answer is recent developments in optical switching and multiplexing technology that we describe in this section. The second is extensions of MPLS-based protocols to control these new technologies at the core of network architecture that is highly scalable, as described in the next section.

In response to the tremendous growth in Internet traffic, an unprecedented level of investment by startup as well as established companies has resulted in an entire zoo of optical components becoming available [Liu 01]. These components are not just an evolution or refinement of earlier technologies but in some cases offer whole new capabilities. Long-haul dense wavelength division multiplexing (DWDM) is taken as a given, and the challenges now are to provide technology that increases capacity while reducing cost, as well as provide reconfiguration and restoration. Components are being developed that can tune to one of a number of wavelengths, by contrast with the single-wavelength technology of the past. Also, significant advances are being made toward making 40 Gbps transmission extend over long distances and achieve significant economies of scale for networks that can use this much capacity. Furthermore, technologies that can increase the number of wavelengths per fiber further increase available capacity that can be turned up on existing fibers. At the point where many fibers intersect, there are switches operating at the optical fiber level. There are also wavelength specific optical add/drop multiplexers (OADMs) for delivering only part of the capacity of a fiber between points in a network. Finally, there are also advances in the technologies necessary to control and manage the increasing number of fibers and wavelengths that are part of a network. But first, there must be a way to map from these lowest-level optical signals to something that is usable by higher-layer protocols.

To address this issue, standards are being defined that will enable MPLS and IP to operate over a set of next-generation transmission, multiplexing, and management protocols with an aim of preserving the necessary features of SONET/SDH, but simplifying the protocol in the process. Figure 30-1 presents an overview of some of the important protocols involved in this area [Bonenfant 01]. At the top are the IP and MPLS protocols. On the right is the familiar PPP over HDLC stack, also known as packet over SONET (POS). All other encapsulations use the IEEE 802.2 logical link control (LLC). On the left is the familiar LLC over ATM AAL5, commonly known as IP over ATM. In the middle is the familiar Ethernet medium access control (MAC) studied in Chapter 9. A newcomer here is the resilient pack ring (RPR) being defined in IEEE 802.17 working group. RPR is optimized for high-speed resilient ring packet transmission, which highlights use of statistical multiplexing, spatial reuse for more efficient ring capacity use through stripping off packets at the destination, and a protection scheme that does not reserve capacity in advance, unlike many other protection schemes. Underneath the Ethernet MAC there are 10 gigabit Ethernet (GbE) LAN and WAN physical layers, with the WAN PHY capable of being adapted to the SONET STS-192c (or SDH STM-64c) transmission system format.

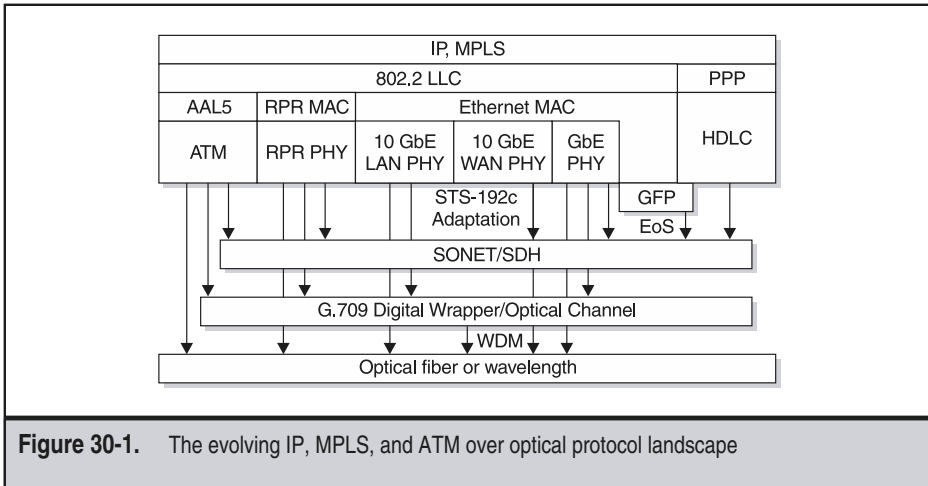


Figure 30-1. The evolving IP, MPLS, and ATM over optical protocol landscape

Moving further down the chart, there are some new protocols, and a number of possible mappings, which can be represented using arrows instead of stacked layers, as was done in [Bonenfant 01]. The generic framing procedure (GFP) defined in ITU-T Recommendation G.704 and ANSI can map framed data, like Ethernet MAC or HDLC, to SONET/SDH or an optical channel. GFP leverages another SONET/SDH standard called virtual concatenation that supports a more flexible set of rates than that supported by the traditional SONET/SDH hierarchy described in Chapter 6. ITU-T Recommendation G.709 defines a digital wrapper that supports the operation and management of a generic digital optical channel. This removes any dependence on SONET/SDH and should help to reduce costs going forward. It is envisioned that opto-electronic devices at the edges of an optically transparent network segment would implement this function. Finally, at the bottom of the figure is an optical fiber, or a wavelength on an optical fiber. Multiple wavelengths may be multiplexed onto a single fiber using WDM.

The many possible relationships between the transmission protocols shown in Figure 30-1 present a significant challenge for network designers and operators. Since a solution is not complete without control protocols, the ITU-T and the IETF have been busily at work defining solutions to this problem. The ITU-T was looking at extensions to ATM PNNI for this purpose, as well as joining with the IETF to investigate extensions to MPLS to solve the problem. We summarize the joint IETF and ITU-T efforts in the next section.

Generalized MPLS (GMPLS)

The IETF Common Control and Measurement Plane (CCAMP) working group was defining a set of standards called generalized MPLS (GMPLS) for defining signaling, routing, and measurement protocols for the purpose of supporting multiple switching

technologies, implementing topology discovery, and facilitating faster and more efficient restoration [GMPLS ARCH]. Although at the time of writing, the group had not yet produced any approved RFCs, much of the work was nearing completion. We give a high-level summary here and a motivational example. See the CCAMP working group page for more details.

In order to generalize MPLS, several things are necessary. First, the notion of what a label is must be generalized when dealing with such a broad range of technologies, since direct switching of TDM time slots, wavelengths, and optical fibers cannot use the MPLS shim header described in RFC 3032. Second, most transmission technologies cannot transfer the signaling and routing messages on which MPLS relies, and therefore a means to carry this information out of band and correlate it with the transmission system being controlled is necessary. Finally, there is also a need to extend routing and signaling protocols to include attributes that are unique to TDM, wavelength, and optical transmission systems.

GMPLS has several significant advantages [Banerjee 01]. First, use of a common control plane for routers and transmission equipment should simplify operation and management of network infrastructure and therefore reduce operational costs. Second, a common control plane avoids the cost of specifying and developing similar functions in multiple technology-specific protocols. Furthermore, GMPLS is structured to support several deployment scenarios, ranging from the most complex and full-featured peer model to a simpler subset called the overlay model. The peer model is similar to the NNI concept first defined by ATM where both signaling and routing are used, while the overlay model is similar to the early networks of IP routers overlaid on top of an ATM network. The overlay model may also support a UNI interface where devices exchange signaling messages, but not routing information.

GMPLS is defined as a set of extensions to existing routing and signaling protocols, along with one new link management protocol. The interior routing protocols, OSPF and IS-IS, are augmented to add information on optical and TDM networks, such as wavelength assignments, diversity specifications, and optical parameters. Signaling protocols, RSVP-TE and CR-LDP, are enhanced to allow establishment of bidirectional connections that may be made up of TDM time slots, wavelengths, or optical fibers. Finally, a new link management protocol (LMP) runs between neighboring nodes when the control plane traffic is carried out of band from the data plane [Zhang 01]. LMP is useful in link provisioning and fault isolation and has no interaction with the interior routing protocol.

GMPLS supports the concept of establishment of a hierarchical LSP that can span multiple technologies [Banerjee 01]. Let's see how this works with reference to the example of Figure 30-2, which shows a network composed of a set of devices with a hierarchically nested set of forwarding capabilities. At the lowest level are conventional label switching routers (LSRs), followed then by TDM digital cross-connects (DXCs), then wavelength division multiplexers (WDM), and finally at the highest level, photonic cross-connects (PXC). As one goes up the hierarchy, the overall bit rate increases and different types of labels are used, as shown in the left-hand side of the figure at the interfaces between the devices. In GMPLS hierarchical LSPs, new lower-level LSPs can be tunneled inside an existing higher-level LSP, which becomes an abstract node along the path of the

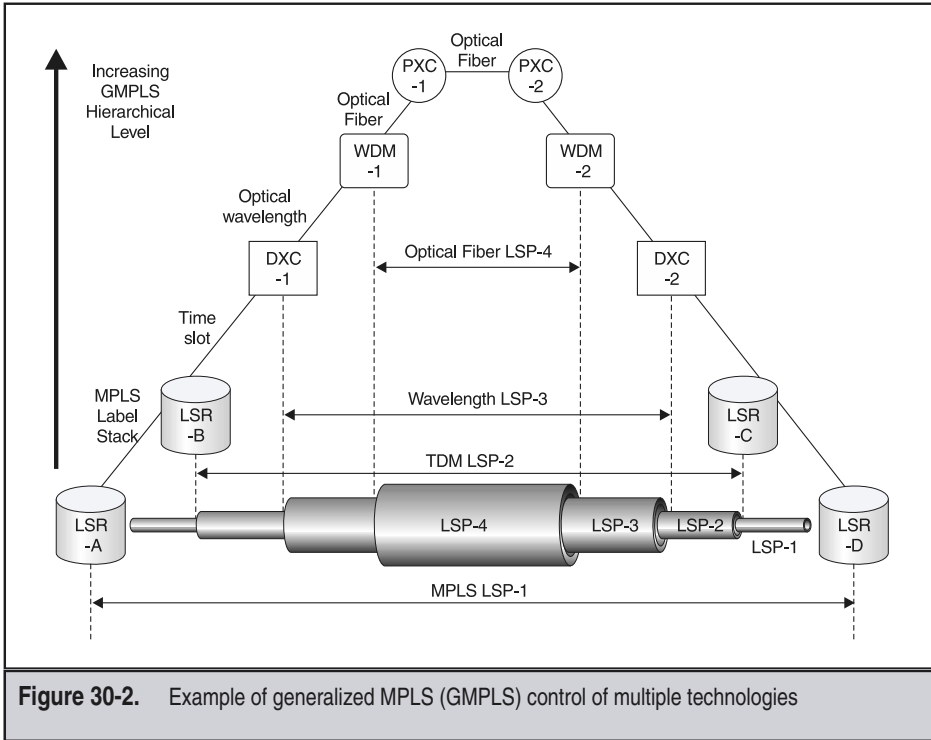


Figure 30-2. Example of generalized MPLS (GMPLS) control of multiple technologies

new LSP. A higher-level LSP may be established if no LSP at that level exists, or if an existing higher-level LSP does not meet a constraint of the new LSP (e.g., diversity). Thus, a request to establish a new lower-level LSP may trigger establishment of a higher-level LSP.

Let's look at how this works in our example, starting at the bottom of the figure for LSP-1 of capacity 100 Mbps to be established between LSR-A and LSR-D. Using the MPLS protocols described in Chapter 14, LSR-A decides that the route through LSR-B and LSR-C is the best path to the destination LSR-D, and adds an MPLS label stack to the packet obtained from a label distribution protocol. At the next level up in the GMPLS hierarchy, LSR-B determines that there is an existing 600 Mbps STS-12c time slot LSP-2 that reaches LSR-C, and that at least 100 Mbps of capacity is available. The TDM DXC-1 grooms this STS-12c into an OC-192 on specific wavelength, which has destination DXC-2 (usually on the same wavelength) that is demultiplexed to an STS-12c connecting to LSR-C. The wavelength multiplexers/switches WDM-1 and WDM-2 implement the wavelength LSP-3 over a concatenated set of optical fibers. Finally, at the top of the hierarchy the photonic switches PXC-1 and PXC-2 implement the optical fiber LSP-4 by switching all light on an entire fiber.

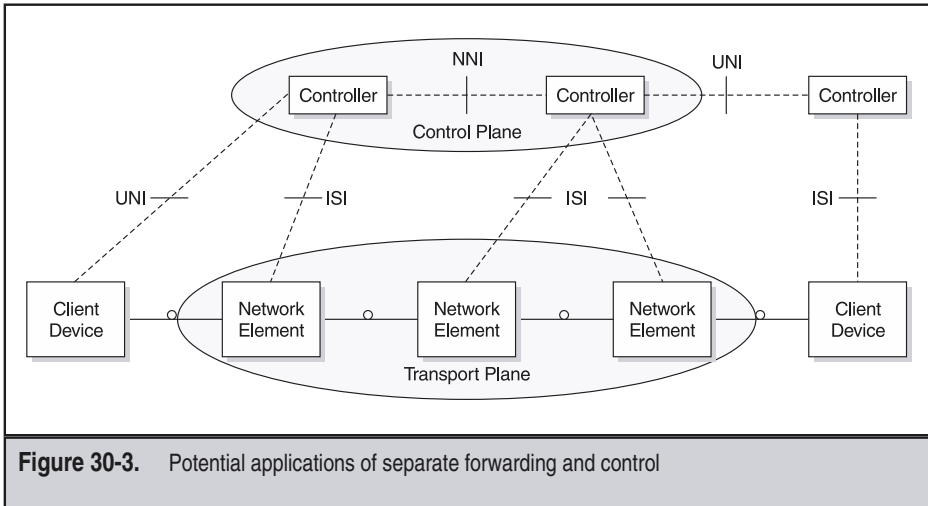
Pretty cool stuff, right? But do TDM SONET/SDH cross-connects, wavelength multiplexers, and photonic switches need have all the intelligence of an Internet router in order to achieve this vision? The answer is no; because implementing all of the IP and MPLS protocols is a complex task, the industry has been working on an approach that combines the best of breed technology in the control plane with the diverse technologies supported in the GMPLS forwarding plane.

Separation of Forwarding and Control

As covered in this book, communications protocols have employed the concept of a logical separation of control functions from user-oriented service functions several times; examples are N-ISDN, ATM, Ipsilon's IP switch, MPLS, and voice over packet. However, in some cases the separation is only logical, while in other cases the separation can be physical when a protocol is defined between the controller and a device. Such is the case with this protocol being the general switch management protocol (GSMP) [RFC 1987] for Ipsilon's IP switch and MEGACO [ITU H.248, RFC 2805] for voice over packet. The application of the same concept to control of SONET/SDH devices, wavelength multiplexers, and photonic cross-connects is a natural use of this concept, since traditionally this class of devices had nothing to do with Internet protocols.

Several industry groups and standards bodies have been working on specifying the interface between the controller and a transmission (usually called transport in ITU-T parlance) device, in addition to the interfaces between controllers and client devices that would use the automatic control of transport network elements. We briefly summarize the general direction of standards efforts and implementations of this architecture for control of non-IP technologies with reference to Figure 30-3 [Benjamin 01]. The principal functions are grouped into a transport plane composed of non-IP network elements like those discussed previously and a control plane composed of one or more controllers. Client devices can signal over a UNI interface to a controller in the control plane to request services, for example, using the Optical Interworking Forum (OIF) UNI 1.0 specification [OIF UNI 1.0]. Controllers communicate using an NNI protocol, for example, GMPLS. As shown in the center of the figure, the controller may be separate from the network element, and one controller may support more than one network element. An intrasystem interface (ISI) is a standard protocol between a controller and a network element; an example of an ISI comprises extensions to the GSMP protocol that add support for control of TDM and optical devices [GSMP OPTICAL]. This separation of forwarding and control may also exist for a client device, for example an MPLS LSR, where the ISI could be version 3 of GSMP that supports configuration of a label-switched cross-connect [RFC 3292].

This design is still a work in progress. For further information, see the automatic switched transport network (ASTN) framework in ITU-T Recommendation G.807/Y.1302, the architecture for the automatically switched optical network (ASON) in ITU-T Recommendation G.8080/Y.1302, and the latest information in the IETF GSMP working group and the OIF.



POSSIBLE FUTURE OF MULTISERVICE NETWORKING

This section presents some speculation on what future drivers for demand could be, as well as what the fate of MPLS, GMPLS, and ATM could be.

What Will Continue the Internet's Explosive Growth?

Internet traffic continues growing at a phenomenal rate. Much of the traffic is currently Web traffic, and most of that TCP. What will the traffic mix of the future be? What will the impact of broadband access eventually be? What new hardware and software capabilities will affordable computers and information appliances have? What applications can sustain the tremendous growth in data traffic? How quickly can a networking need change when people replace their computing devices once every several years? Let's look at some possible scenarios as an answer to these questions.

Currently, all of the telephone networks in the United States combined carry a peak of approximately 20 million conversations during the business busy hour and during peak periods, like Mother's Day. At 64 Kbps, this level of traffic is only about 1 terabit per second (i.e., 10^{12} bps), which we abbreviate as 1 Tbps. Already, hundreds of millions of people around the world spend several hours a day surfing the Web, reading e-mail, participating in chat groups, or following other pursuits. At an average transfer rate of 20 Kbps for 50 million online users, this equates to a residential demand of approximately 1 Tbps. During the business day, assume that the approximately 1 million enterprise sites generate an average of 1 Mbps of traffic each, again approximately 1 Tbps. The commercial and residential busy hours will likely be noncoincident. Therefore, the total domestic

U.S. voice and data network needs to carry on the order of several Tbps. Assuming that Internet traffic doubles every year, the total demand in ten years would be on the order of 1 petabit per second (i.e., 10^{15} bps). Where could all this demand come from? The answer may be quite literally right in front of our faces.

Statistics show that Americans spend several hours a day watching television, more than any other nation (not that this is something to be proud of). The cable system typically carries up 100 channels of approximately 3 MHz each to 50 million households in the United States. Assuming 1 bps per Hz, this equates to approximately 15 Pbps. When broadband DSL, cable, and fiber access deliver video on demand services, a natural conclusion is that the channel changers will spend at least some of their time away from the television. Assume that one-quarter of these households in the United States use video on demand at 1 Mbps during periods of peak demand. This represents an aggregate demand of approximately 10 Tbps. This is ten times the current level of Internet traffic and could be a significant source of future traffic growth, especially if the access capacity bottleneck is resolved.

Therefore, ten years from now the potential data and video traffic will be several orders of magnitude greater than voice traffic. Eventually, technology and demand will eclipse even the benchmark of a petabit per second and we will measure network capacity in terms of exabits per second (10^{18} bps).

Will MPLS Become the Ubiquitous Multiservice Network?

Static, predefined private communications networks are migrating to dynamic virtual private networks (VPNs), which are implemented over public infrastructures employing sophisticated switching and routing intelligence, relieving the enterprise of much of the network management burden. Major corporations increasingly depend upon virtual networking to run their day-to-day businesses, relying on their partitioning and security features. Here MPLS and IP are being increasingly used to achieve these goals.

While today it makes sense to keep voice and data networks separate, network architects must at least consider whether it makes economic and operational sense to merge the two. However, the wholesale replacement of the current PSTN infrastructure may never occur, because there is no compelling reason to replace it. Few would argue that the telephone network is terribly broken and desperately needs fixing.

But in the end, will it be an MPLS network, or an IP network that delivers on these promises? Didn't the ITU-T have a similar goal in assuming that ATM would be everywhere in its multiservice vision? What is different with MPLS? Despite some vendor and trade press claims that MPLS is the answer to everything, there is evidence to the contrary [McDysan 01]. First, it is quite unusual for a networking protocol to become ubiquitous. As of the time of writing, only two protocols had achieved this status: voice and IP. Second, the IETF PPVPN and PWE3 working groups have required that multiple services be supported not only over MPLS, but over IP as well. These standards groups are explicitly making the assumption that MPLS will not be ubiquitous. Finally, service providers need not deploy MPLS ubiquitously even within their own networks, and even if they did, they would have some strong motivations to decline to fully interconnect with other service

providers using MPLS. Within certain network contexts, particularly smaller (sub)networks, MPLS may well be an attractive infrastructure, but it appears unlikely that MPLS will ever become as ubiquitous as voice or IP.

Will GMPLS Effectively Control Next-Generation Backbones?

Service providers must deploy next-generation optical technologies to move to a next-generation architecture to keep up with the sustained Internet traffic growth rate, which is in excess of the increase in speed and capacity achievable with the current architecture as predicted by Moore's law. However, deployment of a technology without a corresponding set of control protocols can result in significant management system and operational expenses, as occurred for TDM networks. Much work has gone into extending MPLS to try to meet these needs. At the time of writing, these types of approaches had yet to find sufficient large-scale deployment to determine how well they would work. But it is promising that work has begun on the problem before it can become an operational issue.

What Will Happen to ATM?

In the LAN, Ethernet derivatives are the clear winner, with ATM LAN emulation relegated to support of specialized applications. In the WAN, Internet service provider backbones will be a mix of packet over optical, SONET, MPLS, and ATM, depending upon what other traffic mix the network carries. In Frame Relay and native ATM service networks, ATM will still be widely used, but some parts of these networks may be trunked over MPLS (or even IP) as service providers introduce new backbone network technologies. Some integrated access and QoS-aware private and virtual private networks will use ATM, but many will use MPLS or IP. ATM may also play an important role in DSL access and aggregation networks, as well as providing for high-quality circuit emulation and voice trunking. However, as we have described, in the long run many of the best parts of ATM have already been adopted in MPLS standards and implementations, and in that sense, the lessons learned from ATM will likely live on for much longer. We hope that this book has helped you learn some of these lessons and will help you avoid mistakes of the past and focus on techniques and approaches that have withstood the test of time.

REVIEW

This final chapter discussed some potential directions that ATM and MPLS could take. It then addressed the likely future roles for ATM as the infrastructure for integrated access and a backbone network for Frame Relay and native ATM, and potential use for circuit emulation and voice trunking. We then summarized the lessons learned from ATM, and how these have been applied in MPLS. Next, the chapter focused on several potential directions for MPLS. Two were already described in Part 4, namely that of multiservice support analogous to that of ATM and also enabling new types of services, like Ethernet MAN/WAN services and layer 2 and layer 3 network-based VPNs. We then focused on

the new set of optical technologies being developed and how IP and MPLS could operate over them. In particular, a generalized MPLS (GMPLS) protocol could control not only IP-based devices, but also traditional transmission devices, including TDM cross-connects, wavelength multiplexers, and optical switches. Toward this end, we also summarized how a physical separation of control and transport planes could make these benefits available more cost-effectively. Finally, the chapter concluded with further speculations on the architecture of tomorrow's networks.

One thing is for certain: the telecommunications world is ever changing. Those that aren't ready to change and adapt will soon become extinct. We hope that this book gave you background in the data communication technologies and analytical methods to better prepare you in your own battle for survival.

APPENDIX A



Acronyms and Abbreviations

ACRONYMS AND ABBREVIATIONS

Abbreviation	Term
μ s	microsecond (10^{-6} seconds)
AAL	ATM Adaptation Layer
ABR	Available Bit Rate (ATM Forum)
ACK	Acknowledgment
ACR	Allowed Cell Rate (ATM ABR)
ADM	Add/Drop Multiplexer
ADPCM	Adaptive Differential Pulse Code Modulation
ADSL	Asymmetric Digital Subscriber Line
ADTF	ACR Decrease Time Factor (ATM ABR)
AESA	ATM End System Address
AF	Assured Forwarding (Diffserv)
AFI	Authority and Format Identifier (AESA)
AINI	ATM Inter-Network Interface
AIR	Additive Increase Rate (ATM ABR)
AIS	Alarm Indication Signal (OAM)
ANS	ATM Name System
ANSI	American National Standards Institute
API	Application Programming Interface
APS	Automatic Protection Switching
ARIS	Aggregate Route-Based IP Switching
ARP	Address Resolution Protocol
AS	Autonomous System (IETF)
ASIC	Application-Specific Integrated Circuit
ASN.1	Abstract Syntax Notation One
ASON	Automatically Switched Optical Network (ITU-T)
ASTN	Automatically Switched Transport Network (ITU-T)
ATC	ATM Transfer Capability (ITU-T)
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
ATMF	ATM Forum
AUI	Attachment Unit Interface (Ethernet 802.3)

Abbreviation	Term
B	Bearer (64 Kbps DS0 Channel in N-ISDN)
BA	Behavior Aggregate (Diffserv)
BCD	Binary-Coded Decimal
BCS	Behavior Class Selector (ATM Forum)
BDI	Backward Defect Indication (ITU-T)
BECN	Backward Explicit Congestion Notification (FR)
Bellcore	Bell Communications Research (Renamed to Telcordia)
BER	Bit Error Ratio or Rate
BGP	Border Gateway Protocol (IETF)
B-HLI	Broadband High-Layer Information
BICC	Bearer Independent Call Control
B-ICI	Broadband Intercarrier Interface
BIP	Bit Interleaved Parity
B-ISDN	Broadband Integrated Services Digital Network (ITU-T)
B-ISUP	Broadband ISDN User Part (ITU-T)
BITS	Building Integrated Timing Supply
B-LLI	Broadband Low-Layer Information
B-NT	Broadband Network Termination
BOC	Bell Operating Company
bps	bits per second
BR	Backward Reporting (ATM OAM)
BRI	Basic Rate Interface (ISDN)
BT	Burst Tolerance (ATM Forum)
B-TA	Broadband Terminal Adapter (ATM)
B-TE	Broadband Terminal Equipment (ATM)
BUS	Broadcast Unknown Server (ATM LANE)
C/R	Command/Response Indicator or Bit
CAC	Connection Admission Control
CAS	Channel Associated Signaling
CBR	Constant Bit Rate (ATM Forum)
CC	Continuity Check (ATM OAM)
CBS	Committed Burst Size
CCAMP	Common Control and Management Plane
CCITT	Consultative Committee on International Telephone and Telegraph (now ITU)

Abbreviation	Term
CCR	Current Cell Rate (ATM ABR)
CCS	Common Channel Signaling
CCS7	Common Channel Signaling System 7
CDF	Cutoff Decrease Factor (ATM ABR)
CDV	Cell Delay Variation (ATM QoS)
CDVT	Cell Delay Variation Tolerance
CE	Customer Edge (Device)
CER	Cell Error Ratio (ATM QoS)
CES	Circuit Emulation Service
CI	Congestion Indicator (ATM ABR)
CID	Channel ID (AAL2, Voice over MPLS)
CIDR	Classless Inter-Domain Routing (IETF)
CIR	Committed Information Rate (FR, Diffserv)
CLLM	Consolidated Link Layer Management (FR)
CLNP	Connectionless Layer Network Protocol (ITU)
CLNS	Connectionless Network Service (OSI)
CLP	Cell Loss Priority (ATM QoS)
CLR	Cell Loss Ratio (ATM QoS)
CMIP	Common Management Interface Protocol (ISO)
CMR	Cell Misinsertion Rate (ATM QoS)
CO	Central Office
Codec	Coder/Decoder
CONS	Connection-Oriented Network Service (ISO)
CPCS	Common Part Convergence Sublayer
CPE	Customer Premises Equipment
CPI	Common Part Indicator
CPS	Common Part Sublayer (AAL2)
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRF	Connection Related Function (ATM)
CR-LDP	Constraint-Based Routing Label Distribution Protocol (MPLS)
CRS	Cell Relay Service
CS	Convergence Sublayer (ATM AAL)
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection

Abbreviation	Term
CSU/DSU	Channel Service Unit/Data Service Unit
CTD	Cell Transfer Delay (ATM QoS)
CV	Continuity Verification (ITU-T MPLS OAM)
DA	Destination Address field
DAL	Dedicated Access Line
DARPA	Defense Advanced Research Projects Agency
DBR	Domain Based Rerouting (ATM Forum)
DBR	Deterministic Bit Rate (ITU-T ATM)
DCC	Data Country Code (AESAs)
DCE	Data Communications Equipment
DCS	Digital Cross-connect System
DE	Discard Eligible (FR)
DHCP	Dynamic Host Configuration Protocol (IETF)
Diffserv	Differentiated Service (IETF)
DLCI	Data Link Connection Identifier (FR)
DMA	Direct Memory Access
DNS	Domain Name Service (IETF)
DoD	Department Code Listing
DQDB	Distributed Queue Dual Bus (SMDS)
DS0	Digital Signal Level 0—64 Kbps
DS1	Digital Signal Level 1—1.544 Mbps
DS3	Digital Signal Level 3—44.76 Mbps
DSAP	Destination Service Access Point (LLC)
DSCP	Diffserv Code Point (IETF)
DSL	Digital Subscriber Line
DSP	Domain Specific Part (AESAs)
DSR	Data Set Ready
DSS	Digital Subscriber Signaling System (1 is N-ISDN, 2 is B-ISDN)
DSU	Data Service Unit
DSX	Digital Signal Cross-Connect
DTE	Data Terminal Equipment
DTL	Designated Transit List (ATM PNNI)
DTMF	Dual Tone Multifrequency
DWDM	Dense Wave Division Multiplexing

Abbreviation	Term
DXC	Digital cross (X)-Connect
DXI	Data Exchange Interface (SMDS, ATM)
E&M	Ear & Mouth or Earth & Magnet
E1	European Transmission Level 1—2 Mbps
E3	European Transmission Level 3—34 Mbps
EA	Address Field Extension (FR)
eBGP	exterior Border Gateway Protocol (IETF)
EBS	Excess Burst Size (Diffserv)
ECN	Explicit Congestion Notification (TCP/IP)
EF	Expedited Forwarding (Diffserv)
EFCI	Explicit Forward Congestion Indication (ATM)
EFS	Error Free Seconds
EGP	External (Exterior) Gateway Protocol
EIA	Electronics Industries Association
EIR	Excess Information Rate (FR)
ELAN	Emulated LAN (ATM LANE)
E-LSP	EXP field-based Label Switched Path (MPLS)
EML	Element Management Layer
EMS	Element Management System
EOM	End of Message
EPD	Early Packet Discard (ATM)
EPRCA	Enhanced Proportional Rate Control Algorithm (ATM ABR)
ER	Explicit Rate
ERO	Explicit Route Object (MPLS)
ES	End System (OSI) or Errored Seconds
ESF	Extended Superframe
ESI	End System Identifier (AESA)
ETSI	European Telecommunications Standards Institute
EXP	Experimental field (MPLS)
FAST	Frame-based ATM over SONET/SDH (ATM Forum)
FCAPS	Fault, Configuration, Accounting, Provisioning, and Security Management (ISO, ITU-T)
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface

Abbreviation	Term
FDM	Frequency-Division Multiplexing
FEBE	Far End Block Error (SONET)
FEC	Forward Error Correction
FEC	Forwarding Equivalence Class (MPLS)
FECN	Forward Explicit Congestion Notification (FR)
FERF	Far End Reporting Failure (now called RDI)
FF	Fixed Filter (RSVP reservation style)
FIB	Forwarding Information Base
FPM	Forward Performance Monitoring (ATM OAM)
FR	Frame Relay
FR-SSCS	Frame Relay Service Specific Convergence Sublayer
FT1	Fractional T1
FTN	FEC to NHLFE Map (MPLS)
FTP	File Transfer Protocol (IETF)
FUNI	Frame-based User-to-Network Interface (ATM Forum)
GAT	Generic Application Transport (ATM)
GbE	Gigabit Ethernet (IEEE)
Gbps	Gigabits per second (10^9 bps)
GCAC	Generic Connection Admission Control (ATM PNNI)
GCRA	Generic Cell Rate Algorithm (ATM)
GFC	Generic Flow Control (ATM)
GFP	Generic Framing Procedure (ITU-T)
GFR	Guaranteed Frame Rate (ATM Forum)
GMPLS	Generalized MPLS (IETF)
GPS	Generalized Processor Sharing
GSMP	Generic Switch Management Protocol (IETF)
GUI	Graphical User Interface
HDLC	High-Level Data Link Control (ISO)
HDTV	High-Definition Television
HEC	Header Error Control (ATM)
HSSI	High-Speed Serial Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol (IETF)
Hz	Hertz or cycles per second

Abbreviation	Term
IA	Implementation Agreement
IANA	Internet Assigned Numbers Authority
iBGP	interior Border Gateway Protocol (IETF)
ICD	International Code Designator (AESA)
ICMP	Internet Control Message Protocol (IETF)
ICR	Initial Cell Rate (ATM ABR)
IDI	Initial Domain Identifier (AESA)
IDP	Initial Domain Part (AESA)
IE	Information Element
IEC	Inter-Exchange Carrier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol (IETF)
IGP	Internal (Interior) Gateway Protocol
IISP	Interim Interswitch Signaling Protocol (ATM Forum)
ILM	Incoming Label Map (MPLS)
ILMI	Integrated Local Management Interface (ATM Forum)
IMA	Inverse Multiplexing over ATM
IME	ILMI Management Entity
InARP	Inverse ARP (ATM)
INE	Interworking Network Element
Intserv	Integrated Service (IETF)
ION	IP Over Non-broadcast multiple access networks (IETF)
IP	Internet Protocol (IETF)
IPsec	IP security (IETF)
IPX	Internetwork Packet Exchange protocol (Novell)
IS	Intermediate System (OSI)
ISDN	Integrated Services Digital Network (ITU-T)
ISDU	Isochronous Service Data Unit (DQDB)
IS-IS	Intermediate System-to-Intermediate System (OSI)
IS-IS TE	IS-IS with Traffic Engineering (MPLS)
ISO	International Standards Organization
ISP	Internet Service Provider
ITU	International Telecommunications Union

Abbreviation	Term
ITU-T	ITU—Telecommunications standardization sector
IWF	Inter-Working Function
IXC	IntereXchange Carrier
JPEG	Joint Photographic Experts Group
kbps	kilobits per second (10^3 bps)
km	kilometers (10^3 meters)
L2TP	Layer 2 Tunneling Protocol (IETF)
LAN	Local Area Network
LANE	LAN Emulation (ATM Forum)
LAP-B	Link Access Procedure—Balanced (X.25)
LAP-D	Link Access Procedure—D (ISDN/Frame Relay)
LAP-F	Link Access Procedure—Frame Mode
LATA	Local Access Transport Area
LB	Loopback (ATM OAM)
LCP	Link Control Protocol
LCT	Last Compliance Time (used in GCRA definition)
LDP	Label Distribution Protocol (MPLS)
LEC	Local Exchange Carrier
LEC	LAN Emulation Client (ATM Forum LANE)
LECS	LAN Emulation Configuration Server (ATM Forum LANE)
LER	Label Edge Router (MPLS)
LES	LAN Emulation Server (ATM Forum LANE)
LFIB	Label Forwarding Information Base (MPLS)
LGN	Logical Group Node (ATM Forum PNNI)
LIB	Label Information Base (MPLS)
LIJ	Leaf Initiated Join (ATM Forum)
LIS	Logical IP Subnet (IETF)
LLC	Logical Link Control (IEEE 802.2)
L-LSP	Label-based Label Switched Path (MPLS)
LMI	Local Management Interface
LMP	Link Management Protocol (GMPLS)
<i>L_n</i>	Layer “ <i>n</i> ” (e.g., L1 is Layer 1, L2 is Layer 2, L3 is Layer 3)
LNNI	LAN Emulation NNI (ATM Forum)
LOC	Loss of Cell Delineation (ATM)

Abbreviation	Term
LOF	Loss of Frame (SONET/SDH)
LOP	Loss of Pointer (SONET/SDH)
LOS	Loss of Signal
LSA	Link State Advertisement (OSPF)
LSB	Least Significant Bit
LSP	Label-Switched Path (MPLS)
LSP	Link State Packet (IS-IS)
LSR	Label Switching Router (MPLS)
LTE	Line Terminating Equipment (SONET)
LUNI	LANE UNI (ATM Forum, LANE)
m	Meter
MAC	Media Access Control (IEEE 802.X)
MAN	Metropolitan Area Network
MARS	Multicast Address Resolution Service (IETF)
Mbps	Megabits per second (10^6 bps)
MBS	Maximum Burst Size (ATM Forum)
MCR	Minimum Cell Rate (ATM ABR)
MCS	Multicast Server (IETF)
MD5	Message Digest Authentication version 5 (IETF)
MDCR	Minimum Desired Cell Rate (ATM Forum)
MEGACO	Media Gateway Control (VoIP, VoATM)
MG	Media Gateway
MGC	Media Gateway Controller
MHz	Megahertz
MIB	Management Information Base
MID	Multiplexing Identifier (ATM)
MIPS	Millions of Instructions per Second
MLPPP	Multi Link Point-to-Point Protocol (IETF)
MPEG	Motion Picture Experts Group
MPLS	Multiprotocol Label Switching (IETF)
MPOA	Multiprotocol over ATM (ATM Forum)
ms	millisecond (one-thousandth of a second, 10^{-3} second)
MSB	Most Significant Bit
MTU	Maximum Transfer Unit (IP, MPLS)

Abbreviation	Term
MUX	Multiplexer
NANP	North American Numbering Plan
NBMA	Non-Broadcast Multiple Access
NCCI	Network Call Correlation Identifier (ATM)
NE	Network Element
NHLFE	Next Hop Label Forwarding Entry (MPLS)
NHRP	Next-Hop Resolution Protocol (IETF)
NIC	Network Interface Card
N-ISDN	Narrowband Integrated Services Digital Network (ITU-T)
NLPID	Network-Layer Protocol Identifier
nm	nanometer (10^{-9} m)
NM	Network Management
NMS	Network Management System
NNI	Network Node Interface, or Network to Network Interface
NP	Network Performance
NPC	Network Parameter Control (ATM)
Nrm	Number of cells between RM cells (ATM ABR)
nrt-VBR	non-real time Variable Bit Rate (ATM Forum)
ns	nanosecond (10^{-9} second)
NSAP	Network Service Access Point
NT	Network Termination
OADM	Optical Add/Drop Multiplexer
OAM	Operations And Maintenance
OAM&P	Operations, Administration, Maintenance, and Provisioning
OC-n	Optical Carrier Level n (SONET)
OID	Object Identifier
OS	Operating Systems
OSI	Open Systems Interconnection
OSI CLNS	Connectionless Network System (OSI)
OSIRM	OSI Reference Model
OSPF	Open Shortest Path First (IETF)
OSPF-TE	Open Shortest Path First with Traffic Engineering (IETF)
OUI	Organizationally Unique Identifier (IEEE)
P	Provider device (IETF)

Abbreviation	Term
PABX	Private Automated Branch Exchange
PAD	Packet Assembler/Disassembler (X.25)
PBX	Private Branch Exchange
PC	Personal Computer
PCM	Pulse Code Modulation
PCR	Program Clock Reference (MPEG2)
PCR	Peak Cell Rate (ATM)
PDB	Per Domain Behavior (Diffserv)
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit
PE	Provider Edge device (IETF)
PG	Peer Group
PGL	Peer Group Leader (PNNI)
PGPS	Packetized Generalized Processor Sharing
PHB	Per Hop Behavior (Diffserv)
PHY	Physical Layer
PID	Protocol Identifier
PIM	Protocol Independent Multicast (IETF)
PING	Packet INternet Groper (IETF)
PIR	Peak Information Rate (Diffserv)
PLCP	Physical Layer Convergence Protocol (ATM)
PMD	Physical Medium Dependent (ATM)
PNNI	Private/Public Network-to-Network or Network-Node Interface (ATM Forum)
PoP	Point of Presence
POS	PPP(or Packet) over SONET
POTS	Plain Old Telephone Service
PPD	Partial Packet Discard
PPP	Point-to-Point Protocol (IETF)
pps	packets per second
PPVPN	Provider Provisioned VPN (IETF)
PRI	Primary Rate Interface (ISDN)
PSN	Packet Switched Network
PSTN	Public Switched Telephone Network
PT	Payload Type

Abbreviation	Term
PTE	Path-Terminating Equipment (SONET)
PTI	Payload Type Identifier
PTSE	PNNI Topology State Element
PTSP	PNNI Topology State Packet
PTT	Postal, Telegraph & Telephone Ministry / Administration
PVC	Permanent Virtual Connection (FR, ATM)
PVCC	Permanent Virtual Channel Connection (ATM)
PVPC	Permanent Virtual Path Connection (ATM)
PWE3	Pseudo Wire Emulation Edge to Edge (IETF)
QoS	Quality of Service
RAM	Random Access Memory
ROBOC	Regional Bell Operating Company
RDF	Rate Decrease Factor (ATM ABR)
RDI	Remote Defect Indication
RED	Random Early Detection (or Discard)
RFC	Request for Comment (IETF)
RIB	Routing Information Base
RIP	Routing Information Protocol (IETF)
RM	Resource Management (ABR)
RMON	Remote MONitoring (IETF)
ROM	Read-Only Memory
RPR	Resilient Packet Ring (IEEE)
RSVP	Resource reSerVation Protocol (IETF)
RTCP	RTP Control Protocol (TCP/IP)
RTP	Real-Time Transport Protocol (IETF)
RTT	Round-Trip Time
rt-VBR	real-time Variable Bit Rate (ATM Forum)
s	Second
SA	Source Address
SAAL	Signaling ATM Adaptation Layer
SAP	Service Access Point
SAR	Segmentation and Reassembly (ATM AAL)
SBR	Statistical Bit Rate (ITU-T)
SCR	Sustainable Cell Rate (ATM Forum)

Abbreviation	Term
SDH	Synchronous Digital Hierarchy (ITU-T)
SDLC	Synchronous Data Link Control protocol (IBM)
SDT	Structured Data Transfer (AAL1)
SDU	Service Data Unit
SE	Shared Explicit (RSVP reservation style)
SEAL	Simple and Efficient Adaptation Layer (AAL5)
SECB	Severely Errored Cell Block (ATM)
SEL	Selector Byte (AESA)
SES	Severely Errored Seconds
SIP	Session Initiation Protocol (IETF)
SLA	Service Level Agreement
SLIP	Serial Line IP (IETF)
SMDS	Switched Multimegabit Data Service
SMF	Single-Mode Fiber
SN	Sequence Number
SNA	Systems Network Architecture (IBM)
SNAP	Sub-Network Attachment Point
SNMP	Simple Network Management Protocol (IETF)
SONET	Synchronous Optical Network (ANSI)
SPE	Synchronous Payload Envelope (SONET)
SPVCC	Soft Permanent Virtual Channel Connection (PNNI)
SPVPC	Soft Permanent Virtual Path Connection (PNNI)
SREJ	Select Reject frame
SRTS	Synchronous Residual Time Stamp (AAL1)
SS7	Signaling System Number 7 (ITU-T)
SSAP	Source Service Access Point (LLC)
SSCF	Service-Specific Coordination Function (AAL)
SSCOP	Service-Specific Connection-Oriented Protocol (AAL)
SSCS	Service-Specific Convergence Sublayer (ATM)
STM	Synchronous Transfer Mode (SDH)
STM- <i>n</i>	Synchronous Transport Module level <i>n</i> (SDH)
STP	Shielded Twisted Pair
STP	Spanning Tree Protocol (IEEE 802.1d)
STS- <i>N</i>	Synchronous Transport Signal Level <i>N</i> (SONET)

Abbreviation	Term
STS-Nc	Concatenated Synchronous Transport Signal Level <i>N</i> (SONET)
SVC	Switched Virtual Connection (FR, ATM)
SVCC	Switched Virtual Channel Connection (ATM)
SVPC	Switched Virtual Path Connection (ATM)
TA	Terminal Adapter
TAT	Theoretical Arrival Time (used in GCRA definition)
TC	Transmission Convergence sublayer of PHY Layer
TCAP	Transaction Capabilities Applications Part (SS7)
TCP	Transmission Control Protocol (IETF)
TCP/IP	Transmission Control Protocol/Internet Protocol (IETF)
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TDS	Time Division Switching
TE	Traffic Engineering or Terminal Equipment
TFTP	Trivial File Transfer Protocol (IETF)
TINA-C	Telecom Information Networking Architecture Consortium
TLV	Type Length Value
TM	Traffic Management
TMN	Telecommunications Management Network (ITU-T)
TNS	Transit Network Selection
TOS	Type of Service (IETF)
TTL	Time-To-Live (IETF)
TUC	Total User Cell Count (ATM OAM)
TUCD	Total User Cell Difference (ATM OAM)
UBR	Unspecified Bit Rate (ATM Forum)
UDP	User Datagram Protocol (IETF)
UDT	Unstructured Data Transfer (AAL1)
UME	UNI Management Entity (used in ILMI Definition)
UNI	User-to-Network Interface
UPC	Usage Parameter Control (ATM)
URL	Uniform Resource Locator (IETF)
UTOPIA	Universal Test and Operations Interface for ATM
UTP	Unshielded Twisted Pair
UUI	User-to-User Indication

Abbreviation	Term
VBR	Variable Bit Rate (ATM Forum)
VC	Virtual Channel (ATM) or Virtual Call (X.25) or Virtual Connection
VCC	Virtual Channel Connection (ATM)
VCI	Virtual Channel Identifier (ATM)
VCL	Virtual Channel Link (ATM)
VC- <i>n</i>	Virtual Container- <i>n</i> (SDH)
VFI	Virtual Forwarding Instance
VFS	Virtual Forwarding and Switching
VLAN	Virtual LAN
VoATM	Voice over ATM
VoD	Video on Demand
VoIP	Voice over IP
VoMPLS	Voice over MPLS
VP	Virtual Path (ATM)
VPC	Virtual Path Connection (ATM)
VPCI	Virtual Path Connection Identifier (ATM)
VPI	Virtual Path Identifier(ATM)
VPL	Virtual Path Link (ATM)
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
VR	Virtual Router
VRF	Virtual Routing and Forwarding (Instance)
VS/VD	Virtual Source/Virtual Destination (ABR)
VSI	Virtual Switching Instance
VT	Virtual Tributary (SONET)
VTOA	Voice and Telephony over ATM (ATM Forum)
VT _{<i>x</i>}	VT of size “ <i>x</i> ” (currently $x = 1.5, 2, 3, 6$)
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queuing
WWW	World Wide Web (IETF)

APPENDIX B

References

REFERENCES

Reference	Source
AF ADDR GUIDE	ATM Forum, <i>ATM Forum Addressing: User Guide</i> , af-ra-0106-000, February 1999
AF ADDR REF	ATM Forum, <i>ATM Forum Addressing: Reference Guide</i> , af-ra-0105.000, January 1999
AF AIC-178	ATM Forum, <i>ATM-MPLS Network Interworking, Version 1.0</i> , AF-AIC-0178.000, August 2001
AF AINI	ATM Forum, <i>ATM Inter-Network Interface (AINI) Specification</i> , af-cs-0125.000, July 1999
AF ANS 2.0	ATM Forum, <i>ATM Named System v2.0</i> , af-dans-0152.000, July 2000
AF ARCH UNI	ATM Forum, <i>ATM UNI Specification Version 4.1</i> , Str-arch-UNI41-01.00, July 2002
AF BCS 1.0	ATM Forum, <i>Modification of Traffic Parameters for an Active Connection Signalling Specification (PNNI, AINI, and UNI) Version 1.0</i> , af-cs-0159.000, October 2000
AF BI ADDR	ATM Forum, <i>ATM Bi-level Addressing, Version 1.0</i> , af-ra-0164.000, April 2001
AF BICI 1.0	ATM Forum, <i>BISDN Inter Carrier Interface (B-ICI) Specification, Version 1.0</i> , August 1993
AF BICI 2.0	ATM Forum, <i>BISDN InterCarrier Interface (B-ICI) Specification, Version 2.0</i> , af-bici-0013.003, December 1995
AF BICI 2.1	ATM Forum, <i>Addendum to BISDN Inter Carrier Interface (B-ICI) Specification, v2.0 (B-ICI Specification, v2.1)</i> , af-bici-0068.000, November 1996
AF CES 2.0	ATM Forum, <i>Circuit Emulation Service 2.0</i> , af-vtoa-0078.000, January 1997
AF CS 116	af-cs-0116.000, <i>PNNI Version 1.0 Security Signaling Addendum</i> (May 1999)
AF CS 176	ATM Forum, <i>Loop Detection, Version 1.0</i> , af-cs-0176.000, April 2002
AF CS 181	ATM Forum, <i>Signalling Congestion Control, Version 1.0</i> , af-cs-0181.000, April 2002
AF CS 182	ATM Forum, <i>Call Processing Priority, Version 1.0</i> , af-cs-0182.000, April 2002
AF DBR	ATM Forum, <i>Domain-Based Rerouting for Active Point- to-Point Calls Version 1.0</i> , af-cs-0173.000, August 2001

Reference	Source
AF DIFF 1.0	<i>Addendum to TM 4.1:Differentiated UBR</i> , af-tm-0149.000, July 2000
AF DXI	ATM Forum, <i>Data Exchange Interface Version 1.0</i> , af-dxi-0014.000, August 1993
AF FAST	ATM Forum, <i>Frame-Based ATM over SONET/SDH</i> , af-fbatm-0151.000, July 2000
AF FATE	ATM Forum, <i>Frame-based ATM Transport over Ethernet (FATE)</i> , af-fbatm-0139.000, March 2000
AF FBATM	ATM Forum, <i>Frame-Based ATM Interface (Level 3)</i> , af-phy-0143.000, March 2000
AF FUNI 2.0	ATM Forum, <i>Frame-Based UNI (FUNI) Specification v2.0</i> , af-saa-0088.000, July 1997
AF GAT	<i>PNNI Addendum for Generic Application Transport Version 1.0</i> , af-cs-0126.000, July 1999
AF IISP	ATM Forum, <i>Interim Inter-Switch Signaling Protocol, version 1.0</i> , af-pnni-0026.000, December 1994
AF ILMI 4.0	ATM Forum, <i>Integrated Local Management Interface (ILMI) Specification Version 4.0</i> , af-ilmi-0065.000, September 1996
AF LANE 1.0	ATM Forum, <i>LAN Emulation Over ATM: Version 1.0 Specification</i> , af-lane-0021.000, January 1995
AF LANE 2.0	ATM Forum, <i>LAN Emulation over ATM—Version 2—LUNI Specification</i> , af-lane-0084.000, July 1997
AF LNNI 2.0	ATM Forum, <i>LAN Emulation over ATM Version 2—LNNI Specification</i> , af-lane-0112.000, February 1999
AF M4 CMIP	<i>M4 Network View CMIP MIB Spec v1.0</i> , af-nm-0073.000, January 1997
AF M4 SNMP	<i>SNMP M4 Network Element View MIB</i> , af-nm-0095.001, July 1998
AF MDCR 1.0	ATM Forum, <i>Addendum to TM4.1 Optional Minimum Desired Cell Rate Indication for UBR</i> , af-tm-0150.000, July 2000
AF MPOA 1.0	ATM Forum, <i>MultiProtocol Over ATM (MPOA), Version 1.0</i> , af-mpoa-0087.000, July 1997
AF NCCI	ATM Forum, <i>Network Call Correlation Identifier v1.0</i> , af-cs-0140.000, March 2000
AF PHY-16	ATM Forum, <i>DS1 Physical Layer Specification—Version 1.0</i> , af-phy-0016, September 1994
AF PNNI 1.0	ATM Forum, <i>Private Network-Network Interface Specification Version 1.0</i> , af-pnni-0055.00, March 1996

Reference	Source
AF PNNI 1.1	ATM Forum, <i>Private Network-Network Interface Specification Version 1.1</i> , af-pnni-0055.002, April 2002
AF SAA-119	ATM Forum, <i>Multiservice extensions to FUNI v2.0 Specification</i> , af-saa-0109.000, February 1999
AF SCC 1.0	ATM Forum, <i>Signalling Congestion Control Version 1.0</i> , af-cs-0181.000, January 2002
AF SECURITY	ATM Forum, <i>ATM Security Specification, Version 1.1</i> , af-sec-0100.002, March 2001
AF SPVC MIB	ATM Forum, <i>Private Network-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)</i> , af-pnni-0066.00, September 1996
AF TAS 1.0	ATM Forum, <i>PNNI Transported Address Stack, Version 1.0</i> , af-cs-0115.000, May 1999
AF TM 4.0	ATM Forum, <i>ATM Forum Traffic Management Specification, Version 4.0</i> , af-tm-0056.000, April 1996
AF TM 4.1	ATM Forum, <i>Traffic Management Specification, Version 4.1</i> , af-tm-0121.000, March 1999
AF TRACE	ATM Forum, <i>PNNI Addendum for Path and Connection Trace Version 1.0</i> , af-cs-0141.000, March 2000
AF UBR 1.0	ATM Forum, <i>UBR with MDCR Addendum to UNI 4.0/PNNI 1.0 AINI</i> , af-cs-0147.000, July 2000
AF UNI 2.0	ATM Forum, <i>ATM User-Network Interface (UNI) Specification, Version 2.0</i> , June 1992
AF UNI 3.1	ATM Forum, <i>UNI Signaling Specification, Version 3.1</i> , af-uni-0010.002, September 1994
AF UNI 4.0	ATM Forum, <i>User-Network Interface Signaling Specification, Version 4.0</i> , af-sig-0061.000, July 1996
AF UNI 4.1	ATM Forum, <i>ATM UNI Signalling Specification, Version 4.1</i> , af-sig-0061.002, April 2002
AF VMOA-145	ATM Forum, <i>Voice and Multimedia over ATM—Loop Emulation Service using AAL2</i> , af-vmoa-0145.000, July 2000
AF VOD 1.0	ATM Forum, <i>Audiovisual Multimedia Services: Video on Demand Specification 1.0</i> , af-saa-0049.000, December 1995
Alles 95	A. Alles, "ATM Interworking," Cisco white paper, 1995
Anatov 96	V. Antonov, "ATM: Another Technological Mirage," Pluris, Inc., 1996
Andersson 02	L. Andersson, "PPVPN: 2 Framework," work in progress, 2002

Reference	Source
ANSI T1.617a	ANSI, <i>ISDN—Signaling Specification for Frame Relay Bearer Service for Digital Subscriber Signaling System Number 1 (DSS1) (Protocol Encapsulation and PICS)</i> , T1.617a-1994, January 1994
ANSI T1.618	ANSI, <i>ISDN—Core Aspects of Frame Protocol for Use with Frame Relay Bearer Service</i> , T1.618-1991, October 1991
ANSI T1.624	ANSI, <i>BISDN UNI: Rates and Formats Specification</i> , T1.624-1993, 1993
ANSI T1.627	ANSI, <i>BISDN ATM Functionality and Specification</i> , T1.627-1993, 1993
ANSI T1.646	ANSI, <i>ANSI Standard for Telecommunications—Broadband ISDN—Physical Layer Specifications for User-Network Interfaces Including DS1/ATM</i> , T1.646-1995, May 12, 1995
ANSI X3.139	ANSI, <i>American National Standard for Information Systems—Fiber Distributed Data Interface (FDDI)—Token Ring Media Access Control (MAC)</i> , X3.139-1987, 1987
ANSI X3.186	ANSI, <i>American National Standard for Information Systems—Fiber Distributed Data Interface (FDDI)—Hybrid Ring Control (HRC)</i> , X3.186-1992, 1992
ANSI X3.239	ANSI, <i>American National Standard for Information Systems—Fiber Distributed Data Interface (FDDI)—Token Ring Media Access Control-2 (MAC-2)</i> , X3.239-1994, -1994
Apisdorf 97	J. Apisdorf, K. Claffy, K. Thompson, R. Wilder, "OC3MON: Flexible, Affordable, High Performance Statistics Collection," INET '97 Conference, June 1997
Armitage 95	G. Armitage, K. Adams, "How Inefficient Is IP over ATM Anyway?" <i>IEEE Network</i> , Jan/Feb 1995
ATMF M4 View	<i>M4 Interface Requirements and Logical MIB: ATM Network View, Version 2</i> , af-nm-0058.001, May, 1999
Augstyn 02	W. Augustyn et al., "Requirements for Virtual Private LAN Service (VPLS)," work in progress, 2002
Awater 91	G. Awater, F. Schoute, "Optimal Queueing Policies for Fast Packet Switching of Mixed Traffic," <i>IEEE JSAC</i> , April 1991
Banerjee 01	A. Banerjee et al., "Generalized MPLS: An Overview of Signaling Enhancements and Recovery Techniques," <i>IEEE Comm.</i> , July 2001
Bear 76	D. Bear, <i>Principles of Telecommunication—Traffic Engineering</i> , Peter Petringus, Ltd, 1976
Bellamy 82	J. Bellamy, <i>Digital Telephony</i> , John Wiley, 1982

Reference	Source
Bellcore SMDS	Bellcore, "Generic System Requirements in Support of Switched Multi-Megabit Data Service," <i>TR-TSV-000772 Issue 1</i> , May 1991
Benjamin 01	D. Benjamin, R. Trudel, E. Kus, "Optical Services over the Intelligent Optical Network," <i>IEEE Comm.</i> , September 2001
Bertsekas 92	D. Bertsekas, R. Gallager, <i>Data Networks, second edition</i> , Prentice-Hall, 1992
Black 92	U. Black, <i>TCP/IP and Related Protocols</i> , McGraw-Hill, 1992
Black 94	U. Black, <i>Frame Relay Networks, Specifications, and Implementations</i> , McGraw-Hill, 1994
Black 95	U. Black, <i>The X Series Recommendations</i> , McGraw-Hill, 1995
Bonaventure 98	O. Bonaventure, "A Simulation Study of TCP with the GFR Service Category," <i>High-Performance Networks for Multimedia Applications</i> , Kluwer, 1998
Bonenfant 01	P. Bonenfant, A. Rodrigues-Moral, "Framing Techniques for IP over Fiber," <i>IEEE Network</i> , July/August 2001
Bonica 2000	R. Bonica, "ICMP Extensions for MultiProtocol Label Switching," work in progress, 2001
Bonica 2001	R. Bonica et al., "Tracing Requirements for Generic Tunnels," work in progress, 2001
Bonomi 95	F. Bonomi, K. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," <i>IEEE Network</i> , March/April 1995
Brown 99	S. Brown, <i>Implementing Virtual Private Networks</i> , McGraw-Hill, 1999
Bryant 02	S. Bryant, "Protocol Layering in PWE3," work in progress, 2002
Cahn 98	R. Cahn, <i>Wide Area Network Design—Concepts and Tools for Optimization</i> , Morgan-Kaufmann, 1998
Cerf 74	V. Cerf, R. Kahn, "A Protocol for Packet Network Interconnection," <i>IEEE Trans. Comm.</i> , May 1974
Chiong 97	J. Chiong, <i>Internetworking ATM for the Internet and Enterprise Networks</i> , McGraw-Hill, 1997
Cikoski 96	T. Cikoski, "The Complexities and Future Evolution of SNMP as a Management Protocol," <i>Telecommunications</i> , August 1996
Claffy 98	K. Claffy, G. Miller, K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone" INET '98 Conference, April 1998
Comer 91	D.L. Comer, <i>Interworking with TCP/IP—Volume I: Principles, Protocols, and Architecture</i> , Prentice-Hall, 1991

Reference	Source
Cooper 81	R. Cooper, <i>Introduction to Queuing Theory, second edition</i> , North-Holland, 1981
Cope 01	J. Cope, "Sprint Drops ION, Freezes Fixed Wireless Service," <i>Computerworld</i> , October 22, 2001
Cucchiara 01	J. Cucchiara et al., "Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)," work in progress, 2001
Cypser 78	R. Cypser, <i>Communications Architecture for Distributed Systems</i> , Addison-Wesley, 1978
Dayton 91	R. Dayton, <i>Telecommunications</i> , McGraw-Hill, 1991
Dron 91	L. Dron, Sengupta Ramamurthy, "Delay Analysis of Continuous Bit Rate Traffic over an ATM Network," <i>IEEE JSAC</i> , April 1991
DSL TR017	DSL Forum TR-017, <i>ATM over ADSL</i> , Recommendations, 1999
Dziong 97	Z. Dziong, <i>ATM Network Resource Management</i> , McGraw-Hill, 1997
Ellaumi 98	O. Ellaumi, H. Afifi, "Evaluation of FIFO-based Management for TCP over Guaranteed Frame Rate Service," <i>IEEE ATM '98 Workshop Tutorial</i> , Fairfax, VA, May 1998
Extreme 01	Extreme Networks, "Leveraging MPLS to Enhance Service Network Transport Capabilities," www.extremenetworks.com , 2001
Extreme 02	Extreme Networks, "Metro Layer 2 VPN Services," www.extremenetworks.com , 2002
FDDI 98	FDDI Consortium, "FDDI Tutorial," http://www.iol.unh.edu/training/fddi/htmls/index.html , 1998
Feldman 97	N. Feldman, A. Viswanathan, "ARIS specification," work in progress, September, 1997, http://www.networking.ibm.com/isr/arisspec.html
Felstaine 99	E. Felstaine, R. Cohen, "On the Distribution of Routing Computation in Hierarchical ATM Networks," <i>IEEE/ACM Transactions on Networking</i> , December 1999
Finn 96	N. Finn, T. Mason, "ATM LAN Emulation," <i>IEEE Comm.</i> , June 1996
Floyd 93	Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," <i>IEEE/ACM Transactions on Networking</i> , August 1993
Floyd 94	S. Floyd, "TCP and Explicit Congestion Notification," <i>ACM Computer Communication Review</i> , October 1994

Reference	Source
Fowler 91	Leland, Fowler, "Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management," <i>IEEE JSAC</i> , September 1991
Fowler 99	D. Fowler, <i>Virtual Private Networks</i> , Morgan-Kaufmann, 1999
FRF FUNI	Frame Relay Forum, "Frame Relay and Frame-Based ATM: A Comparison of Technologies," 1995
FRF.1.2	Frame Relay Forum, "Frame Relay Forum UNI," FRF 1.2, July 2000
FRF.2.2	Frame Relay Forum, "Frame Relay Network-to-Network Interface IA," FRF.2.2, March 2002
FRF.20	Frame Relay Forum, "Frame Relay IP Header Compression IA," FRF.20, June 2001
FRF.3.2	Frame Relay Forum, "Multiprotocol Encapsulation IA, FRF.3.2," April 2000 .
FRF.4.1	Frame Relay Forum, "Frame Relay User-to-Network SVC IA," FRF.4.1, January 2000
FRF.4.1	Frame Relay Forum, "Frame Relay User-to-Network SVC IA," FRF.4.1, January 2000
FRF.5	Frame Relay Forum, "Frame Relay / ATM PVC Network Interworking IA," FRF.5, December 20, 1994
FRF.7	Frame Relay Forum, "Frame Relay PVC Multicast Service and Protocol Description," FRF.7, October 1994
FRF.8.1	Frame Relay Forum, "Frame Relay / ATM PVC Service Interworking IA," FRF.8.1, February 2000
FRF.9	Frame Relay Forum, "Data Compression over Frame Relay IA," FRF.9, January 1996
FRF.10.1	Frame Relay Forum, "Frame Relay Network-to-Network SVC," FRF.10.1, January 2000
FRF.11.1	Frame Relay Forum, "Voice over Frame Relay," FRF.11.1, December 1998
FRF.12	Frame Relay Forum, "Frame Relay Fragmentation," FRF.12, December 1997
FRF.13	Frame Relay Forum, "Service Level Definitions," FRF.13, August 1998
FRF.14	Frame Relay Forum, "Physical Layer Interface IA," FRF.14, December 1998
FRF.15	Frame Relay Forum, "End to End Multilink Frame Relay IA," FRF.15, August 1999

Reference	Source
FRF.16.1	Frame Relay Forum, "MultiLink Frame Relay UNI/NNI IA," FRF.16.1, 2002
FRF.17.1	Frame Relay Forum, "Frame Relay Privacy IA," FRF.17, January 2000
FRF.18	Frame Relay Forum, "Network to Network Frame Relay / ATM SVC Service Interworking IA," FRF.18, April 2000
FRF.19	Frame Relay Forum, "OA&M IA," FRF.19, April 2001
FRMPLS	FR Forum, MPLS Forum, "DRAFT Frame Relay and MPLS Network Interworking IA," 2002
Gallagher 68	R. Gallagher, <i>Information Theory</i> , John Wiley & Sons, 1968
Germain 96	E. Germain, "Fast Lanes on the Internet," <i>Science</i> , August 2, 1996
GMPLS ARCH	E. Mannie (editor), "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," <i>IETF</i> , work in progress, 2002
Goralski 00	W. Goralski, <i>SONET, second edition</i> , McGraw-Hill, 2000
Goralski 95	W. Goralski, <i>Introduction to ATM Networking</i> , McGraw-Hill, 1995
GR-303	Telcordia Generic Requirements GR-303-CORE Issue 2, 1998, Integrated Digital Loop Carrier System Generic Requirements, Objectives, and Interface
Gray 01	E. Gray, <i>MPLS, Implementing the Technology</i> , Addison-Wesley, 2001
Green 92	P. Green, "An All-Optical Computer Network: Lessons Learned," <i>IEEE Network</i> , March 1992
Gross 85	Gross, Harris, <i>Fundamentals of Queuing Theory</i> , Wiley, 1985
GSMP OPTICAL	G. Kullgren et al., "Requirements for Adding Optical Support to GSMPv3," <i>IETF</i> , work in progress, 2002
Guerin 91	R. Guerin, H. Ahmadi, M. Naghshineh, "Equivalent Capacity and Bandwidth Allocation," <i>IEEE JSAC</i> , September 1991
Held 95	G. Held, R. Sarch, <i>Data Communications</i> , McGraw-Hill, 1995
Hellstrand 98	F. Hellstrand, A. Veres, "Simulation of TCP/IP Router Traffic over ATM using GFR and VBR.3," ATM Forum 98-0087, February 1998
Henderson 96	L. Henderson, "Multimedia over IP: A new choice?" <i>Telephony</i> , July 22, 1996
Hluchyj 88	M. Hluchyj, M. Karol, "Queuing in High-Performance Packet Switching," <i>IEEE JSAC</i> , December 1988
Hobbes 02	R. Hobbes, "Hobbes' Internet Timeline," 2002, v5.6, http://www.isoc.org/guest/zakon/Internet/History/HIT.html .

Reference	Source
Hong 91	D. Hong, T. Suda, "Congestion Control and Prevention in ATM Networks," <i>IEEE Network Magazine</i> , July 1991
Hui 88a	B. Arthurs, J. Hui, "A Broadband Packet Switch for Integrated Transport," <i>IEEE JSAC</i> , December 1988
Hui 88b	J. Hui, "Resource Allocation for Broadband Networks," <i>IEEE JSAC</i> , December 1988
Huitema 95	C. Huitema, <i>Routing in the Internet</i> , Prentice-Hall, 1995
Hummel 02	H. Hummel, "Hierarchical LSP," Internet Draft, work in progress, March 2002
ID DSIM	Y. Bernet, <i>An Informal Management Model for Diffserv Routers</i> , IETF, work in progress.
ID IPDV	C. Demichelis et al., <i>Instantaneous Packet Delay Variation Metric for IPPM</i> , IETF, work in progress.
ID MPLSDS	F. Le Faucheur et al., <i>MPLS Support of Differentiated Services</i> , IETF, work in progress.
IEEE802.1D	ANSI/IEEE, <i>IEEE Standard for Information Technology, Telecommunications and Information Exchange Between Systems, Local and Metropolitan Area Networks, Common Specifications, Part 3: Media Access Control (MAC) Bridges</i> , Std 802.1D-1998, 1998
IEEE802.1Q	IEEE Std 802.1Q-1998, <i>IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks</i> , IEEE, December, 1998
IEEE802.2	ISO/IEC 8802-2, <i>ANSI/IEEE Std 802.2, Information Processing Systems—Local and Metropolitan Area Networks—Part 2: Logical Link Control, second edition</i> , ISO/IEC, 1994
IEEE802.3	ISO/IEC 8802-3, <i>ANSI/IEEE Std 802.3, Information Processing Systems—Local and Metropolitan Area Networks—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, fifth edition</i> , ISO/IEC, 1996
IEEE802.12	IEEE Std 802.12-1995, <i>Demand Priority Access Method, Physical Layer, and Repeater Specification for 100 Mb/s Operation</i> , IEEE, 1995
ISIS TE	T. Li, H. Smit, <i>IS-IS Extensions for Traffic Engineering</i> , work in progress, 2002
ISO 13239	ISO/IEC, <i>Information Technology—Telecommunications and Information Exchange Between Systems—High-Level Data Link Control (HDLC) Procedures</i> , ISO/IEC 13239, 1997-06-15
ISO 13818-3	ISO/IEC, <i>Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 3: Audio</i> , ISO/IEC 13818-3:1998

Reference	Source
ISO 8802.2	ISO/IEC, <i>Local and Metropolitan Area Networks—Specific Requirements—Part 2: Logical Link Control</i> , Standard 8802-2, 1994
ISO/IEC 8348	ISO/IEC, <i>Information Technology—Telecommunications and Information Exchange Between Systems—Network Service Definition</i> , Standard 8348, 1993
ISO8802-6	ISO/IEC, <i>Local and Metropolitan Area Networks—Part 6: Distributed Queue Dual Bus (DQDB) Subnetwork of a MAN</i> , 8802-6, 1993
ITU E.164	ITU-T, <i>The International Public Telecommunication Numbering Plan</i> , Rec. E.164, May 1997
ITU G.131	ITU-T, <i>Control of Talker Echo</i> , Rec. G.131, August 1996
ITU G.704	ITU-T, <i>Generic Framing Procedure (GFP)</i> , Rec. G.7041/Y.1303, December 2001
ITU G.707	ITU-T, <i>Network Node Interface for the Synchronous Digital Hierarchy (SDH)</i> , Rec. G.707, October 2000
ITU G.709	ITU, <i>Interfaces for the Optical Transport Network (OTN)</i> , Rec. G.709/Y.1331, February 2001
ITU G.803	ITU-T, <i>Architecture of Transport Networks Based on the Synchronous Digital Hierarchy (SDH)</i> , Rec. G.803, March 2000
ITU G.805	ITU-T, <i>Generic Functional Architecture of Transport Networks</i> , Rec. G.805, March 2000
ITU G.807	ITU-T, <i>Requirements for Automatic Switched Transport Networks (ASTN)</i> , Rec. G.807/Y.1302, July 2001
ITU G.8080	ITU-T, <i>Architecture for the Automatically Switched Optical Network (ASON)</i> , Rec. G.8080/Y.1304, November 2001
ITU H.222.0	ITU-T, <i>Generic Coding of Moving Pictures and Associated Audio Information: Systems</i> , Rec. H.222.0, February 2000 (Note: identical to ISO/IEC 13818-1:2000)
ITU H.222.1	ITU-T, <i>Multimedia Multiplex and Synchronization for Audiovisual Communication in ATM Environments</i> , Rec. H.222.1, March 1996
ITU H.225	ITU-T, <i>Call Signalling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems</i> , Rec. H.225, November 2000
ITU H.245	ITU-T, <i>Control Protocol for Multimedia Communication</i> , Rec. H.245, July 2001
ITU H.248	ITU-T, <i>Gateway Control Protocol</i> , Rec. H.248, June 2000
ITU H.262	ITU-T, <i>Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video</i> , Rec. H.262, February 2000 (Note: identical to ISO/IEC 13818-2:2000)

Reference	Source
ITU H.320	ITU-T, <i>Narrow-Band Visual Telephone Systems and Terminal Equipment</i> , Rec. H.320, May 1999
ITU H.321	ITU-T, <i>Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments</i> , Rec. H.321, February 1998
ITU H.323	ITU-T, <i>Packet-Based Multimedia Communications Systems</i> , Rec. H.323, November 2000
ITU I.150	ITU-T, <i>B-ISDN Asynchronous Transfer Mode Functional Characteristics</i> , February 1999
ITU I.311	ITU-T, <i>B-ISDN General Network Aspects</i> , Rec. I.311, August 1996
ITU I.321	ITU-T, <i>B-ISDN Protocol Reference Model and Its Application</i> , Rec. I.321, April 1991
ITU I.326	ITU-T, <i>Functional Architecture of Transport Networks Based on ATM</i> , Rec. I.326, November 1995
ITU I.350	ITU-T, <i>General Aspects of Quality of Service and Network Performance in Digital Networks, Including ISDNs</i> , Rec. I.350, March 1993
ITU I.356	ITU-T, <i>B-ISDN ATM Layer Cell Transfer Performance</i> , Rec. I.356, March 2000
ITU I.357	ITU-T, <i>B-ISDN Semi-Permanent Connection Availability</i> , Rec. I.357, November 2000
ITU I.361	ITU-T, <i>B-ISDN ATM Layer Specification</i> , Rec. I.361, February 1999
ITU I.362	ITU-T, <i>B-ISDN ATM Adaptation Layer (AAL) Functional Description</i> , Rec. I.362, March 1993
ITU I.363.1	ITU-T, <i>B-ISDN ATM Adaptation Layer Specification: Type 1 AAL</i> , Rec. I.363.1, August 1996
ITU I.363.2	ITU-T, <i>B-ISDN ATM Adaptation Layer Specification: Type 2 AAL</i> , Rec. I.363.2, November 2000
ITU I.363.3	ITU-T, <i>B-ISDN ATM Adaptation Layer: Type 3/4 AAL</i> , Rec. I.363.3, August 1996
ITU I.363.5	ITU-T, <i>B-ISDN ATM Adaptation Layer: Type 5 AAL</i> , Rec. I.363.3, August 1996
ITU I.365.1	ITU-T, <i>B-ISDN ATM Adaptation Layer Sublayers: Frame Relaying Service-Specific Convergence Sublayer (FR-SSCS)</i> , Rec. I.365.1, November 1993
ITU I.365.2	ITU-T, <i>B-ISDN ATM Adaptation Layer Sublayers: Service-Specific Coordination Function to Provide the Connection-Oriented Network Service</i> , Rec. I.365.2, November 1995

Reference	Source
ITU I.365.3	ITU-T, <i>B-ISDN ATM Adaptation Layer Sublayers: Service-Specific Coordination Function to Provide The Connection-Oriented Transport Service</i> , Rec. I.365.3, November 1995
ITU I.365.4	ITU-T, <i>B-ISDN ATM Adaptation Layer Sublayers: Service-Specific Convergence Sublayer for HDLC Applications</i> , Rec. I.365.4, August 1996
ITU I.366.1	ITU-T, <i>Segmentation and Reassembly Service-Specific Convergence Sublayer for the AAL Type 2</i> , Rec. I.366.1, June 1998
ITU I.366.2	ITU-T, <i>AAL Type 2 Service-Specific Convergence Sublayer for Narrow-Band Services</i> , Rec. I.366.2, November 2000
ITU I.370	ITU-T, <i>Congestion Management for the ISDN Frame Relaying Bearer Service</i> , Rec. I.370, October 1991
ITU I.371	ITU-T, <i>Traffic Control and Congestion Control in B-ISDN</i> , Rec. I.371, March 2000
ITU I.371.1	ITU-T, <i>Guaranteed Frame Rate ATM Transfer Capability</i> , Rec. I.371.1, November 2000
ITU I.380	ITU-T, <i>Internet Protocol Data Communication Service—IP Packet Transfer and Availability Performance Parameters</i> , Rec. I.380, February 1999
ITU I.413	ITU-T, <i>B-ISDN User-Network Interface</i> , Rec. I.413, March 1993
ITU I.432	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification</i> , Rec. I.432, March 1993
ITU I.432.1	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification: General Characteristics</i> , Rec. I.432.1, February 1999
ITU I.432.2	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification: 155 520 kbit/s and 622 080 kbit/s Operation</i> , Rec. I.432.2, February 1999
ITU I.432.3	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification: 1544 kbit/s and 2048 kbit/s Operation</i> , Rec. I.432.3, February 1999
ITU I.432.4	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification: 51 840 kbit/s Operation</i> , Rec. I.432.4, February 1999
ITU I.432.5	ITU-T, <i>B-ISDN User-Network Interface—Physical Layer Specification: 25 600 kbit/s Operation</i> , Rec. I.432.5, February 1999
ITU I.555	ITU-T, <i>Frame Relaying Bearer Service Interworking</i> , Rec. I.555, September 1997
ITU I.555	ITU-T, <i>Frame Relaying Bearer Service Interworking</i> , Rec. I.555, September 1997

Reference	Source
ITU I.580	ITU-T, <i>General Arrangements for Interworking Between B-ISDN and 64 kbit/s Based ISDN</i> , Rec. I.580, November 1995
ITU I.610	ITU-T, <i>B-ISDN Operation and Maintenance Principles and Functions</i> , Rec. I.610, February 1999, with addendum and corrigendum March 2000
ITU I.630	ITU-T, <i>ATM Protection Switching</i> , Rec. I.630, February 1999
ITU I.761	ITU-T, <i>Inverse Multiplexing for ATM (IMA)</i> , Rec. I.761, March 2000
ITU J.82	ITU-T, <i>Transport of MPEG-2 Constant Bit Rate Television Signals in B-ISDN</i> , Rec. J.82, July 1996
ITU M.3010	ITU-T, <i>Principles for Telecommunications Management Network</i> , Rec. M.3010, February 2000
ITU M.3400	ITU-T, <i>TMN Management Functions</i> , Rec. M.3400, February 2000
ITU Q.1901	ITU-T, <i>Bearer Independent Call Control Protocol</i> , Rec. Q.1901, June 2000
ITU Q.2100	ITU-T, <i>B-ISDN Signalling ATM Adaptation Layer (SAAL) Overview Description</i> , Q.2100, July 1994
ITU Q.2110	ITU-T, <i>Service-Specific Connection Oriented Protocol (SSCOP)</i> , Rec. Q.2110, July 1994
ITU Q.2111	ITU-T, <i>SSCOP in a Multi-Link And Connectionless Environment</i> , Rec. Q.2111, December 1999
ITU Q.2130	ITU-T, <i>B-ISDN SAAL—SSCF at UNI</i> , Rec. Q.2130, July 1994
ITU Q.2140	ITU-T, <i>B-ISDN SAAL—SSCF at NNI</i> , Rec. Q.2140, February 1995
ITU Q.2144	ITU-T, <i>B-ISDN SAAL—Layer Management for the SAAL at the NNI</i> , Rec. Q.2144, October 1995
ITU Q.2610	ITU-T, <i>B-ISDN—Usage of Cause and Location in B-ISUP and DSS 2</i> , Rec. Q.2610, December 1999
ITU Q.2610	ITU-T, <i>Broadband Integrated Services Digital Network (B-ISDN)—Usage of Cause and Location in B-ISDN User Part and DSS 2</i> , Rec. Q.2610, February 1995
ITU Q.2761	ITU-T, <i>Functional Description of B-ISUP of SS7</i> , Q.2761, December 1999
ITU Q.2762	ITU-T, <i>General Functions of Messages and Signals of B-ISUP of SS7</i> , Rec. Q.2762, December 1999
ITU Q.2763	ITU-T, <i>SS7 B-ISUP—Formats and Codes</i> , Rec. Q.2763, December 1999
ITU Q.2764	ITU-T, <i>SS7 B-ISUP—Basic Call Procedures</i> , Rec. Q.2764, December 1999

Reference	Source
ITU Q.2765	ITU-T, <i>SS7 B-ISUP—Application Transport Mechanism (APM)</i> , Rec. Q.2765, December 1999
ITU Q.2766.1	ITU-T, <i>Switched Virtual Path Capability</i> , Rec. Q.2766.1, May 1998
ITU Q.2767.1	ITU-T, <i>Soft PVC Capability</i> , Rec. Q.2767.1, June 2000
ITU Q.2931	ITU-T, <i>B-ISDN—DSS 2—User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control</i> , Rec. Q.2931, February 1995
ITU Q.2934	ITU-T, <i>DSS2—Switched Virtual Path Capability</i> , Rec. Q.2934, May 1998
ITU Q.2941	ITU-T, <i>DSS2—Generic Identifier Transport</i> , Recommendations Q.2941.1-3, 1997-2000
ITU Q.2951	ITU-T, <i>Number Identification Supplementary Services Using DSS2—Basic Call</i> , Rec. Q.2951.1-8, February, 1995
ITU Q.2957.1	ITU-T, <i>DSS2—Basic Call: User-to-User Signalling (UUS)</i> , Rec. Q.2957.1, February 1995
ITU Q.2961	ITU-T, <i>Broadband Integrated Services Digital Network (B-ISDN)—Digital Subscriber Signalling System No. 2 (DSS 2)—Additional Traffic Parameters</i> , Rec. Q.2961, October 1995
ITU Q.2961.3	ITU-T, <i>DSS2—Additional Traffic Parameters: Signalling Capabilities to Support Traffic Parameters for ABR ATM Transfer Capability</i> , Rec. Q.2961.3, September 1997
ITU Q.2962	ITU-T, <i>DSS2—Connection Characteristics Negotiation During Call/Connection Establishment Phase</i> , Rec. Q.2962, May 1998
ITU Q.2963	ITU-T, <i>Traffic Parameter Modification</i> , Rec. Q.2963.1-3, 1997-1999
ITU Q.2965.2	ITU-T, <i>DSS2—Signalling of Individual QoS Parameters</i> , Rec. Q.2965.2, December 1999
ITU Q.2971	ITU-T, <i>B-ISDN—DSS 2—UNI Layer 3 Specification for Point-to-Multipoint Call/Connection Control</i> , Rec. Q.2971, October 1995
ITU Q.2971	ITU-T, <i>Broadband Integrated Services Digital Network (B-ISDN)—Digital Subscriber Signalling System No. 2 (DSS 2)—User-Network Interface Layer 3 Specification for Point-to-Multipoint Call/Connection Control</i> , Rec. Q.2971, October 1995
ITU Q.922	ITU-T, <i>ISDN Data Link Layer Specification for Frame Mode Bearer Services</i> , ITU, Rec. Q.922, 1992

Reference	Source
ITU Q.933	ITU-T, <i>ISDN Signaling Specification for Frame Mode Bearer Services</i> , Rec. Q.933, 1991
ITU X.121	ITU-T, <i>International Numbering Plan for Public Data Networks</i> , Rec. X.121, October 2000
ITU X.144	ITU-T, <i>User Information Transfer Performance Parameters for Data Networks Providing International Frame Relay PVC Service</i> , Rec. X.144, October 2000
ITU X.213	ITU-T, <i>Information Technology—Open Systems Interconnection—Network Service Definition</i> , Rec. X.213, November 1995
ITU X.219	ITU-T, <i>Remote Operations: Model, Notation, and Service Definition</i> , Rec. X.219, November 1988
ITU X.229	ITU-T, <i>Remote Operations: Protocol Specification</i> , Rec. X.229, November 1988
ITU X.25	ITU-T, <i>Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit</i> , Rec. X.25, October 1996
ITU X.710	ITU-T, <i>Information Technology—Open Systems Interconnection—Common Management Information Service</i> , Rec. X.710, October 1997
ITU X.711	ITU-T, <i>Information Technology—Open Systems Interconnection—Common Management Information Protocol: Specification</i> , Rec. X.711, October 1997, along with later amendments and corrigenda
ITU X.722	ITU-T, <i>Information Technology—Open Systems Interconnection—Structure of Management Information: Guidelines for the Definition of Managed Objects</i> , Rec. X.722, January 1992, along with later amendments and corrigenda
ITU X.76	ITU-T Study Group 7, <i>Draft Amendment 1 to Rec. X.76 (SVC Part)</i> , April 1996
ITU Y.1310	ITU-T, <i>Transport of IP over ATM in Public Networks</i> , Rec. Y.1310, March 2000
ITU Y.1311	ITU-T, <i>Network Based VPNs—Generic Architecture and Service Requirements</i> , Rec. Y.1311, February 2002
ITU Y.1311.1	ITU-T, <i>Network-Based IP VPN over MPLS Architecture</i> , Rec. Y.1311.1, July 2001
ITU Y.1710	ITU-T, <i>Requirements for OAM Functionality for MPLS Networks</i> , Rec. Y.1710, July 2001

Reference	Source
Jabbari 92	B. Jabbari, D. McDysan, "Performance of Demand Assignment TDMA and Multicarrier TDMA Satellite Networks," <i>IEEE JSAC</i> , February 1992
Jain 88	R. Jain, K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals, and Methodology," <i>Computer Networking Symposium</i> , April 1988
Jain 96	R. Jain et al., "Source Behavior for ATM ABR Traffic Management: An Explanation," <i>IEEE Comm.</i> , November 1996
Jajszczyk 93	Mouftah Jajszczyk, "Photonic Fast Packet Switching," <i>IEEE Comm.</i> , February 1993
Keiser 85	B. Keiser, E. Strange, <i>Digital Telephony and Network Integration</i> , Van Nostrand Reinhold, 1985
Keshav 98	S. Keshav, R. Sharma, "Issues and Trends in Router Design," <i>IEEE Comm.</i> , May 1998
Kessler 93	G. Kessler, <i>ISDN, Concepts, Facilities, and Services, second edition</i> , McGraw-Hill, 1993
Kleinrock 75	L. Kleinrock, <i>Queuing Systems Volume I: Theory</i> , Wiley, 1975
Kleinrock 92	L. Kleinrock, "The Latency/Bandwidth Tradeoff in Gigabit Networks," <i>IEEE Comm.</i> , April 1992
Knight 02	P. Knight et al., "Network Based IP VPN Architecture Using Virtual Routers," work in progress, 2002
Kompella 02	K. Kompella, "A Traffic Engineering MIB," work in progress, 2002
Korn 85	I. Korn, <i>Digital Communications</i> , Van Nostrand Reinhold, 1985
Kosiur 98	D. Kosiur, <i>Building and Managing Virtual Private Networks</i> , Wiley, 1998
Kostas 98	T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec, J. Mahler, "Real-Time Voice over Packet Switched Networks," <i>IEEE Network</i> , January/February 1998
Krapf 01	E. Krapf, "A Bright Future for ATM?," <i>BCR</i> , September 2001
Kumar 98	V. Kumar, T. Lakshman, D. Stiliadis, "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet," <i>IEEE Comm.</i> , May 1998
Kung 95	H.T. Kung, R. Morris, "Credit-Based Flow Control for ATM Networks," <i>IEEE Network</i> , March/April 1995
Lai 02	W. Lai, D. McDysan (editors), "Network Hierarchy and Multilayer Survivability," work in progress, 2002

Reference	Source
Lambarelli	L. Lambarelli, "ATM Service Categories: The Benefits to the User," ATM Forum
Leland 93	Willinger Leland, Wilson Taqqu, "On the Self-Similar Nature of Ethernet Traffic," ACM Sigcomm '93, September 1993
Li 96a	C. Li, Y. Ofek, "Distributed Source-Destination Synchronization," <i>IEEE</i> , 1996
Li 96b	H. Li, K-Y Siu, H-Y Tzeng, "A Simulation Study of TCP Performance in ATM Networks with ABR and UBR Services," <i>IEEE</i> , 1996
Li 99	T. Li, "MPLS and the Evolving Internet Architecture," <i>IEEE Comm.</i> , December 1999
Liu 01	K. Liu, J. Ryan, "All the Animals in the Zoo: The Expanding Menagerie of Optical Components," <i>IEEE Comm.</i> , July 2001
Lyon 91	T. Lyon, "Simple and Efficient Adaptation Layer (SEAL)," <i>ANSI T1S1.5/91-292</i> , August 1991
Manchester 98	J. Manchester, J. Anderson, B. Doshi, S. Dravida, "IP over SONET," <i>IEEE Comm.</i> , May 1998
Martini 02	L. Martini et al., "Transport and Encapsulation of Layer 2 Frames over IP and MPLS Networks," work in progress, 2002
McDysan 00a	D. McDysan, <i>QoS and Traffic Management in IP and ATM Networks</i> , McGraw-Hill, 2000
McDysan 00b	D. McDysan, <i>VPN Applications Guide</i> , Wiley, 2000
McDysan 01	D. McDysan, "Implications of (G)MPLS Ubiquity (or Lack Thereof) on MPLS-based Services," MPLS 2001 Conference, October 2001
McDysan 89	D. McDysan, "Performance Analysis of Queuing System Models for Resource Allocation in Distributed Computer Networks," D.Sc. Dissertation, George Washington University, 1989
McDysan 94	D. McDysan, D. Spohn, <i>ATM: Theory and Application</i> , McGraw-Hill, 1994
McDysan 98	D. McDysan, D. Spohn, <i>ATM: Theory and Application, signature edition</i> , McGraw-Hill, 1998
McQuillan 97	J. McQuillan, "Deconstructing ATM," <i>BCR</i> , March 1997
Mier 00	E. Mier, "Integrated Access Devices: Mixing It Up," <i>BCR</i> , November 2000
Moy 98	J. Moy, <i>OSPF—Anatomy of an Internet Routing Protocol</i> , Addison-Wesley, 1998

Reference	Source
MSF ARCH 1.0	Multiservice Switching Forum, <i>System Architecture Implementation Agreement</i> , MSF-ARCH-001.00-FINAL IA, May 2000
MSF MGC	Multiservice Switching Forum, <i>Implementation Agreement for MEGACO/H.248 Profile for Media Gateway Controller/Media Gateway over IP and ATM Trunking</i> , MSF-IA-MEGACO.001, 002 and 003-FINAL, 2001 through 2002
Nadeau 01	T. Nadeau et al., "Multiprotocol Label Switching (MPLS) Management Overview," work in progress, 2001
Nadeau 02a	T. Nadeau et al., "Multiprotocol Label Switching (MPLS) FEC-To-NHLFE (FTN) Management Information Base," work in progress, 2002
Nadeau 02b	T. Nadeau et al., "MPLS/BGP Virtual Private Network Management Information Base Using SMIPv2," work in progress, 2002
Nolle 00	T. Nolle, "Access—ATM and ADSL Win," <i>BCR</i> , December 2000
Norros 95	I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Ethernet Traffic (Extended Version)," <i>IEEE JSAC</i> , August 1995
Odlyzko 01	A. Odlyzko, "Internet Pricing and the History of Communications," <i>Computer Networks 36 (2001)</i> , pp. 493–517.
OIF UNI 1.0	OIF, <i>User Network Interface (UNI) 1.0 Signaling Specification</i> , October 1, 2001
Onvural 97	R. Onvural, R. Cherakuri, <i>Signaling in ATM Networks</i> , Artech, 1997
OPSF TE	D. Katz, D. Yeung, K. Kompella, "Traffic Engineering Extensions to OSPF," work in progress, 2002
Orezessek 98	M. Orezessek, P. Sommer, <i>ATM & MPEG-2—Integrating Digital Video into Broadband Networks</i> , Prentice-Hall, 1998
Papoulis 91	A. Papoulis, <i>Probability, Random Variables, and Stochastic Processes, third edition</i> , McGraw-Hill, 1991
Partridge 94	C. Partridge, <i>Gigabit Networking</i> , Addison-Wesley, 1994
Pazos 95	C. Pazos, M. Gerla, V. Signore, "Comparing ATM Controls for TCP Sources," <i>IEEE</i> , 1995
Pepelnjak 01	I. Pepelnjak, J. Guichard, <i>MPLS and VPN Architectures</i> , Cisco Press, 2001
Perkins 97	D. Perkins, E. McGinnis, <i>Understanding MSMP MIBs</i> , Prentice-Hall, 1997

Reference	Source
Perlman 92	R. Perlman, <i>Interconnections</i> , Addison-Wesley, 1992
Personick 85	S. Personick, <i>Fiber Optics Technology and Applications</i> , Plenum, 1985
Peterson 72	F. Peterson, F. Weldon, <i>Error-Correcting Codes</i> , MIT Press, 1972
Petrovsky 98	M. Petrovsky, <i>Optimizing Bandwidth</i> , McGraw-Hill, 1998
Ping 02	P. Ping et al., "Detecting Data Plane Liveliness in MPLS," work in progress, 2002
PPVPN Framework	R. Callon, M. Suzuki, J. DeClerq, B. Gleeson, A. Malis, K. Muthukrishnan, E. Rosen, C. Sargor, J. Yu, "A Framework for Layer 3 Provider Provisioned Virtual Private Networks," 2002, work in progress.
PPVPN Requirements	M. Carugi, D. McDysan, L. Fang, A. Nagarajan, J. Sumimoto, R. Wilder, "Service Requirements for Provider Provisioned Virtual Private Networks," 2002, work in progress.
Proakis 83	J. Proakis, <i>Digital Communications</i> , McGraw-Hill, 1983
PWE3 Rqmts	X. Xiao et al., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)," work in progress, 2002
Ranade 89	J. Ranade, G. Sackett, <i>Introduction to SNA Networking</i> , McGraw-Hill, 1989
Rasmussen 91	Sorenson Rasmusen, Jacobsen Kvols, "Source-Independent Call Acceptance Procedures in ATM Networks," <i>IEEE JSAC</i> , April 1991
RFC 768	J. Postel, <i>User Datagram Protocol</i> , IETF, RFC 768, August 1980
RFC 791	J. Postel, <i>Internet Protocol</i> , IETF, RFC 791, September 1981
RFC 792	J. Postel, <i>Internet Control Message Protocol</i> , IETF, RFC 792, September, 1981
RFC 793	J. Postel, <i>Transmission Control Protocol</i> , IETF, RFC 793, September 1981
RFC 826	D. Plummer, <i>Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware</i> , IETF, RFC 826, November 1982
RFC 950	J.C. Mogul, J. Postel, <i>Internet Standard Subnetting Procedure</i> , RFC 950, IETF, August 1985
RFC 1034	P. Mockapetris, <i>Domain Names—Concepts and Facilities</i> , IETF, RFC 1034, November 1987
RFC 1055	J. Romkey, <i>A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP</i> , IETF, RFC 1055, IETF, June 1988

Reference	Source
RFC 1072	V. Jacobson, R. Braden, <i>TCP Extensions for Long-Delay Paths</i> , IETF, RFC 1072, October 1988
RFC 1112	S. Deering, <i>Host Extensions for IP Multicasting</i> , IETF, RFC 1112, August 1992
RFC 1122	R. Braden, <i>Requirements for Internet Hosts—Communication Layers</i> , IETF, RFC 1122, October 1989
RFC 1142	D. Oran, ed., <i>OSI IS-IS Intra-Domain Routing Protocol</i> , IETF, RFC 1142, February 1990
RFC 1157	J. Case, M. Fedor, M. Schoffstall, C. Davin, <i>Simple Network Management Protocol (SNMP)</i> , IETF, RFC 1157, May 1990
RFC 1191	J.C. Mogul, S.E. Deering, <i>Path MTU Discovery</i> , IETF, RFC 1191, November 1990
RFC 1195	R. Callon, <i>Use of OSI IS-IS for Routing in TCP/IP and Dual Environments</i> , IETF, RFC 1195, December 1990
RFC 1213	K. McCloghrie, M.T. Rose, <i>Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II</i> , IETF, RFC 1213, March 1991
RFC 1247	J. Moy, <i>OSPF Version 2</i> , IETF, RFC 1247, July 1991
RFC 1293	T. Bradley, C. Brown, <i>Inverse Address Resolution Protocol</i> , IETF, RFC 1293, January 1992
RFC 1323	V. Jacobson, R. Braden, D. Borman, <i>TCP Extensions for High Performance</i> , IETF, RFC 1323, May 1992
RFC 1363	C. Partridge, <i>A Proposed Flow Specification</i> , IETF, RFC 1363, September 1992
RFC 1435	S. Knowles, <i>IESG Advice from Experience with Path MTU Discovery</i> , IETF, RFC 1435, March 1993
RFC 1483	J. Heinanen, <i>Multiprotocol Encapsulation over ATM Adaptation Layer 5</i> , IETF, RFC 1483, July 1993 (Rendered obsolete by RFC 2684)
RFC 1490	T. Bradley, C. Brown, A. Malis, <i>Multiprotocol Interconnect over Frame Relay</i> , IETF, July 1993 (Rendered obsolete by RFC 2427)
RFC 1518	Y. Rechter, T. Li, <i>An Architecture for IP Address Allocation with CIDR</i> , IETF, RFC 1518, September 1993
RFC 1519	V. Fuller, T. Li, J. Yu, K. Varadhan, <i>Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy</i> , IETF, RFC 1519, September, 1993
RFC 1550	S. Bradner, A. Mankin, <i>IP: Next Generation (IPNG) White Paper Solicitation</i> , IETF, RFC 1550, December 1993

Reference	Source
RFC 1577	M. Laubach, <i>Classical IP and ARP over ATM</i> , IETF, RFC 1577, January 1994 (Rendered obsolete by RFC 2225)
RFC 1633	R. Braden, D. Clark, S. Shenker, <i>Integrated Services in the Internet Architecture: An Overview</i> , IETF, RFC 1633, June 1994
RFC 1661	W. Simpson, <i>The Point-to-Point Protocol (PPP)</i> , IETF, RFC 1661, July 1994
RFC 1662	W. Simpson, <i>PPP in HDLC-Like Framing</i> , IETF, RFC 1662, July 1994
RFC 1752	S. Bradner, A. Mankin, <i>The Recommendation for the IP Next Generation Protocol</i> , IETF, RFC 1752, January 1995
RFC 1755	M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman, A. Malis, <i>ATM Signaling Support for IP over ATM</i> , IETF, RFC 1755, February 1995
RFC 1771	Y. Rekhter, T. Li, <i>A Border Gateway Protocol 4 (BGP-4)</i> , IETF, RFC 1771, March 1995
RFC 1812	F. Baker, <i>Requirements for IP Version 4 Routers</i> , IETF, RFC 1812, June 1995
RFC 1817	Y. Rekhter, <i>CIDR and Classful Routing</i> , IETF, RFC 1817, August 1995
RFC 1883	S. Deering, R. Hinden, <i>Internet Protocol, Version 6 (IPv6) Specification</i> , IETF, RFC 1883, December 1995
RFC 1884	R. Hinden, S. Deering, <i>IP Version 6 Addressing Architecture</i> , IETF, RFC 1884, December 1995
RFC 1885	A. Conta, S. Deering, <i>Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)</i> , IETF, RFC 1885, December 1995
RFC 1886	S. Thomson, C. Huitema, <i>DNS Extensions to Support IP Version 6</i> , IETF, RFC 1886, December 1995
RFC 1889	H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, <i>RTP: A Transport Protocol for Real-Time Applications</i> , IETF, RFC 1889, January, 1996
RFC 1902	J. Case, K. McCloghrie, M. Rose & S. Waldbusser, <i>Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1902, January 1996
RFC 1903	J. Case, K. McCloghrie, M. Rose & S. Waldbusser, <i>Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1903, January 1996

Reference	Source
RFC 1904	J. Case, K. McCloaghrie, M. Rose, S. Waldbusser, <i>Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1904, January 1996
RFC 1905	J. Case, K. McCloaghrie, M. Rose, S. Waldbusser, <i>Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1905, January 1996
RFC 1906	J. Case, K. McCloaghrie, M. Rose, S. Waldbusser, <i>Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1906, January 1996
RFC 1907	J. Case, K. McCloaghrie, M. Rose, S. Waldbusser, <i>Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)</i> , IETF, RFC 1907, January 1996
RFC 1908	J. Case, K. McCloaghrie, M. Rose, S. Waldbusser, <i>Coexistence Between Version 1 and Version 2 of the Internet-Standard Network Management Framework</i> , IETF, RFC 1908, January 1996
RFC 1909	K. McCloaghrie, <i>An Administrative Infrastructure for SNMPv2</i> , IETF, RFC 1909, February 1996
RFC 1910	G. Waters, <i>User-Based Security Model for SNMPv2</i> , IETF, RFC 1910, February 1996
RFC 1932	R. Cole, D. Shur, C. Villamizar, <i>IP over ATM: A Framework Document</i> , IETF, RFC 1932, April 1996
RFC 1953	P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Liaw, T. Lyon, G. Minshall, <i>Ipsilon Flow Management Protocol</i> , IETF, RFC 1953, May 1996
RFC 1954	P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Liaw, T. Lyon, G. Minshall, <i>The Transmission of Flow Labelled IPv4 on ATM Data Links</i> , IETF, RFC 1954, May 1996
RFC 1968	G. Meyer, <i>The PPP Encryption Control Protocol (ECP)</i> , IETF, RFC 1968, June 1996
RFC 1969	K. Sklower, G. Meyer, <i>The PPP DES Encryption Protocol (DESE)</i> , IETF, RFC 1969, June 1996
RFC 1987	P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Liaw, T. Lyon, G. Minshall, <i>Ipsilon General Switch Management Protocol</i> , IETF, RFC 1987, August 1996
RFC 1990	K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti, <i>The PPP Multilink Protocol (MP)</i> , IETF, RFC 1990, August 1996
RFC 1997	R. Chandra, P. Traina, T. Li, <i>BGP Communities Attribute</i> , IETF, RFC 1997, August 1996

Reference	Source
RFC 2001	W. Stevens, <i>TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms</i> , IETF, RFC 2001, January 1997
RFC 2003	C. Perkins, <i>IP Encapsulation Within IP</i> , IETF, RFC 2003, October 1996
RFC 2022	G. Armitage, <i>Support for Multicast over UNI 3.0/3.1 based ATM Networks</i> , IETF, RFC 2022, November 1996
RFC 2105	Y. Rekhter, B. Davie, D. Katz, E. Rosen, G. Swallow, <i>Cisco Systems' Tag Switching Architecture Overview</i> , IETF, RFC 2105, February 1997
RFC 2205	R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, <i>Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification</i> , IETF, RFC 2205, September 1997
RFC 2206	F. Baker, J. Krawczyk, A. Sastry, <i>RSVP Management Information Base Using SMIv2</i> , IETF, RFC 2206, September 1997
RFC 2207	L. Berger, T. O'Malley, <i>RSVP Extensions for IPSEC Data Flows</i> , IETF, RFC 2207, September 1997
RFC 2208	A. Mankin, Ed., F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, <i>Resource ReSerVation Protocol (RSVP)—Version 1 Applicability Statement: Some Guidelines on Deployment</i> , IETF, RFC 2208, September 1997
RFC 2210	J. Wroclawski, <i>The Use of RSVP with IETF Integrated Services</i> , IETF, RFC 2210, September 1997
RFC 2211	J. Wroclawski, <i>Specification of the Controlled-Load Network Element Service</i> , IETF, RFC 2211, September 1997
RFC 2212	S. Shenker, C. Partridge, R. Guerin, <i>Specification of Guaranteed Quality of Service</i> , IETF, RFC 2212, September 1997
RFC 2215	S. Shenker, J. Wroclawski, <i>General Characterization Parameters for Integrated Service Network Elements</i> , IETF, RFC 2215, September 1997
RFC 2225	M. Laubach, J. Halpern, <i>Classical IP and ARP over ATM</i> , IETF, RFC 2225, April 1998
RFC 2236	W. Fenner, <i>Internet Group Management Protocol, Version 2</i> , IETF, RFC 2236, November 1997, Updates RFC 1112.
RFC 2283	T. Bates, <i>Multiprotocol Extensions for BGP-4</i> , IETF, RFC 2283, February 1998
RFC 2328	J. Moy, <i>OSPF Version 2</i> , IETF, RFC 2328, April 1998

Reference	Source
RFC 2330	V. Paxson, G. Almes, J. Nahvadi, M. Mathis, <i>Framework for IP Performance Metrics</i> , IETF, RFC 2330, May 1998
RFC 2331	M. Maher, <i>ATM Signalling Support for IP over ATM—UNI Signalling 4.0 Update</i> , IETF, RFC 2331, April 1998
RFC 2332	J. Luciani, D. Katz, D. Piscitello, B. Cole, N. Doraswamy, <i>NBMA Next Hop Resolution Protocol (NHRP)</i> , IETF, RFC 2332, April 1998
RFC 2363	G. Gross, M. Kaycee, A. Li, A. Malis, J. Stephens, <i>PPP over FUNI</i> , IETF, RFC 2363, July 1998
RFC 2370	R. Coltun, <i>The OSPF Opaque LSA Option</i> , IETF, RFC 2370, July 1998
RFC 2381	M. Garrett, M. Borden, <i>Interoperation of Controlled-Load Service and Guaranteed Service with ATM</i> , IETF, RFC 2381, August 1998
RFC 2390	T. Bradley, C. Brown, A. Malis, <i>Inverse Address Resolution Protocol</i> , IETF, RFC 2390, September 1998, Obsoletes RFC 1293
RFC 2401	S. Kent, R. Atkinson, <i>Security Architecture for the Internet Protocol</i> , IETF, RFC 2401, November 1998
RFC 2427	C. Brown, A. Malis, <i>Multiprotocol Interconnect over Frame Relay</i> , IETF, September 1998
RFC 2473	A. Conta, S. Deering., <i>Generic Packet Tunneling in IPv6 Specification</i> , IETF, RFC 2473, December 1998
RFC 2474	K. Nichols, S. Blake, F. Baker, and D. Black, <i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i> , IETF, RFC 2474, December 1998
RFC 2475	S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, <i>An Architecture for Differentiated Services</i> , IETF, RFC 2475, December 1998
RFC 2493	K. Tesink et al., <i>Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals</i> , IETF, RFC 2493, January 1999
RFC 2508	S. Casner, V. Jacobson, <i>Compressing IP/UDP/RTP Headers for Low-Speed Serial Links</i> , IETF, RFC 2508, February 1999
RFC 2512	K. McCloghrie, J. Heinanen, W. Greene, A. Prasad, <i>Accounting Information for ATM Networks</i> , IETF, RFC 2512, February 1999
RFC 2513	K. McCloghrie, J. Heinanen, W. Greene, A. Prasad, <i>Managed Objects for Controlling the Collection and Storage of Accounting Information for Connection-Oriented Networks</i> , RFC 2513, IETF, February 1999

Reference	Source
RFC 2514	M. Noto, E. Spiegel, K. Tesink, <i>Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management</i> , IETF, RFC 2514, February 1999
RFC 2515	K. Tesink, <i>Definitions of Managed Objects for ATM Management</i> , IETF, RFC 2515, February 1999
RFC 2543	M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, <i>SIP: Session Initiation Protocol</i> , IETF, RFC 2543, March 1999
RFC 2547	E. Rosen, Y. Rekhter, <i>BGP/MPLS VPNs</i> , IETF, RFC 2547, March 1999
RFC 2547bis	E. Rosen et al., "BGP/MPLS VPNs," work in progress, 2002
RFC 2558	K. Tesink, <i>Definitions of Managed Objects for the SONET/SDH Interface Type</i> , IETF, RFC 2558, March 1999
RFC 2570	J. Case, R. Mundy, D. Partain, B. Stewart, <i>Introduction to Version 3 of the Internet-Standard Network Management Framework</i> , IETF, RFC 2570, April 1999
RFC 2571	B. Wijnen, D. Harrington, R. Presuhn, <i>An Architecture for Describing SNMP Management Frameworks</i> , IETF, RFC 2571, April 1999
RFC 2572	J. Case, D. Harrington, R. Presuhn, B. Wijnen, <i>Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)</i> , IETF, RFC 2572, April 1999
RFC 2573	D. Levi, P. Meyer, B. Stewart, <i>SNMP Applications</i> , IETF, RFC 2573, April 1999
RFC 2574	U. Blumenthal, B. Wijnen, <i>User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)</i> , IETF, RFC 2574, April 1999
RFC 2575	B. Wijnen, R. Presuhn, K. McCloghrie, <i>View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i> , IETF, RFC 2575, April 1999
RFC 2576	R. Frye, D. Levi, S. Routhier, B. Wijnen, <i>Coexistence Between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework</i> , March 2000
RFC 2578	K. McCloghrie, D. Perkins, J. Schoenwaelder, <i>Structure of Management Information Version 2 (SMIv2)</i> , IETF, RFC 2578, April 1999
RFC 2579	K. McCloghrie, D. Perkins, J. Schoenwaelder, <i>Textual Conventions for SMIv2</i> , IETF, RFC 2579, April 1999
RFC 2597	F. Baker, J. Heinanen, W. Weiss, J. Wroclawski, <i>Assured Forwarding PHB Group</i> , IETF, RFC 2597, June 1999

Reference	Source
RFC 2615	A. Malis, W. Simpson, <i>PPP over SONET/SDH</i> , IETF, RFC 2615, June 1999
RFC 2638	K. Nichols, V. Jacobson, and L. Zhang, <i>A Two-Bit Differentiated Services Architecture for the Internet</i> , IETF, RFC 2638, July 1999
RFC 2661	W. Townsley et al., <i>Layer Two Tunneling Protocol 'L2TP'</i> , IETF, RFC 2661, August 1999
RFC 2674	E. Bell, A. Smith, P. Langille, A. Rijhsinghani, K. McCloghrie, <i>Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions</i> , IETF, RFC 2674 August 1999
RFC 2678	J. Mahvadi, V. Paxson, <i>IPPM Metrics for Measuring Connectivity</i> , IETF, RFC 2678, September 1999
RFC 2679	G. Almes et al., <i>A One-Way Delay Metric for IPPM</i> , IETF, RFC 2679, September 1999
RFC 2680	G. Almes et al., <i>A One-Way Packet Loss Metric for IPPM</i> , IETF, RFC 2680, September 1999
RFC 2681	G. Almes et al., <i>A Round-Trip Delay Metric for IPPM</i> , IETF, RFC 2681, September 1999
RFC 2684	D. Grossman, J. Heinanen, <i>Multiprotocol Encapsulation over ATM Adaptation Layer 5</i> , IETF, RFC 2684, September 1999
RFC 2685	B. Fox, B. Gleeson, <i>Virtual Private Networks Identifier</i> , IETF, RFC 2685, September 1999
RFC 2702	D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, <i>Requirements for Traffic Engineering over MPLS</i> , IETF, RFC 2702, September 1999
RFC 2784	D. Farinacci et al., <i>Generic Routing Encapsulation (GRE)</i> , IETF, RFC 2784, March 2000
RFC 2796	T. Bates, R. Chandra, E. Chen, <i>BGP Route Reflection: An Alternative to Full Mesh IBGP</i> , IETF, RFC 2796, April 2000
RFC 2805	N. Greene, M. Ramalho, B. Rosen, <i>Media Gateway Control Protocol Architecture and Requirements</i> , IETF, RFC 2805, April 2000
RFC 2858	T. Bates, Y. Rekhter, R. Chandra, D. Katz, <i>Multiprotocol Extensions for BGP-4</i> , IETF, RFC 2858, June 2000
RFC 2917	K. Muthukrishnan, A. Malis, <i>A Core MPLS IP VPN Architecture</i> , IETF, RFC 2917, September 2000
RFC 2961	L. Berger et al., <i>RSVP Refresh Overhead Reduction Extensions</i> , IETF, RFC 2961, April 2001

Reference	Source
RFC 2983	D. Black, <i>Differentiated Services and Tunnels</i> , RFC 2983, IETF, October 2000
RFC 3015	F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, J. Segers, <i>Megaco Protocol Version 1.0</i> , IETF, RFC 3015, November 2000
RFC 3031	E. Rosen, A. Viswanathan, R. Callon, <i>Multiprotocol Label Switching Architecture</i> , IETF, RFC 3031, January 2001
RFC 3032	E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta, <i>MPLS Label Stack Encoding</i> , IETF, RFC 3032, January 2001
RFC 3033	M. Suzuki, <i>The Assignment of the Information Field and Protocol Identifier in the Q.2941 Generic Identifier and Q.2957 User-to-User Signaling for the Internet Protocol</i> , IETF, RFC 3033, January 2001
RFC 3034	A. Conta, P. Doolan, A. Malis, <i>Use of Label Switching on Frame Relay Networks Specification</i> , IETF, RFC 3034, January 2001
RFC 3035	B. Davie, J. Lawrence, K. McCloghrie, E. Rosen, G. Swallow, Y. Rekhter, P. Doolan, <i>MPLS Using LDP and ATM VC Switching</i> , IETF, RFC 3035, January 2001
RFC 3036	L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas, <i>LDP Specification</i> , IETF, RFC 3036, January 2001
RFC 3037	B. Thomas, E. Gray, <i>LDP Applicability</i> , IETF, RFC 3037, January 2001
RFC 3038	K. Nagami et al., <i>VCID Notification over ATM Link for LDP</i> , IETF, RFC 3038, January 2001
RFC 3063	Y. Ohba, Y. Katsube, E. Rosen, P. Doolan, <i>MPLS Loop Prevention Mechanism</i> , IETF, RFC 3063, February 2001
RFC 3086	B. Carpenter, K. Nichols, <i>Definition of Differentiated Services per Domain Behaviors and Rules for Their Specification</i> , IETF, RFC 3086, April 2001
RFC 3107	Y. Rekhter, E. Rosen, <i>Carrying Label Information in BGP-4</i> , IETF, RFC 3107, May 2001
RFC 3108	R. Kumar, M. Mostafa, <i>Conventions for the Use of the Session Description Protocol (SDP) for ATM Bearer Connections</i> , IETF, RFC 3108, May 2001
RFC 3168	K.K. Ramakrishnan, S. Floyd, D. Black, <i>The Addition of Explicit Congestion Notification (ECN) to IP</i> , IETF, RFC 3168, September 2001
RFC 3209	D. Awduche et al., <i>RSVP-TE: Extensions to RSVP for LSP Tunnels</i> , IETF, RFC 3209, December 2001

Reference	Source
RFC 3210	D. Awduche et al., <i>Applicability Statement for Extensions to RSVP for LSP-Tunnels</i> , IETF, RFC 3210, December 2001
RFC 3212	B. Jamoussi, ed., <i>Constraint-Based LSP Setup using LDP</i> , IETF, RFC 3212, January 2002
RFC 3213	G. Ash et al., <i>Applicability Statement for CR-LDP</i> , IETF, RFC 3213, January 2002
RFC 3214	G. Ash et al., <i>LSP Modification Using CR-LDP</i> , IETF, RFC 3214, January 2002
RFC 3215	C. Boscher et al., <i>LDP State Machine</i> , IETF, RFC 3215, January 2002
RFC 3246	B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, <i>An Expedited Forwarding PHB (Per-Hop Behavior)</i> , IETF, RFC 3246, March 2002
RFC 3292	A. Doria, F. Hellstrand, K. Sundell, T. Worster, <i>General Switch Management Protocol (GSMP) V3</i> , IETF, RFC 3292, June 2002
RHK 02	M. Chalabi, J. Ryan, "Internet Traffic Soars, but Revenues Glide," <i>RHK</i> , May 2002
Riley 97	E. Riley, MPOA: MultiProtocol Over ATM, www.techguide.com , 1997
Romanow 95	A. Romanow, "Dynamics of TCP Traffic over ATM Networks," <i>IEEE JSAC</i> , May 1995
Rose 95	M Rose, K. McCloghrie, <i>How to Manage Your Network Using SNMP: The Networking Management Practicum</i> , Prentice-Hall, January 1995
Rose 96	M. Rose, <i>The Simple Book: An Introduction to Networking Management</i> , Prentice-Hall, April 1996
Sackett 96	G. Sackett, C. Metz, <i>ATM and Multiprotocol Internetworking</i> , McGraw-Hill, 1996
Saunders 96	S. Saunders, <i>The McGraw-Hill High-Speed LANs Handbook</i> , McGraw-Hill, 1996
Schwartz 77	M. Schwartz, <i>Computer Communication Design and Analysis</i> , Addison-Wesley, 1977
Schwartz 96	M. Schwartz, <i>Broadband Integrated Networks</i> , Prentice-Hall, 1996
Selga 81	J. Selga, J. Rivera, "HDLC Reliability and the FRBS Method to Improve It," <i>Proceedings of the 7th Data Communications Symposium</i> , Mexico City, 1981
Shah 02	H. Shah et al., "ARP Mediation for IP Interworking of Layer 2 VPN," work in progress, 2002

Reference	Source
Shannon 48	C. Shannon, "A Mathematical Theory of Communication," <i>Bell System Technical Journal</i> , vol. 27, October 1948
Sharma 02	V. Sharma et al., "Framework for MPLS-Based Recovery," work in progress, 2002
SIG DXI	SMDS Interest Group, SMDS Data Exchange Interface Protocol, Revision 3.2, <i>SIG-TS-001/1991</i> , October 1991
SIG SMDS-ATM	SMDS Interest Group, <i>Protocol Interface Specification for Implementation of SMDS over and ATM-Based Public UNI, Revision 2.0</i> , October 1996
Spohn 96	D. Spohn, <i>Data Network Design, second edition</i> , McGraw-Hill, 1996
Srinivasan 02a	C. Srinivasan et al., "Multiprotocol Label Switching (MPLS) Label Switch Router (LSR) Management Information Base," work in progress, 2002
Srinivasan 02b	C. Srinivasan et al., "Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base," work in progress, 2002
Stallings 98	W. Stallings, <i>High-Speed Networks—TCP/IP and ATM Design Principles</i> , Prentice-Hall, 1998
Streenstrup 95	M. Streenstrup, <i>Routing in Communication Networks</i> , Prentice-Hall, 1995
Sullebarger 97	R. Sullebarger, "ATM and Frame Relay: Making the Connection," <i>Data Communications</i> , June 1997
Swallow 96	G. Swallow, "MPOA, VLANS and Distributed Routers," <i>ATM Forum 53 Bytes, Volume 4, Issue 3</i> , 1996
Tannenbaum 96	A. Tannenbaum, <i>Computer Communications, third edition</i> , Prentice-Hall, 1996
Taylor 98	E. Taylor, <i>Internetworking Handbook</i> , McGraw-Hill, 1998
Telstra 02	Telstra, Internet BGP Table, http://bgp.potaroo.net/as1221/bgp-active.html , July 2002
Thompson 97	K. Thompson, G. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," <i>IEEE Network</i> , November/December 1997
Thompson 97	K. Thompson, G. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," <i>IEEE Network</i> , November/December 1997
Turner 86	J. Turner, "Design of an Integrated Services Packet Network," <i>IEEE Transactions on Communications</i> , November 1986

Reference	Source
Vickers 96	R. Vickers, H. Shimung, "Voice on ATM—Issues and Potential Solutions," <i>Proceedings of the International Conference on Communication Technology, ICCT '96</i> , Beijing, China.
VoMPLS 1.0	MPLS Forum, <i>Voice Over MPLS—Bearer Transport Implementation Agreement, Version 1.0</i> , July 27, 2001
Wernik 92	M. Wernik, Gilbert Aboul-Magd, "Traffic Management for B-ISDN Services," <i>IEEE Network</i> , September 1992
Woodruff 90	G. Woodruff, R. Kositpaiboon, "Multimedia Traffic Management Principles for Guaranteed ATM Network Performance," <i>IEEE JSAC</i> , April 1990
Wright 96	D. Wright, "Voice over ATM: An Evaluation of Network Architecture Alternatives," <i>IEEE Network</i> , September/October 1996
Yang 95	C. Yang, A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks," <i>IEEE Network</i> , July/August 1995
Zhang 01	Z. Zhang et al., "Lightpath Routing for Intelligent Optical Networks," <i>IEEE Network</i> , July/August 2001
Zhang 93	L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource ReSerVation Protocol," <i>IEEE Network</i> , September 1993



Index

Specification numbers appear under the respective standards bodies—CCITT, FRF, IEEE, IETF, ITU-T, and so on.

Numbers

- 100BASE-FX standard for 100 Mbps Fast Ethernet, specifications of, 223
- 100BASE-T4 standard for 100 Mbps Fast Ethernet, specifications of, 223
- 100BASE-TX standard for 100 Mbps Fast Ethernet, specifications of, 223
- 100VG-AnyLAN IEEE standard, explanation of, 223–224
- 1 protection switching, role in ATM, 808–809
- 1.544 Mbps North American N-ISDN PRI frame structure, diagram of, 116

▼ A

- A notation in queuing system models, meaning of, 705
- AA (Administrative Authority) in ATM control plane, explanation and diagram of, 381–382
- AAL and ATM service category used by applications, table of, 358
- AAL (ATM adaptation layer)
 - and ATM control plane protocols, 361–368
 - attributes, 322–323
 - components of, 320
 - dynamics of, 282, 320–323
 - layered signaling model for, 365–366
 - sublayers of, 282
- AAL bearer classes, list of, 320–321
- AAL PDUs, role in MPLS over ATM, 316–317
- AAL protocol structure, explanation of, 321–322

- AAL1
 - and AAL2 support for voice, 828
 - and CES (Circuit Emulation Service), 463–466
 - clock recovery methods, 330–332
 - CS (Convergence Sublayer) functions, 325–326
 - explanation of, 323–332
 - SAR (Segmentation and Reassembly) sublayer, 324–325
 - SDT CS (Structured Data Transfer Convergence Sublayer), 327–328
 - unstructured mode CS, 329–330
- AAL1-based CES
 - explanation of, 464–465
 - structured mode circuit emulation, 465–466
 - unstructured mode circuit emulation, 464–465
- AAL2
 - broadband local loop emulation with, 459–463
 - CPS PDU format, diagram of, 335
 - explanation of, 323, 332–336
 - narrowband SSCS message types and packet content, 454–455
 - narrowband SSCS UII codepoint
 - determination of packet content, 454–455
 - example of, 335–336
 - protocol structure and PDU formats, 333
- AAL2 narrowband SSCS
 - role in VoATM trunk signaling, 452–456
 - type 1 packets versus VoMPLS primary subframes, 459
 - using with ATM Forum loop emulation service, 460
- AAL3/4, 323, 337–340
 - versus AAL5, 345
 - CPCS sublayer, 338–339
 - multiplexing, 340–341

- operation, 339–340
SAR sublayer, 337–338
- AAL5
versus AAL3/4, 345
connection performance statistics table, 786
CPCS sublayer, 342–343
explanation of, 323, 340–346
MTU path discovery over, 560–561
multiprotocol encapsulation over, 506–511
role in GFR optimization for packet switching, 598
SAR sublayer, 342
- AAL5 frames, tagging in GFR.2, 600
- AAL5 multiplexing, example of, 344–345
- AAL5 operation, example of, 343–344
- AALs (ATM adaptation layers), role in packet switching, 74
- a.b.* notation for IP addresses, explanation of, 310
- abbreviations and acronyms, list of, 866–880
- ABM (asynchronous balanced mode), role in HDLC, 126
- ABR (Available Bit Rate) class of service
in ATM as end-to-end service, 261
binary mode ABR, 660–661
in closed-loop flow control, 656, 657–659
conformance checking, 666
in congestion control, 650
destinations in, 660
ER ABR, 661–662
explanation of, 587
parameters and resource management cells, 663–666
service parameters, 663–666
sources in, 660
VS/VD ABR, 662–663
- ABR RM cells insertion and feedback, diagram of, 664
- ABR setup parameter in UNI 4.1 advisory about, 370
role in UNI 4.1 signaling messages, 372
- abstract nodes, role in MPLS signaling and routing protocols, 392
- AC (Access Control) field in Token Ring MAC PDU, purpose of, 222
- AccBCT (Acceptable Burst Cell Tolerance), role in GFR traffic contract, 599
- accelerating bandwidth principle, explanation of, 21–22
- access
ATM support for, 476
attributes for, 476
diagram of, 475
access connections, role in VPNs, 546
access lines, explanation of, 50
- access side, role in multiplexing, 57–58
- accounting management of networks, explanation of, 765
- ACF (Access Control Field) in DQDB, purpose of, 201
- ACK (acknowledgments) in LAP-B, purpose of, 134
- ACK bit in TCP, purpose of, 191
- Acknowledgment Number field in TCP, purpose of, 191
- ACM (address complete message), role in VoATM trunk signaling, 452
- ACR (Available Cell Rate)
nonadditive link attributes
default value for, 663–664
role in PNNI, 428
- acronyms and abbreviations, list of, 866–880
- Activation/Deactivation procedures in ATM PM, explanation of, 811–812
- adaptive flow control. *See* closed-loop flow control
- ADD PARTY messages, role in point-to-multipoint connections, 377
- additive link attributes, role in PNNI, 428
- address field formats in HDLC frames, explanation of, 127
- address multiplexing, explanation of, 55, 60–61
- address switching
example of, 64–65
occurrence of, 54
- addressing and hierarchy, role in scalability analysis, 837–838
- addressing in ATM control plane, explanation of, 378
- admission control, role in traffic contracts and resource management, 634–638
- ADSL (Asymmetric Digital Subscriber Line), explanation of, 293
- ADTF (ACR Decrease Time Factor), default value for, 663
- advertisements, role in LDP protocol, 399–400
- AESA (ATM End System Address) in ATM control plane
explanation and diagram of, 379–380
explanation of, 381–383
- AF (Assured Forwarding) PHB, role in Diffserv, 594
- AFI (Authority and Format Identifier) in ATM control plane, explanation and diagram of, 381–382
- aggregated networks, role in PNNI, 424–426
- aggregated routing
and network-based VPNs using tunnels, 550–554
- role in IP VPNs over MPLS or IP tunnels, 550
- AINI (ATM Internetwork Interface) asymmetric soft rerouting, 437
explanation of, 434–439
versus PNNI, 434
role of RHC in, 439
symmetric soft rerouting, 438
- AIS and RDI theory and operation, role in ATM OAM fault management, 800–803
- Alamo Trader's Market in Texas network, sample network addresses in, 243–245
- ALERTING messages
in ATM signaling, 374
in N-ISDN D-channel switching, 118
- ANM (answer message), role in VoATM trunk signaling, 452
- ANS (ATM Name System) specification, role in ATM control plane addressing, 384–385
- ANSI (American National Standards Institute), purpose of, 30
- ANSI and ITU-T standards for Frame Relay, explanation of, 151–152
- ANSI Standards
T1.617 Annexes B and D (Frame Relay), 153–154
T1.618 (Frame Relay Congestion Control), 147, 151
X3.239 (FDDI MAC protocol), 227
X3.241-1994 (FRF.9 default compression function), 166
- anycast
in ATM control plane addressing, 382–383
in UNI 4.0, 370
- APPC (Advanced Program-to-Program Communication), development of, 86
- application layer (L7) of OSIRM, dynamics of, 84
- applications
impact of delay on, 732–734
impact of delay variation on, 738–742
impact of loss on, 735–738
- APPN (Advanced Peer-to-Peer Networking), development of, 86
- AREA (Area Identifier) in ATM control plane, explanation and diagram of, 381–382
- ARIS (Aggregate Route-Based IP Switching), role in MPLS, 264, 270–272
- ARM (asynchronous response mode), role in HDLC, 126
- ARP (Address Resolution Protocol) example configuration, 246
explanation of, 210, 245–247
in FR/ATM service interworking, 483–484
in TCP/IP, 178

- ARPANET, origins of, 71
- AS (autonomous system) numbers
 - usage with BGP, 405–406
 - usage with path vectors, 212
- AS-PATH path attribute for NLRI information, role in BGP, 406
- ASIC (application-specific integrated circuit), role in hardware and software complexity analysis, 843–844
- ASN.1 (Abstract Syntax Notation), role in SNMP, 778
- associated signaling channel configuration, role in ATM control plane protocols, 363
- asymmetric soft rerouting, role in AINI, 437
- asymptotic results in output buffer overflow probability, comparing, 715–716
- asynchronous clock recovery, role in AAL1, 330
- asynchronous data transmission, explanation of, 50–52
- ATDM (asynchronous time division multiplexing), explanation of, 55, 60–61
- ATM access, diagram of peer-to-peer communications for, 262
- ATM Activation/Deactivation OAM cell function-specific fields, diagram of, 811–812
- ATM and AAL service category used by applications, table of, 358
- ATM and MPLS circuit emulation, using circuit emulation with, 463–467
- ATM and MPLS control plane protocols
 - generic control plane functions, 359
 - switched and permanent ATM virtual connections, 359–360
- ATM and MPLS implementations, hardware price-performance trade-offs of, 280–281
- ATM and MPLS mix of voice and IP traffic, average efficiency of, 837
- ATM and MPLS protocol stack, diagram of, 274
- ATM and MPLS protocol support for IP, diagram of, 532
- ATM and SDH management plane reference architecture, diagram of, 797
- ATM (Asynchronous Transfer Mode)
 - accessing SMDS, 488–489
 - application of, 256–262
 - and B-ISDN, 254–256
 - connection methods for, 359
 - diagram of, 256
 - versus DS, 604
 - as end-to-end service, 261
 - explanation of, 50–53
 - frame-based interfaces for, 489
 - and FUNI, 494
 - future directions and applications of, 852–853
 - future of, 863
 - infrastructure of, 4–6
 - as integrated access, 259–260
 - as interface, 258–259
 - as protocol, 257–258
 - as scalable infrastructure, 261–262
 - as technology, 257
 - VCs (virtual containers), 297–302
 - voice trunking, 449–450
 - VPs (virtual paths), 297–302
- ATM CAC, role in traffic contracts and resource management, 635–637
- ATM CBR versus rt-VBR service category support for voice, 457
- ATM cell format including header and payload, diagram of, 303
- ATM cell sizes, rational for choices of, 278–279
- ATM cell transfer outcomes, explanations of, 814–816
- ATM cells
 - carrying over MPLS, 346
 - dynamics of, 280, 302–306
- ATM CES, role in AAL clock recovery methods, 330–331
- ATM control plane addressing
 - AESA formats, 381–383
 - bi-level addressing, 384
 - formats, 379–381
 - group addresses and anycast, 382–383
 - ILMI protocol, 382–383
 - level addressing, 378–379
- ATM control plane protocols and AAL, 361–368
 - explanation of, 360–361
 - signaling channel configurations for, 363–365
- ATM devices, dynamics of, 298
- ATM DXI
 - versus FUNI and FR, 495–496
 - header formats, 492
 - LMI (local management interface) summary, 492–493
 - Mode 1a and Mode 1b, 490–491
 - Mode 2, 491–492
- ATM for multiservice networking
 - advisory about per-flow level, 853–854
 - bandwidth reservation for constraint-based routing, 854
 - network assumptions, 854–855
 - QoS and traffic management on aggregates, 854
- ATM Forum
 - diagram of leaky bucket configurations, 584
 - diagram of RM cell contents, 665
 - versus FRF, 29
 - LANE (LAN Emulation), 511–520
 - loop emulation service using AAL2 narrowband SSCS, 460
 - purpose of, 27–28
 - QoS classes and service categories, 586–588
 - UNI 3.1 specification, 368
 - VOD (Voice on Demand) specification, 469–470
- ATM Forum and ITU-T QoS definitions, mapping between, 591–594
- ATM Forum and ITU-T UNI signaling capabilities, diagram of, 369
- ATM ILMI MIBs, dynamics of, 783–784
- ATM integrated access, diagram of, 260
- ATM interworking and Frame Relay, dynamics of, 477–478
- ATM LAN emulation and legacy LAN devices, diagram of, 512
- ATM layer
 - explanation of, 296
 - QoS and service categories, 306–308
- ATM MIBs. *See also* MIBs (Management Information Bases), MPLS MIBs
 - ATM Forum ILMI, 783–786
 - AToM, 786–787
 - other types of, 787–788
- ATM OAM
 - CC (continuity checks), 806
 - cell formats, 798–800
 - fault management, 800–806
 - flow reference architecture, 796–798
- ATM over MPLS
 - encapsulation modes, 349
 - network interworking, 348–351
- ATM performance parameters
 - CDV (cell delay variation) measurement, 817–818
 - cell error ratio, 816
 - cell loss ratio, 817
 - cell misinsertion rate, 817
 - CTD (Cell Transfer Delay) measurement, 817–819
 - severely errored cell block ratio, 816
- ATM performance specification and measurement
 - network performance and QoS, 810
 - NP/QoS parameter estimation, 814–819
 - PM (performance measurement), 810–814
- ATM physical layer
 - inverse multiplexing over ATM, 290–292
 - Physical Medium-Dependent sublayer of, 285–287
 - table of interfaces, media, and bit rates, 286–287
 - TC cell rate decoupling, 290
 - TC HEC (Header Error Check) functions, 288–290

TC (Transmission Convergence)
 sublayer of, 287–288
 xDSL physical layer for ATM,
 292–295

ATM PM cells, detection
 activity of, 813

ATM policing, dynamics of, 613

ATM protection switching,
 dynamics of, 806–809

ATM protocol model
 hardware-to-software
 progression, diagram of, 285

ATM PT encoding and meaning,
 table of, 307

ATM QoS parameters, mapping
 network impairments to, 577

ATM service categories
 diagram of attributes for, 589
 GFR (Guaranteed Frame Rate)
 optimized for packet
 switching, 597–603
 QoS parameters for, 588

ATM signaling procedures
 point-to-multipoint connections,
 376–378
 point-to-point connections,
 373–375

ATM SVCs, role in Classical IP over
 ATM, 536

ATM switches, diagram of IP
 routers connected with, 263

ATM traffic conformance,
 explanation of, 583–585

ATM traffic descriptors
 allocation of tolerances, 581
 and tolerances, 581

ATM traffic parameters,
 diagram of, 580

ATM UNI signaling
 Q.2931 and UNI 3.1 base
 signaling functions, 368
 UNI 4.0 and ITU-T standards,
 368–370

ATM user plane protocols diagram
 for voice, video, and WAN
 data, 355

ATM versus IP QoS and traffic
 parameters, comparison of,
 596–597

ATM VP/VC 1+1 linear protection
 switching, diagram of, 808–809

ATM VPs and label stacked MPLS
 LSPs, dynamics of, 638–639

ATMARP (ATM Address
 Resolution Protocol), role in IP
 over ATM VPNs, 534–535

AToM MIBs, dynamics of, 786–787

audio and video protocols,
 sensitivity to delay variation, 738

AUGs (administrative unit groups),
 role in SONET frame format, 112

AUs (administrative units), role in
 SONET frame format, 112

availability, explanation of and
 concerns about, 846–847

AVCs (aggregated virtual circuits),
 role in MLFR, 157, 159

▼ B

B (bearer) channels, role in
 N-ISDN, 114

B-ICI (Broadband Inter-carrier
 Interface), explanation of, 433

B-ISDN/ATM layer and sublayer
 model, diagram of, 283

B-ISDN (Broadband Integrated
 Services Digital Network)
 architecture of, 255–256
 and ATM, 254–256

B-ISDN control plane, explanation
 of, 362

B-ISDN layers
 hardware and software
 implementations of, 284–285
 mapping to OSI layers, 284

B-ISDN protocol layer structure,
 explanation of, 283–285

B-ISDN protocol model,
 diagram of, 254

B-ISDN UNI and NNI signaling
 protocol stack, explanation and
 diagram of, 362–363

B-ISUP protocol
 explanation of, 362–363
 role of RHC in, 439

B notation in queuing system
 models, meaning of, 705

B-pictures, usage in MPEG-2, 469

balanced HDLC control links,
 explanation of, 126

Balanced interchanges, role in
 DTE-to-DCE connections, 49

bandwidth-limited application,
 occurrence of, 732

bandwidth reservation, complexity
 analysis considerations, 844–845

bandwidth, role in binary on/off
 keying, 681

basic mode, role in HDLC, 128

baud, role in QAM, 683

BCDs (binary coded decimals) in
 ATM control plane addresses,
 digits in, 380

BCFs (bearer control functions), role
 in VoATM trunk signaling, 451

BCT (Burst Cell Tolerance), role in
 GFR traffic contract, 598

Bearer Classes A, C, and X in AAL,
 explanation of, 321

BECN (Backward Explicit
 Congestion Notification), role in
 Frame Relay, 142, 146–147

BEDC-0+1 field in PM, meaning
 of, 812

BER (bit error rate), role in
 communications
 engineering, 677

BER performance of modulation
 schemes, chart of, 688

Bernoulli processes and random
 trials, role in communications
 engineering, 678

BGP (Border Gateway Protocol)
 distributing labels for LSPs
 with, 412–413
 explanation of, 212
 using for label distribution,
 405–407

BGP/MPLS VPNs
 configuration complexity of, 558
 considerations and trade-offs
 with, 556
 usage of protocols by, 551–553

BGP update message contents,
 diagram of, 406

bidirectional rings
 explanation of, 109
 role in VCs, 299

binary mode ABR, dynamics of,
 660–661

binary on/off keying, dynamics of,
 680–681

BISUP (B-ISDN User Services Part),
 explanation of, 433–434

bit stuffing, role in HDLC, 127

bit transmission in Frame Relay,
 explanation of, 144–145

bitwise parity checking, performing,
 689–690

Block Control Check, role in BSC, 52

blocked calls cleared switching
 systems, dynamics of, 709–711

blocked calls held switching system,
 dynamics of, 711–712

BLSR (bidirectional line-switched
 ring), explanation of, 109

BOM (Beginning of Message) in
 AAL3/4, explanation of, 337–338

BPSR (bidirectional path-switched
 ring), explanation of, 109

BR (backward reporting) ATM
 function-specific fields in PM,
 diagram of, 813

BRI and PRI service and protocol
 structures, explanations of,
 115–117

BRI (Basic Rate Interface), role in
 N-ISDN, 113–114

bridged protocols, features of,
 507–509

bridges
 network design of, 234–235
 taxonomy of, 231–232

bridging
 basic terminology for, 208–210
 concepts, systems, and protocols
 for, 229–235
 context for, 230–231
 versus routing, 248
 and routing system design,
 247–249

bridging protocols, 802.1Q VLAN
 standard as, 220

broadband local loop emulation,
 using AAL2 for, 459–463

broadcast subnets, explanation
 of, 208

broadcast topology, explanation of,
 42–44

- BSC (Binary Synchronous Communications), role in UBR and MDCR, 603–604
- BSC communications channel error model, explanation of, 685
- BTag (Beginning Tag) in AAL3/4 CPCS sublayer, explanation of, 338
- buckets, purpose in IP and MPLS, 612
- buffer management, modifying for GFR, 602–603
- buffer requirements versus constant rate sources, chart of, 719
- buffering methods
 - input versus output queuing performance, 713–714
 - output buffer overflow probability, 714–716
 - performance of, 713–717
 - shared buffer performance, 716–717
- buffers, role in levels of QoS delivery, 633
- burst error communications channel error model, explanation of, 685
- burstiness
 - meaning of, 122
 - role in source traffic, 700, 705
- BUS (Broadcast and Unknown Server), role in LANE, 514–518
- Busy bit in DQDB ACF, purpose of, 203
-
- C**
- C (channel capacity), dynamics of, 686–687
- C-plane (control plane) in Frame Relay
 - context for, 149–150
 - role in Frame Relay, 138–139
- C/R (Command/Response), role in Frame Relay, 142
- C/W (channel capacity) versus SNR (signal-to-noise ratio), diagram of, 687
- cable modems versus DSL, 294
- cabling between DTE and DCE, dynamics of, 48–49
- CAC (Connection Admission Control)
 - role in PNNI, 427
 - role in traffic contracts and resource management, 635–637
- call attempts, role in traffic modeling, 708–709
- CALL PROCEEDING messages
 - in ATM signaling, 374
 - in N-ISDN D-channel switching, 118
 - in point-to-multipoint connections, 376–377
 - using in Frame Relay SVC operation, 168
- call references, role in UNI 4.1 signaling messages, 372
- CB radios, using as analogy to CSMA/CD, 217
- CBC (Cipher Block Chaining), role in FRPP, 166–167
- CBDS (Connectionless Broadband Data Service) versus SMDS, 206
- CBR (Constant Bit Rate) class of service
 - ATM as end-to-end service, 261
 - in CES, 464
 - in deterministic constant rate performance, 717–720
 - explanation of, 586
- CBR sources, random phases of, 718
- CBS (committed burst size), role in IP traffic conformance, 585
- CC (continuity checks), role in ATM OAM, 806–807
- CCITT/ITU (Comite Consultatif International de Telegraphique et Telephonique)/International Telecommunication Union, development of STM-1 by, 107–108
- CCITT/ITU-T 2.048 Mbps N-ISDN PRI frame structure, diagram of, 117
- CCITT/ITU-T E1-based PRI, explanation of, 116
- CCITT Recommendations
 - I.327 (B-ISDN), 255
 - X.3, X.28, and X.29 (asynchronous DTEs), 124
- CCS (call century seconds), role in call attempts, 709
- CCs (Country Codes) in ATM control plane addresses, assignment of, 380
- CDF (Cutoff Decrease Factor), default value for, 663
- CDMA (Code Division Multiple Access), explanation of, 55
- CDSL (Consumer Digital Subscriber Line), explanation of, 293
- CDV and CTD ATM performance parameters, definition of, 817–818
- CDV (cell delay variation) ATM QoS parameter
 - dynamics of, 307
 - role in ATM traffic contracts, 568–569
 - role in QoS, 573–575
- CDV per switch versus number of connections, chart of, 720
- CDVT (Cell Delay Variation Tolerance), role in ATM traffic descriptors, 579, 581
- CE (consumer edge) devices
 - in Ethernet over MPLS, 522–523
 - in PWE3, 350
 - in VPNs, 546
- Ce equivalent capacity, formula for, 726
- ceiling function, role in undetected error performance of HDLC and AAL5, 694
- cell entry events, role in NP/QoS parameter estimation, 814–815
- cell error ratio ATM performance parameter, definition of, 816
- cell exit events, role in NP/QoS parameter estimation, 814–815
- cell loss ratio ATM performance parameter, definition of, 817
- cell misinsertion rate ATM performance parameter, definition of, 817
- cell sizes, rational for choices of, 278–279
- cells
 - explanation of, 302–306
 - versus frames, 277–280
- centralized versus distributed OAM&P architectures, 764–765
- CER (Cell Error Ratio) ATM QoS parameter, dynamics of, 307
- CER (cell error ratio), role in ATM QoS, 576–577
- CES (Circuit Emulation Service), AAL1-based type of, 463–466
- CES reference model, diagram of, 464
- Cg required channels
 - formula for, 726
 - role in statistical multiplex gain model, 723
- channel cell switching, example of, 301–302
- checksums, performing, 689–690
- child peer groups, role in PNNI, 423
- CI (Congestion Indication) field, role in closed-loop flow control, 658–659
- CID (Channel ID) in AAL2, explanation of, 334–335
- CID values in VoMPLS, range of, 458
- CIDR (Classless Inter-Domain Routing), explanation of, 184
- CIR (committed information rate) in Frame Relay, 144–145
- in IP traffic conformance, 585
- circuit emulation efficiency, dynamics of, 826–828
- circuit emulation over MPLS, explanation of, 466–467
- circuit-switched data services
 - implementation of, 99
 - price of, 98
- circuit switching
 - digital data type of, 98–99
 - history of, 96–99
- circuits in private-line networks, permanent versus switched types of, 103–104
- Cisco's tag switching, role in MPLS, 267–270
- Classes of Internet addresses
 - explanations of, 183–184
 - subnets masks for, 243
- classes of service
 - ABR (Available Bit Rate), 587
 - CBR (Constant Bit Rate), 586
 - Diffserv PHB (per-hop behavior), 594–595
 - GFR (Guaranteed Frame Rate), 588

- ITU-T ATM QoS classes, 588–591
- mapping between ATM Forum and ITU-T QoS definitions, 591–594
- MPLS support for Diffserv, 595–596
- nrt-VBR (non-real-time variable bit rate), 587
- rt-VBR (real-time variable bit rate), 586
- UBR (Unspecified Bit Rate), 587
- Classical IP over ATM
 - explanation of, 533
 - signaling considerations, 535–536
- CLLM (Consolidated Link Layer Management), role in Frame Relay congestion control, 147
- CLNS (Connectionless Network Services), dynamics of, 92–94
- closed-loop congestion control, dynamics of, 645–646, 650
- closed-loop flow control
 - dynamics of, 654–655
 - GFC (Generic Flow Control), 656–657
 - methods of, 655–656
 - rate-based versus credit-based scheme, 659–666
 - role of ABR in, 657–659
- CLP (Cell Loss Priority) bit
 - in ATM conformance, 583
 - in ATM over MPLS network interworking, 349
 - in congestion control, 650
 - in leaky bucket policing, 613
 - purpose of, 304, 305–306
 - in selective discard, 667–668
- CLR (cell loss ratio)
 - in ATM conformance, 583
 - in ATM QoS, 576–577
 - dynamics of, 307
- CMIP command primitives, list of, 780–781
- CMIP (Common Management Interface Protocol), dynamics of, 780–781
- CMR (Cell Misinsertion Rate) ATM QoS parameter, dynamics of, 307, 576–577
- CO-IWF (central office IWF)
 - control plane signaling in, 462
 - role in broadband local loop emulation using AAL2, 460–462
- code bits field in TCP,
 - purpose of, 191
- code division multiplexing,
 - occurrence of, 54, 61–62
- COM (Continuation of Message) in AAL3/4, explanation of, 337
- Common bus, definition of, 42
- communications
 - brief history of, 14–17
 - defining demand for, 17–19
 - residential and commercial users of, 17
- communications channel error models
 - AWGN (Additive White Gaussian Noise), 685
 - BSC (Binary Synchronous Communications), 685
 - burst error channel model, 685–686
 - error performance of common modulation methods, 688–689
- communications channel model, diagram of, 676–677
- communications engineering
 - communications channel model, 676–677
 - deterministic versus random modeling, 677
 - Normal/Gaussian distribution, 678–679
 - probability theory, 677–679
 - pulse shaping, 681
 - QAM (Quadrature Amplitude Modulation), 681–685
 - random trials and Bernoulli processes, 678
- communications networks,
 - randomness in, 677
- complexity analysis
 - hardware and software, 843–844
 - QoS and bandwidth reservation, 844–845
 - signaling, 842–843
 - simple versus complex protocols, 843
- configuration management of networks, explanation of, 765
- conformance
 - achieving, 610–612
 - ATM policing, 613
 - checking, 612–624
 - ensuring with shaping, 624–629
 - GCRA (Generic Cell Rate Algorithm), 619–620
 - leaky bucket policing examples, 613–619
- conforming cell flow, diagram of, 614
- conforming packet flow, diagram of, 621
- congested scenarios, TCP/IP performance in, 743–746
- congestion
 - busy seasons, days, and hours, 643–644
 - examples of, 644–645
 - impact of, 644
 - impact on performance, 646–649
 - nature of, 642–663
- congestion avoidance
 - congestion indication, 653–654
 - connection blocking, 654
 - policing and tagging, 654
 - in TCP, 194
- congestion avoidance, explanation of, 651
- congestion control
 - categories and levels, 651
 - open- and closed-looped solution, 645–646
- congestion control approaches, categorization of, 649–651
- congestion control schemes,
 - measuring effectiveness of, 647–648
- congestion indication, role in congestion avoidance, 653–654
- congestion management
 - network engineering, 652–653
 - resource allocation, 652
- congestion notification in Frame Relay, dynamics of, 146
- congestion recovery procedures
 - disconnection and/or rerouting, 670
 - EPD/PPD (Early/Partial Packet Discard), 668–670
 - operational procedures, 671
 - selective discard, 667–668
 - UPC (Usage Parameter Control), 670
- congestion recovery procedures, operation of, 651
- congestion window values for TCP, calculation of, 195
- CONNECT and CONNECT ACKNOWLEDGE messages
 - in ATM signaling, 374
 - in N-ISDN D-channel switching, 118
 - in point-to-multipoint connections, 377
- CONNECT message, using in Frame Relay SVC operation, 168
- connection blocking, role in congestion avoidance, 654
- connection-oriented protocols,
 - handling of capacity-constrained routing by, 390–391
- connection-oriented versus connectionless services, 94, 840
- connectivity, role in IPPM QoS, 578
- CONS (connection-oriented services) network service paradigm, dynamics of, 91–92, 94
- conservative versus liberal label retention mode, role in MPLS, 392–394
- constraint-based routing, role in MPLS control plane protocols, 390–391
- continuous type of Poisson process, explanation of, 703–704
- control and data forwarding plane architectures, diagram of, 542
- control field formats for HDLC frames
 - differences between, 129
 - explanation of, 127–131
- control packet sequence, diagram of, 133
- control plane protocols
 - diagram of components of MPLS label switching routers, 389
 - purpose of, 388

role in packet voice networking, 446

convergence and integrated access of ATM, future directions and applications of, 853

convergence, explanation of, 211

convergence time, role in link-state routing protocols, 241

Correlation Tag field, role in ATM OAM loopback, 802–803

COT (continuity tone test), role in VoATM trunk signaling, 452

CP (Common Part) layer of AAL, explanation of, 321

CP-IWF (customer premises interworking function), role in broadband local loop emulation using AAL2, 460–462

CPI (Common Part Indicator) in AAL3/4 CPCS sublayer, explanation of, 338

CPS (Common Part Sublayer), role in AAL2, 333–334

CPS PH (CPS Packet Header), role in AAL2, 334–335

CR-LDP (constraint-based routing counterpart LDP), 361

benefits of, 405

role in MPLS, 404–405

crankback procedure, role in AINI, 435–436

crankback signaling and routing, role in PNNI, 430–431

CRC codes

- explanation of, 690–691
- selecting and evaluating performance of, 691–693

CRC (cyclical redundancy check), role in packet switching, 72

CRC field in AAL SAR, explanation of, 324–325

CRC operation, diagram of, 691

credit-based versus rate-based scheme in closed-loop flow control, 659–666

CRM (Cell Rate Margin)

- nonadditive link attributes, role in PNNI, 428

CRM (Count of Missing RM cells), default value for, 663

crosspoint nodal function, using with space division switching, 62–63

CS (Class Selector) PHBs, role in Diffserv, 595

CS (Convergence Sublayer) of AAL, explanation of, 282, 321, 325–326

CS-MUX (Circuit Switching multiplexers), role in FDDI-II, 228

CSFs (call service functions), role in VoATM trunk signaling, 451

CSMA/CD (Carrier Sense Multiple Access with Collision Detection), explanation of, 217–219

CSR (cell switch routers), role in MPLS, 267

CSRC (Contributing Source) identifiers, role in RTP, 197

CSU/DSU (channel service unit/data service unit), role in digital data circuit switching, 98

CTD and CDV ATM performance parameters, definition of, 817–818

CTD (Cell Transfer Delay)

- dynamics of, 307
- role in QoS, 573–575

CVCs (constituent virtual circuits), role in MLFR, 157–158

CWR bit in TCP, purpose of, 191

▼ D

D (data) channels, role in N-ISDN, 114

D notation in queuing system models, meaning of, 705

data and control forwarding plane architectures, diagram of, 542

data circuit-switched services, availability of, 99

data communications and private lines, explanation of, 47–50

data compression, dynamics of, 694–696

data link (L2) layer of OSIRM. *See also* HDLC (High-Level Data Link Control) protocol

- dynamics of, 82–83
- protocol model, 76
- of X-series standards, 123

Data Offset field in TCP, purpose of, 191

data transmission methods, explanation of, 50–53

datagrams in X.25 standard, explanation of, 132

DBR (Domain-Based Rerouting), role in PNNI and AINI, 437

DCE (data communications equipment) and DTE (data terminal equipment) connections

- explanation of, 48–49
- role in physical layer of OSIRM, 81–82

DCP (Data Compression Protocol), role in Frame Relay, 164–165

DDR (data delivery ratio) role in Frame Relay OA&M, 164

DE (Discard Eligibility) bit, role in Frame Relay, 142, 145, 146

defect type and location, role in ATM OAM fault management, 800–801

delay

- impact on applications, 732–734
- role in congestion, 646–649

delay sources within ATM networks, diagram of, 574

delay variation, impact on applications, 738–742

delay versus load performance for input and output queuing, chart of, 713

DES (Data Encryption Standard), role in FRPP, 166–167

destinations, role in ABR, 660

deterministic constant rate performance, dynamics of, 717–720

deterministic versus random modeling

- in communications engineering, 677
- in traffic engineering, 698–699

devices, nodes as, 40

diagnostic usage and loopback operation, role in ATM OAM fault management, 802–806

differentiated UBR

- support for Diffserv, 604–605
- support for IEEE 802.x, 605

digital data circuit switching, explanation of, 98–99

digital samples of analog signals, taking, 97

digital signals and spectra

- explanation of, 679
- telegraph pulse and binary on/off keying, 680–681

digitized voice transmission and switching, explanation of, 97–98

Dijkstra algorithm, usage with link-state routing algorithms, 241

direct mapping of ATM cells, options for, 288

DISCONNECT messages

- role in N-ISDN D-channel switching, 118
- using in Frame Relay SVC operation, 168

disconnection and/or rerouting, role in congestion recovery, 670

discovery, role in LDP protocol, 399

discrete type of Poisson process, explanation of, 703–704

distance vectors, role in routing algorithms, 211

distributed computer communication protocols, explanation of, 20

DIX Ethernet MAC PDU and IEEE 802.3 CSMA/CD frames, diagram of, 219

DLCI (Data Link Connection Identifier) fields, role in Frame Relay, 141–142

DLE (Data Link Escape) character, role in BSC, 51–52

DNS (Domain Name Service), role in TCP/IP, 178

DoS (denial service attacks), advisory when choosing tunnel types, 557–558

downstream DS domains, role in Diffserv traffic contracts, 570

downstream versus unsolicited downstream on demand label distribution, role in MPLS, 392–393

DQDB and SMDS, traffic congestion control aspects of, 204–205

DQDB architecture, diagram of, 203

DQDB (Distributed Queue Dual Bus) protocol
 ACFs in, 201
 origins of, 74
 and SMDS operation, 202–204
 DQDB slot structure, diagram of, 202
 DROP PARTY messages, role in point-to-multipoint connections, 377
 DS BA (behavior aggregate), role in Diffserv traffic contracts, 570
 DS (Diffserv)
 versus ATM, 604
 MPLS support for, 595–596
 PHB (Per Hop Behavior), 594–595
 role in QoS in IP networks, 189–190
 supporting with Differentiated UBR, 604–605
 terminology and reference model, 569
 using with RTP, 197
 DS (digital signal), role in PDH, 104
 DS domains, role in Diffserv traffic contracts, 570
 DS ingress and egress modes, role in Diffserv traffic contracts, 570
 DS nodal traffic function model, diagram of, 612
 DS PHB and DSCP mapping, table of, 594
 DS regions, role Diffserv traffic contracts, 570
 DS0 (digital signal) and DS01 formats, explanations of, 97
 DS1 direct TC mapping of ATM cells, explanation of, 288
 DS3 PLCP ATM interface tables, contents of, 786
 DSCP (Differentiated Services Code Point)
 explanation of, 189
 role in traffic contracts, 569–570
 DSL (digital subscriber line)
 versus cable modems, 294
 explanation of, 50, 292–293
 DSL Forum, purpose of, 30
 DSLAMs (DSL Access Multiplexers), usage of, 294–295
 DSP (Domain Specific Part) in ATM control plane, explanation and diagram of, 379–380
 DSU (data service unit), role in digital data circuit switching, 98
 DTE (data terminal equipment)-to-DCE (data communications equipment) connections
 explanation of, 48–49
 role in physical layer of OSIRM, 81–82
 DTLs (Designated Transit Lists), role in PNNI, 419–420, 429–431
 dual start topology, benefits of, 44
 Dump gremlin, role in leaky bucket policing example, 615–616

DUV (Data Under Voice), role in FDM, 58–59
 DXI (Data Exchange Interface), support for ATM, 489–493
 Dynamic GCRA, dynamics of, 666

▼ E

E-LSP (EXP field-based LSP), role in MPLS support for Diffserv, 595–596
 E1-based PRI, explanation of, 116
 EA (address field extension) bit, role in Frame Relay, 142
 EBS (excess burst size), role in IP traffic conformance, 585
 ECE bit in TCP, purpose of, 191
 ECE (ECN Echo) bit, role in TCP, 195
 ECN (Explicit Congestion Notification) queue management protocol for TCP, explanation of, 195
 EF (Expedited Forwarding) PHB, role in Diffserv, 594
 EFCI (explicit forward congestion indication) bits
 in ATM over MPLS network interworking, 349
 in congestion indication, 653
 efficiency analysis
 of cells versus frames for packet switching, 829–831
 circuit emulation efficiency, 826–828
 IP/ATM, IP/MPLS, and IP/SONET, 831–834
 multiservice efficiency comparison, 835–837
 packet video, 834–835
 packetized voice efficiency, 828–829
 EGP (Exterior Gateway Protocol), explanation of, 238
 ELANs (Emulated LANs)
 explanation of, 511–520
 in LANE, 519–520
 in MPOA, 539, 541
 encapsulation efficiency, average for IP packets, 833
 encapsulation mode, operation of bridges in, 232
 encapsulation protocols
 choosing, 831
 efficiency of IP transport over, 833
 encoding and relaying, recurring trends in, 14–15
 end-to-end paths, explanation of, 209–210
 end-to-end QoS reference model for traffic contracts, explanation of, 567
 end-to-end UBR signaling with user-generated MDCCR, diagram of, 606
 enterprises in VPNs, explanation of, 546

EOB (End of Bus) role in DQDB and SMDS operation, 202
 EPD/PPD (Early/Partial Packet Discard), role in congestion recovery, 668–670
 equivalent capacity
 approximation of, 726–728
 dynamics of, 720–721
 fluid flow approximation, 721–722
 role in ATM CAC, 637
 statistical multiplex gain model, 722–725
 equivalent terminals, role in ATM traffic contracts, 568–569
 ER (Explicit Rate) mode of ABR, role in closed-loop flow control, 659, 661–662, 666
 Erlang-B blocking, chart of, 710
 Erlang-C queuing probability versus offered load, chart of, 712
 Erlang model
 for blocked calls cleared, 709–711
 blocked calls held formula, 711–712
 for call attempts, 708–709
 EROs (explicit route objects), role in RSVP-TE, 402
 error-detecting and -correcting codes, parity check schemes, 689–690
 error models and channel capacity
 AWGN (Additive White Gaussian Noise), 685
 Shannon's channel capacity, 686–687
 error notification messages in RSVP-TE, explanation of, 402
 error performance of common modulation methods, dynamics of, 688–689
 errored outcomes, role in NP/QoS parameter estimation, 815
 ESI (End System Identifier) in ATM control plane, explanation and diagram of, 381–382
 ETAG (Ending Tag) in AAL3/4 CPCS sublayer, explanation of, 338
 Ethernet
 100 Mbps Fast Ethernet, 222–223
 Gigabit and 10 Gbps Ethernet, 224
 Ethernet over MPLS
 internetworking network layer protocols over MPLS, 526–528
 Martini encapsulation of, 520–521
 metropolitan and wide area Ethernet over MPLS, 528–529
 VPLS and access to the Internet, 525–526
 VPLS (Virtual LAN Service), 521–525
 Ethernet protocol
 evolution of, 75
 use with broadcast medium, 43

Ethernet unicast and broadcast over MPLS support in VPLS, diagram of, 523

ETSI (European Telecommunications Standards Institute), purpose of, 30

ETX (End of Text) character, role in BSC, 52

even parity, explanation of, 689

explicit congestion notification, role in packet switching, 73

explicit source routing and topology database update, diagram of, 411

export targets, role in aggregated routing network-based VPNs using tunnels, 553

extended mode, role in HDLC, 128

exterior links, role in PNNI, 421

extranet and intranet connectivity, diagram of, 559

extranets, role in VPNs, 546

▼ F

F (Flag) sequence in HDLC frames, explanation of, 127

F1-F5 notation in ATM OAM reference architecture, meanings of, 796-798

Fast Ethernet, advantages of, 223

FAST (Frame-based ATM over SONET/SDH Transport), dynamics of, 496-497

fast rerouting, role in MPLS, 820

fast retransmit and fast recovery in TCP, advantages of, 195

FATE (Frame-based ATM Transport over Ethernet), dynamics of, 497-498

fault management of networks, explanation of, 765

FC (Frame Control) field in Token Ring MAC PDU, purpose of, 222

FCAPS functional areas for network management, explanations of, 765

FCS (frame check sequence)

- in Frame Relay, 140
- in HDLC frames, 127-128
- in Token Ring MAC PDU, 222

FDDI (Fiber Distributed Data Interface), 224-229

FDDI-II (Hybrid Ring Control), explanation of, 228-229

FDDI-II protocol structure, diagram of, 229

FDDI protocol stack, diagram of, 226

FDDI token and MAC frame formats, diagram of, 227

FDM (frequency division multiplexing)

- example of, 58-60
- explanation of, 54-55

FDR (frame delivery ratio) role in Frame Relay OA&M, 163-164

FEC aggregation and granularity, role in MPLS, 397

FEC (forwarding equivalence class), role in MPLS forwarding operations, 309-310

FEC TLVs, role in LDP, 399

FECN (Forward Explicit Congestion Notification), role in Frame Relay, 142, 146-147

FF (fixed filter) style, role in RSVP-TE, 402

FIB (forwarding information base)

- in IP VPNs over MPLS or IP tunnels, 549
- in MPLS forwarding operations, 309, 388-389

flattening, definition of, 16

flooding protocols, explanation of, 211, 240-241

Flow Label field of IPv6, purpose of, 185-186

flow-oriented switching, dynamics of, 840

fluid flow approximation, role in equivalent capacity, 721-722

FMD (function management data) services, role in SNA, 87

fn clock frequency, explanation of, 331

formal standards bodies versus industry forums, 26

four-fiber rings, usage of, 110

Fourier transform, spectrum of, 680

FPGA (field programmable gate), role in hardware and software complexity analysis, 843-844

FPM (Forward Performance Monitoring) cell function-specific fields in ATM PM, diagram of, 812

FPS (Fast Packet Switching), origins of, 74

FR/ATM ARP, diagram of, 484

FR/ATM-based enterprise network, diagram of, 487

FR/ATM control plane service interworking diagram of, 485

FR/ATM network interworking applying, 486-487

- congestion control and traffic parameter mapping, 480-481
- control plane protocol stacks diagram, 480
- explanation of, 478
- FR-SSCS (Frame Relay Service Specific Convergence Sublayer), 479
- layered model diagram, 478
- scenarios, 477
- status signaling conversion, 480

FR/ATM service interworking ARP (Address Resolution Protocol) interworking, 483-484

- diagram of user plane, 482
- explanation of, 481
- status signaling interworking, 482-483

FR/ATM SVC service interworking, explanation of, 484-486

FR (Frame Relay) LSRs (label switching routers) versus ATM, 318

FR over MPLS network interworking, explanation of, 502

FR-SSCS (Frame Relay Service Specific Convergence Sublayer)

- explanation of, 142
- role in FR/ATM network interworking, 479

FRAG (fragmentation) field, role in ATM over MPLS network interworking, 349

frame-based interfaces supporting ATM, DXI (Data Exchange Interface), 489-493

frame mode bearer service, terms and concepts of, 140-141

Frame Relay

- advantages of, 138
- versus ATM DXI and FUNI, 495-496
- and ATM interworking, 477-478
- control plane, 149
- control protocol networking context, 149-150
- diagram of capabilities on single DLCI, 165
- example of, 143
- fragmentation and compression, 164-166
- frame format of, 140-142
- functions of, 142
- ITU-T and ANSI standards, 151-152
- ITU-T and ANSI status signaling message formats for, 153
- modes of operation for, 152
- networking context for, 139-140
- notation for order of bit transmission in, 141
- operations, administration, and maintenance of, 161-164
- origins of, 74, 137-138
- protocol structure of, 138-139
- and PVC status signaling, 152-155
- role of C-plane in, 138-139
- role of U-plane in, 138-139
- sample virtual private network client/server application diagram, 148
- service aspects of, 147-149
- signaling message information elements, 169-173
- SLAs (service level agreements), 159-161
- standards and specifications, 150-152
- SVCs (Switched Virtual Connections), 168
- traffic and congestion control aspects of, 144-147
- versus X.25 packet switching, 137-138

Frame Relay information-frame format, comparing to ISDN and X.25, 131

Frame Relay PVC status signaling, example of, 155–157

Frame Relay status signaling, diagram of context for, 153

Frame Relay SVC operation, example of, 168–169

frames

- versus cells, 277–280
- role in data link layer of OSIRM, 82

frequency passband, role in binary on/off keying, 681

frequency switching, explanation of, 56

Frequency/wavelength switching, example of, 66–67

FRF (Frame Relay Forum)

- agreements, 29, 150–151
- 11.1 (Voice over Frame Relay), 165
- 12 (fragmentation), 158
- 1.2 (PVC management procedures), 155
- 13 (SLAs), 159
- 16.1 (multiple UNI) versus FRF.15 (minimal interval timers to declare lost frames) in MLFR, 159
- 17 (FRPP), 166
- 19 (OA&M), 159–161
- 20 (IP header compression), 165
- 9 (compressing data frames), 165

FRIHCP (Frame Relay IP Header Control Protocol), explanation of, 166

FRPP (Frame Relay Privacy Protocol)

- explanation of, 166–168
- modes of operation for, 166–167

FRTT (Fixed Round Trip Time), default value for, 663

FS (Frame Status) byte in Token Ring MAC PDU, purpose of, 222

FTD (frame transfer delay), role in Frame Relay OA&M, 163

FTN (FEC-to-NHLFE) of FIBs, purpose of, 311–312

FTN MIBs, usage of, 788–789

FTP (File Transfer Protocol), interface to TCP, 178

full-duplex communications, explanation of, 47–48

full-duplex mode of physical layer of OSIRM, explanation of, 81

full mesh network, example of, 46–47

full-rate allocation of CIR, role in Frame Relay, 145

FUNI (Frame-based User-to-Network Interface) versus ATM DXI and FR, 495–496

- explanation of, 493–496
- versus FAST, 496

FUNI headers, contents of, 495

FUNI PDUs, components of, 495

G

G notation in queuing system models, meaning of, 705

G statistical multiplexing gain, explanation of, 722–724

GAT (Generic Application Transport), role in PNNI and AINI, 436

Gaussian/Normal distribution, role in communications engineering, 678–679

GCAC (Generic Connection Admission Control) algorithm, role in PNNI, 428–429

GCRA (Generic Cell Rate Algorithm)

- role in ATM traffic descriptors, 580
- and virtual scheduling, 619–620
- virtual scheduling and leaky bucket algorithms for, 619

GDMO library, significance or CMIP, 781

generic link layer protocols

- TCP/IP, explanation of, 180–182

GET and GET NEXT messages in SNMP, purpose of, 778

GETBULK protocol messages, role in SNMP, 779

GFC configuration and function, diagram of, 658

GFC (Generic Flow Control)

- in ATM cells, 303–304
- in closed-loop flow control, 656–657

GFR (Guaranteed Frame Rate) class of service

- ATM service category
- optimizations for packet switching, 597–603
- explanation of, 588
- role in ATM as end-to-end service, 261
- switch modifications in support of, 601–603

GFR information, logical unit of, 599

GFR service parameters, table of, 599

GFR.1, dynamics of, 599–600

GFR.2, dynamics of, 600–601

Gigabit and 10 Gbps Ethernet, explanation of, 224

GMPLS (generalized MPLS)

- development of, 792
- future of, 863

Go-Back N retransmission strategy

- role in loss, 736–737
- versus SSCOP, 367

gratuitous ARP, explanation of, 246

group addresses and anycast in ATM control plane, explanation of, 382–383

GSMF (general switch management protocol), development of, 860

GUI (graphical user interface), role in network design and modeling, 753–754

H

H-channels, role in PRIs, 116

half-duplex communications, explanation of, 47–48

half-duplex mode of physical layer of OSIRM, explanation of, 81

Hamming distance, role in parity checking, 689

hard rerouting, role in AINI, 437

hardware and software in emulated LANs, dynamics of, 511–513

HDLC and AAL5, undetected error performance of, 694

HDLC control field lengths and sequence number modulus, table of, 130

HDLC frame formats, explanation of, 127–131

HDLC (High-Level Data Link Control) protocol. *See also* data link (L2) layer of OSIRM

- development of, 74, 126
- efficiency of cells versus frames for packet switching, 829

HDSL (High Data Rate Digital Subscriber Line), explanation of, 50, 293

header checksum in IPv4, purpose of, 182–183

header, role in address or label switching, 64

header values, preassigned reserved type of, 304–305

HEC (Header Error Check)

- functions
- explanation of, 288–290
- role in AAL2, 334
- usage of, 691–693

Hello messages

- in Frame Relay OA&M, 163
- in RSVP-TE, explanation of, 402

Hello protocol

- in link-state routing protocols, 238–240
- in PNNI, 422

hierarchical addressing, disadvantage of, 419

hierarchical tunnels, role in VPNs, 547, 549

HO-DSP (High-Order DSP) in ATM control plane, explanation and diagram of, 381–382

HOB (Head of Bus), role in DQDB and SMDS operation, 202

HOL (head of line) blocking, role in input versus output queuing performance, 713–714

hop-by-hop flow control, role in VS/VD ABR, 662

Hop Limit field of IPv6, purpose of, 185–186

hops, role in frequency/wavelength switching, 66–67

horizontal links, role in PNNI, 421

hosts, definition of, 208

- HRC (Hybrid Ring Control),
 explanation of, 228–229
- HSSI (High Speed Serial Interface),
 speeds of, 52
- HTTP (Hyper Text Transfer
 Protocol), role in TCP/IP, 178
- Hub and spoke topology,
 explanation of, 45
- Hurst parameter, role in output
 buffer overflow probability, 715
-
- I-pictures, usage in MPEG-2, 469
- I-PNNI (Integrated PNNI) model,
 role in MPOA, 538
- I.321 (B-ISDN) ITU-U
 Recommendation, explanation
 of, 255
- I.327 (B-ISDN) CCIT
 Recommendation, explanation
 of, 255
- IAB (Internet Activities Board),
 purpose of, 28
- IANA (Internet Assigned Numbers
 Authority), purpose of, 184
- iBGP, usage of, 405
- ICMP (Internet Control Message
 Protocol), role in TCP/IP,
 177–178
- ICMP PING IP-based management
 tool, using with MPLS, 790–791
- ICP (IMA Control Protocol) cells,
 explanation of, 291
- ICR (Initial Cell Rate), default
 value for, 663
- IDI (Initial Domain Identifier) in
 ATM control plane, explanation
 and diagram of, 381–382
- idle cells, explanation of, 290–291
- IDP (Initial Domain Part) in ATM
 control plane, explanation and
 diagram of, 379–380, 381–382
- IEEE 802.1Q VLAN and Priority
 Tag field in 802.3 Ethernet frame,
 diagram of, 220
- IEEE 802.2 LLC PDU, diagram of, 315
- IEEE 802.3 CSMA/CD and DIX
 Ethernet MAC PDU frames,
 diagram of, 219
- IEEE 802.5 Token Ring MAC frame
 and routing information field,
 diagram of, 234
- IEEE 802.5 Token Ring MAC PDU,
 diagram of, 222
- IEEE 802.X series
 (LAN/MAN/WAN),
 explanation of, 87–89
- IEEE extended LLC/SNAP header,
 diagram of, 216
- IEEE (Institute of Electrical and
 Electronics Engineers) standards
 801.12 (LANs), 223–224
 802.1D and 802.1Q (VLANs),
 219–220
 802.1D (Spanning Tree
 Protocol), 232–233
- 802.2 (LLC sublayer), 214–215
- 802.3 (100 Mbps Fast Ethernet),
 222–223
- 802.3 (MAC), 210
- 802.5 MAC (Token Ring), 220
- 802.5 (Source Routing Protocol),
 233–234
- 802.6 (DQDB), 198–199, 202
- IEEE standards for LANs
 100 Mbps Fast Ethernet, 222–223
 100VG-AnyLAN, 223–224
 Ethernet and CSMA/CD 802.3
 MAC sublayer, 217–219
 Ethernet user priority and
 VLANs, 219–220
 Gigabit and 10 Gbps
 Ethernet, 224
 LLC and MAC sublayer
 implementations, 213–214
 LLC (Logical Link Control)
 layer, 213
 LLC sublayer, 214–215
 MAC sublayer, 215–217
 Token Ring protocol, 220–222
- IEs (information elements) in Frame
 Relay, purpose of, 169–173
- IETF (Internet Engineering Task
 Force) standards body, purpose
 of, 26, 28–29
- IETF multiprotocol label switching,
 role in MPLS, 272–274
- IETF RFCs. *See also* RFCs (Requests
 For Comments)
 1106 (selective
 retransmission), 191
 1122 (version 4 TCP packet
 format), 190
 1142 (IS-IS), 408
 1323 (larger window
 size field), 191
 1700 (well-known ports), 190
 1771 (BGPv4), 405
 1889 (RTP), 196–197
 2001 (Slow Start TCP
 algorithm), 192, 195
 2022 (IP multicast over ATM),
 543–544
 2098 (IP over ATM), 267
 2105 (Cisco's tag switching), 267
 2205 (RSVP), 401, 403
 2211 (Tspec), 188–189
 2212 (Guaranteed QoS), 189
 2225 (Classical IP over ATM), 533
 2236 (IP multicasting), 542
 2283 (BGP multiprotocol
 extensions), 407
 2328 (OSPF), 408
 2370 (LSA for OSPF-TE), 408
 2390 (InARP), 533
 2427 (NLPID/SNAP format for
 FR), 481
 2474 (TOS byte), 189
 2515 (ATM cell layer
 information), 786
 2547 (network-based IP
 VPNs), 550
 2678 (connectivity in IPPM), 578
- 2679 (one-way delay in
 IPPM), 578
- 2680 (one-way packet loss
 metric in IPPM), 578
- 2681 (round-trip delay in
 IPPM), 578
- 2684 (IP), 356, 506
- 2684 (LLC/SNAP format for
 FR), 481
- 2702 (RSVP-TE), 404
- 2917 (virtual routers), 554–556
- 3031 (multiprotocol label
 switching), 272, 309, 391, 397
- 3032 (MTU path discovery over
 MPLS networks), 561
- 3032 (shim headers), 312–313
- 3034 (MPLS over Frame Relay),
 317–318
- 3035 (MPLS over ATM), 315–317
- 3036 (LDP), 397
- 3037 (LDP), 398
- 3107 (BGP), 405, 407
- 3168 (Explicit Congestion
 Notification), 195
- 3209 (RSVP-TE), 401, 404
- 3212 (constraint-based routing
 extensions to LDP), 404
- 768 (UDP), 196
- 793 (version 4 TCP packet
 format), 190
- 826 (ARP), 245
- IFMP (Ipsilon's Flow Management
 Protocol), role in MPLS, 266
- IGMP (Internet Group Management
 Protocol)
 role in IP multicast over ATM,
 542–543
 role in TCP/IP, 178
- IGP (Internet Gateway Protocol)
 explanation of, 238
 role in MPLS control plane, 361
- IGP traffic engineering extensions
 IS-IS TE modifications, 408
 modifications for, 407–408
 open issues and challenges, 409
 OSPF-TE modifications, 408–409
- IISP (Interim Interswitch Signaling
 Protocol), explanation of, 416
- ILM (incoming label map), role in
 MPLS forwarding operations, 311
- ILMI MIBs, dynamics of, 783–786
- ILMI protocol, explanation of,
 383–384
- IMA (Inverse Multiplexing over
 ATM) specification, explanation
 of, 290–292
- IMAC (Isochronous MAC), role in
 FDDI-II, 228
- IMEs (Interface Management
 Entities), role of ILMI in, 783
- implicit congestion notification, role
 in packet switching, 73
- import targets, role in aggregated
 routing network-based VPNs
 using tunnels, 553
- in-rate RM cells, purpose of, 664
- in-service measurement of
 ATM PM, explanation of, 810

- InARP (Inverse Address Resolution Protocol), role in IP over ATM VPNs, 533–534
- incast topology, explanation of, 43–44
- independent versus ordered LSP control, role in MPLS, 394–396
- INE (internetworking network element), explanation of, 348
- INFORM protocol messages, role in SNMP, 779
- information elements, role in ISDN D-channel switching, 117
- information field formats in HDLC frames, explanation of, 127
- information frame, role in HDLC, 128
- information transfer rate, increasing with Shannon's channel capacity, 687
- input versus output queuing performance, role in buffering, 713–714
- interarrival time probability density, role in Poisson and Markov processes, 702–703
- interfaces, role in protocol layers, 75–76
- Internet
 - future growth of, 861–862
 - origins of, 176–177
- internetworking, basic terminology for, 208–210
- interswitch signaling protocols, role in ATM and MPLS control plane, 359–360
- intranet and extranet connectivity, diagram of, 559
- intranets, role in VPNs, 546
- Intserv (Integrated Services) and RSVP (resource reservation protocol), role in QoS in IP networks, 187–189
- IP addressing
 - a.b.* notation for, 310
 - example of, 243–245
 - explanation of, 183–184
- IP and LAN-based applications, diagram of ATM support for, 357
- IP and MPLS policing
 - token bucket algorithm, 623–624
 - token bucket example, 620–623
- IP/ATM internetworks versus MPLS, 273
- IP backbones, traffic engineering in, 409–411
- IP-based management tools for MPLS
 - ICMP PING and Traceroute, 790–791
 - IETF direction for, 792–793
 - vendor-proprietary ICMP extensions for, 791–792
- IP-based tunnels versus MPLS tunnels, 557–558
- IP forwarding, explanation of, 209
- IP HL (Internet Protocol Header Length) field in IPv4, purpose of, 182
- IP hosts, determination of routing by, 242
- IP (Internet Protocol) architecture
 - ATM and MPLS protocol support for, 532
 - dynamics of, 85
 - forwarding and control separation, 860–861
 - future applications and directions of, 855–861
 - generic link layer protocols for, 180–182
 - GMPLS (generalized MPLS) development, 857–860
 - next generation multiservice network infrastructure, 855
 - optical networking for scalability, 855–857
 - role in TCP/IP, 179
- IP multicast over ATM
 - components and operation, 543–545
 - lessons learned from, 544–545
 - overview of, 543–545
- IP multicast over ATM, explanation of, 542–545
- IP networks
 - address design in, 180
 - QoS in, 186–190
 - traffic engineering of, 275–276
- IP over ATM VPNs
 - ATM SVCs, 536
 - ATMARP (ATM Address Resolution Protocol), 534–535
 - Classical IP over ATM, 533
 - Classical IP over ATM signaling considerations, 535–536
 - diagram of, 535
 - explanation of, 533
 - InARP (Inverse Address Resolution Protocol), 533–534
 - interconnecting logical IP subnetworks, 536–537
 - IP multicast over ATM, 542–545
 - MPOA (Multiprotocol over ATM), 537–542
- IP over MPLS architecture and terminology, explanations of, 308–309
- IP packets
 - forwarding, 312
 - versus relative frequency of packet size, 832
- IP path MTU discovery
 - over AAL5, 560–561
 - over MPLS, 561–562
- IP PF (Precedence Forwarding) PHB, role in Diffserv, 595
- IP QoS versus ATM traffic parameters, 596–597
- IP routers
 - diagram of connection via ATM switches, 263
 - explanation of, 209
- IP subnets, functionality of, 208
- IP suite, diagram of, 177
- IP switching, role in MPLS, 265–266
- IP traffic conformance, explanation of, 585
- IP traffic descriptors, explanation of, 582
- IP transport over encapsulation protocols, efficiency of, 833
- IP VPN over MPLS or IP tunnels
 - network-based concepts of, 548–550
 - terminology and concepts, 545–548
- IPng (IP next generation), explanation of, 184–185
- IPPM (IP Performance Metrics), role in QoS, 578
- Ipsilon's IP switching, role in MPLS, 265–266
- IPv4 packet datagram format, explanation of, 182–183
- IPv6, enhancements over IPv4, 184–185
- IS-IS (Intermediate System to Intermediate System) routing protocol, explanation of, 241
- IS-IS TE, modifications to, 408
- ISDN D-channel switching, dynamics of, 117–118
- ISDN (Integrated Services Digital Network)
 - comparing to X.25 and Frame Relay information-frame formats, 131
 - dynamics of, 89–91
 - multiple layered protocol planes in, 90
 - user, control, and management plane protocols in, 90
- ISO NSAP-based ATM AESA formats, diagram of, 381–382
- ISR (integrated switch router), role in ARIS, 272
- ISs (intermediate systems), role in ATM interface, 258
- ITU-T and ANSI standards for Frame Relay, explanation of, 151–152
- ITU-T ATM QoS classes
 - QoS in international ATM networks, 590–591
 - specified QoS class, 589–590
 - unspecified QoS class, 590
- ITU-T B-ISDN signaling protocols, role in ATM control plane, 362–363
- ITU-T/CCITT Recommendations
 - E.164 (numbering plan for identifying interfaces), 168
 - Q.921 (ISDN LAP-D and LAP-F), 138
 - X.121 (numbering plan for identifying interfaces), 168
- ITU-T (International Telecommunications Union-Telecommunications) standards body, purpose of, 26

- ITU-T Recommendations
- E.164 (numbering plan for identifying interfaces), 379–380
 - G.805 (generic transport architecture applied to ATM and SDH), 770
 - H.222 (MPEG-2), 468
 - I.1113 (NNI), 296
 - I.2610 (cause code), 372
 - I.321 (B-ISDN), 254, 290
 - I.356 (QoS class objectives), 591
 - I.361 (header field values for UNI), 304
 - I.362 (AAL functions), 320
 - I.363 (AAL), 320–323
 - I.363.1 (AAL1 protocol), 323
 - I.363.2 (AAL2), 332
 - I.363.3 (AAL3/4), 337
 - I.366.2 (AAL2 SSCS), 452–453
 - I.371 (ATM transfer capabilities), 308
 - I.555 (FR/ATM interworking), 477
 - J.82 (MPEG-S), 470
 - Q.1901 (BICC) protocol, 450
 - Q.2100 (SAAL), 365
 - Q.2111 (SSCOP extensions), 368
 - Q.2931 (B-ISDN), 139, 150, 360, 362, 368
 - Q.2951 (UUS), 369
 - Q.2959 (Call Processing Priority IE), 439
 - Q.922 (LAP-F), 140, 168
 - Q.931 (ISDN), 139, 150
 - Q.933 (Frame Relay), 139, 150–151, 153–154, 168, 169, 362
 - (SSCOP), 366
 - Y.1310 (CR-LDP), 405
 - Y.1710 (MPLS forwarding components), 771
- ITU-T signaling standards, diagram of, 363
- ITU-T TINA network management architecture, dynamics of, 769–771
- ITU-T UNI and NNI signaling standards, mapping, 434
- ITU TMN, dynamics of, 766–769
- IWF (internetworking function) in N-ISDN, role in ATM control plane, 362–363
- IXC (interexchange carrier), role in interfaces for switched services, 99
-
- L**
- L-LSP (label-based LSP), role in MPLS support for Diffserv, 595–596
- L1 (physical layer) of OSIRM, dynamics of, 81–82
- L2 (data link layer) of OSIRM, dynamics of, 82–83
- L2TPv3 tunneling protocol, development of, 347
- L3 (network layer) of OSIRM, dynamics of, 83
- L4 (transport layer) of OSIRM, dynamics of, 83
- L5 (session layer) of OSIRM, dynamics of, 84
- L6 (presentation layer) of OSIRM, dynamics of, 84
- L7 (application layer) of OSIRM, dynamics of, 84
- label mapping messages, role in LDP, 399
- label multiplexing, explanation of, 55, 60–61
- label release messages, role in LDP, 399
- label requests
 - in LDP, 399
 - in RSVP-TE, 401
- label stacked MPLS LSPs and ATM VPs, dynamics of, 638–639
- label stacking, explanation of, 314–315
- label swapping
 - occurrence of, 54
 - role in tag switching, 268
- label switching
 - example of, 64–65
 - occurrence of, 54
- label TLVs, role in LDP, 399
- label withdraw messages, role in LDP, 399
- LAN and internetworking terminology, diagram of, 209
- LAN and IP-based applications, diagram of ATM support for, 357
- LAN protocol standards layered model, diagram of, 213
- LAN standards by IEEE
 - 100 Mbps Fast Ethernet, 222–223
 - 100VG-AnyLAN, 223–224
 - Ethernet and CSMA/CD 802.3
 - MAC sublayer, 217–219
 - Ethernet user priority and VLANs, 219–220
 - Gigabit and 10 Gbps Ethernet, 224
 - LLC and MAC sublayer implementations, 213–214
 - LLC (Logical Link Control) layer, 213
 - LLC sublayer, 214–215
 - MAC sublayer, 215–217
- LANE (LAN Emulation)
 - components and connection types, 514
 - diagram of components and interconnections, 515
 - emulating broadcast medium with, 519
 - explanation of, 511–520
 - implementation considerations, 519–520
 - and spanning tree, 518
- LANE operation, summary of, 514–518
- LANE protocol data flows, explanation and diagram of, 512–513
- LANs (local area networks)
 - origins of, 15–16
 - role of multipoint topology in, 42
- LAP-B frame
 - role in HDLC, 128–129
 - and X.25 packet layer payload diagram, 132
- LAP-B protocol, usage of store-and-forward approach by, 133–134
- LAP-F (Link Access Procedure), explanation of, 140
- LAP (Link Access Procedure) protocols, comparison of, 130–131
- latency/bandwidth crossover point, role in delay, 734
- latency-limited application, occurrence of, 732–733
- layered data communication architectures
 - IEEE 802.X series
 - (LAN/MAN/WAN), 87–89
 - IP (Internet Protocol), 85
 - ISDN (Integrated Services Digital Network), 89–91
 - SNA (Systems Network Architecture), 86–87
- layered protocol model, diagrams of, 79–80
- LCNs (logical channel numbers), assignment of, 132
- LCP (Link Control Protocol) of PPP, role in TCP/IP, 180
- LDP discovery messages, exchange of, 398
- LDP identifiers, components of, 399
- LDP (label distribution protocol)
 - downstream on demand
 - independent control, 399–400
 - dynamics of, 397–400
 - label stacking-based application
 - example, 411–412
 - role in MPLS control plane, 361
- LDP MIBs, usage of, 788
- LDP peers, explanation of, 398
- LE Topology Request messages, LANE support for, 518
- leaf, definition of, 40
- leaky bucket buffering, role in conformance and shaping, 626–627
- leaky bucket configurations, diagram of, 584
- leaky bucket policing
 - confirming cell flow, 614
 - examples of, 613–619
 - nonconforming cell flow, 615
 - sliding and jumping window
 - policing, 616–619
- Leaky Buckets versus Token Buckets, 596
- leased lines, characteristics of, 100

LECS (LAN Emulation Configuration Server), role in LANE, 514–516

LECs (local exchange carriers) in interfaces for switched services, 99
in LANE, 514–516
in LANE implementation, 519
in MPOA, 539

LERs (label edge routers), role in MPLS, 264, 308–309, 388–389

LES (LAN Emulation Server), role in LANE, 514–516

LFIB (label forwarding information base)
in IP VPNs over MPLS or IP tunnels, 550
in LDP, 399
in MPLS forwarding operations, 309, 311, 388–389

LGNs (logical group nodes), role in PNNI, 423, 426

LI (Length Indicator) field
in AAL2, 334–335
in AAL3/4, 338

LIB (label information base)
in LDP, 399
in MPLS forwarding operations, 309

liberal versus conservative label retention mode, role in MPLS, 392–394

LIJ (Leaf Initiated Join) protocol, role in ATM signaling, 376

link coloring
in IGP traffic engineering, 407
in IS-IS TE, 408
in RSVP-TE, 404

link layer protocols, MPLS support for, 498–503

link speed, effect on packet performance, 277–280

link-state advertisement methods, usage with distance vectors, 212

link-state routing protocols, explanation of, 238–241. *See also* routing protocols

links as transmission paths, dynamics of, 40

LISs (Logical IP Subnetworks) explanation of, 242–245
interconnecting, 536–537
role in IP over ATM VPNs, 533

LLC and MAC physical interface points, diagram of, 214

LLC encapsulation
for bridged protocols, 508–509
diagram of, 507
for routed protocols, 507–509
versus VC multiplexing, 510–511

LLC headers, contents of, 507

LLC (Logical Link Control) sublayer in FDDI-II, 228
in FDDI protocol stack, 226
in IEEE 802.X series, 88
role in bridging context, 230

role in data link layer of OSIRM, 82

LMI (local management interface) standard for Frame Relay development of, 153
role in ATM DXI, 492–493

logical access
ATM support for, 476
attributes for, 476
diagram of, 475

logical links, role in PNNI, 421

logical topology, definition of, 40–41

loopback in ATM OAM
verification/problem diagnosis, diagram of, 805

Loopback Location ID field, role in ATM OAM, 803

loopback operation and diagnostic usage, role in ATM OAM fault management, 802–806

loopback primitives in ATM OAM, diagram of, 804

loosely explicitly routed paths in MPLS, explanation of, 392

loss
impact on applications, 735–738
role in congestion, 647

LSAs (link state advertisements), role in OSPF-TE and IGP, 408

LSP (label switched path), role in ATM and MPLS, 4–5

LSP (labeled switch path)
in ATM over MPLS network interworking, 349
in CONS, 92
in IS-IS TE, 408

LSP tunnels, role in MPLS signaling and routing protocols, 391–392

LSRs in MPLS, diagram of, 275

LSRs (label switching routers) and related MIBs, 787–788
role in MPLS, 263–264, 308–309, 388–389, 391–392
role in MTU path discovery over MPLS, 561–562

LUs (logical units), role in SNA, 87

▼ M

M-* CMIP command primitives, list of, 780–781

M/D/1 and M/M/1 queuing systems, diagram of, 706–707

M/M/1/B queuing system, example of, 714

M/M/1 queuing theory, role in delay, 734

M (maximum packet size) parameter, role in IP traffic descriptors, 582

m (minimum-policed unit) parameter, role in IP traffic descriptors, 582

M (More) bit, role in packet layer, 132

M4 security and logical MIB, ATM Forum requirements documents for, 787

MAC frame and FDDI token formats, diagram of, 227

MAC (Media Access Control) sublayer
in bridging context, 230
in data link layer of OSIRM, 82
in FDDI protocol stack, 226
in IEEE 802.X series, 88

MAC sublayer standards, attributes of, 216

MANs (metropolitan area networks)
development of, 198
role of point-to-point topology in, 41

Markov processes and Poisson arrivals, role in queuing theory, 702–705

MARS (Multicast Address Resolution Server), role in IP multicast over ATM, 544

Martini encapsulation
of Ethernet over MPLS, 520–521
explanation of, 351–352
and transport of FR, AAL5, HDLC, and ATM over MPLS, 500–501

MBS (maximum burst size)
in ATM traffic descriptors, 579
in GFR traffic contract, 598

MCDV (Maximum Cell Delay Variation) additive link attributes, role in PNNI, 428

MCLR (Maximum Cell Loss Ratio) additive link attributes, role in PNNI, 428

MCR (Minimum Cell Rate), default value of, 663–664

MCR (minimum cell rate), role in GFR traffic contract, 598

MCS (multicast server), role in IP multicast over ATM, 543

MCSN (Monitoring Cell Sequence Number) field in PM, meaning of, 812

MCTD (Maximum Cell Transfer Delay) additive link attributes, role in PNNI, 428

MDCR (minimum desired cell rate) in UBR and BSC, 603–604
parameters used with UBR class of service, 605–607

mean cell transfer delay, role in ATM performance, 818

measurement skew, role in leaky bucket policing, 618

MEGACO (media gateway protocol), role in packet voice networking, 445

memoryless process, Poisson arrivals as, 703

Mesh topology, explanation of, 46–47

metro and wide area Ethernet over MPLS, diagram of, 528

MFS (maximum frame size), role in GFR traffic contract, 598, 600

- MGCs (media gateway controllers)
 - in VoATM trunk signaling, 450–451
 - in VoPackets, 449
- MGs (media gateways), role in VoPackets, 449
- MIBs (Management Information Bases), role in SNMP, 777–778. *See also* ATM MIBs and MPLS MIBs
- microflows, role in Diffserv, 189
- MICs (Medium Interface Connectors) in FDDI protocol stack, purpose of, 226
- MID (Multiplex Identification) field in AAL3/4, explanation of, 337–338
- MLP (Multilink Procedure), role in HDLC, 130
- MMPP (Markov Modulated Poisson Process), dynamics of, 703–705
- modernization of transmission infrastructures, explanation of, 20
- Moore's law, explanation of, 19–20, 21–22
- Moore's Law, role in scalability analysis, 839–840
- MPEG-2
 - over AAL1 mapping, 471
 - SPTS over AAL5 mapping, 470
 - video and audio encoding and decoding, 469
 - video over ATM and packet networks, 468–471
- MPEG (Motion Photographic Experts Group) video coding standard, explanation of, 467–468
- MPLS admission control, dynamics of, 638
- MPLS and ATM control plane protocols
 - generic control plane functions, 359
 - switched and permanent ATM virtual connections, 359
- MPLS and ATM protocol stack, diagram of, 274
- MPLS control plane protocols
 - architecture of, 388–397
 - constraint-based routing, 389–391
 - explanation and diagram of, 361–362
 - label distribution control protocol attributes, 391–397
- MPLS encapsulation over POS and Ethernet, diagram of, 316
- MPLS encapsulation standards, MPLS shim header, 312–315
- MPLS Forum, purpose of, 29–30
- MPLS forwarding of IP packets, example of, 312
- MPLS forwarding plane, diagram of, 311
- MPLS in IP networks
 - connectivity across multiple providers, 412–413
 - label distribution in support of other services, 411–412
 - traffic engineering in IP backbone example, 409–411
- MPLS label distribution control protocol attributes
 - conservative versus liberal label retention mode, 392–394
 - FEC aggregation and granularity, 397
 - hop by hop versus explicit routing LSP tunnels, 391–392
 - merging versus nonmerging LSRs, 397
 - ordered versus independent LSP control, 394–396
 - unsolicited downstream versus downstream on demand, 392–393
- MPLS label distribution modes by protocols, diagram of, 398
- MPLS label distribution signaling protocols
 - BGP (Border Gateway Protocol), 405–407
 - CR-LDP (constraint-based routing counterpart LDP), 404–405
 - LDP (label distribution protocol), 397–400
 - RSVP-TE, 400–404
- MPLS label-switching router and label edge router, diagram of, 310
- MPLS LSRs, diagram of, 275
- MPLS management, IETF direction for, 792–793
- MPLS MIBs. *See also* ATM MIBs, MIBs (Management Information Bases)
 - LSR (label switching router) type, 788–789
 - multiservice PPVPN and PWE3, 789–790
 - TE (traffic engineering) type, 789
- MPLS (Multi Protocol Label Switching)
 - architecture of, 308–312
 - and ARIS (Aggregate Route-Based IP Switching), 270–272
 - basics of, 274–275
 - circuit emulation over, 466–467
 - and Cisco's tag switching, 267–270
 - control and forwarding plane model, 388–389
 - and CSR, 267
 - and early IETF multiprotocol label switching, 272–274
 - forwarding and control separation, 860–861
 - forwarding-component management, 771
 - forwarding operations, 309–312
 - future applications and directions of, 855–861
 - GMPLS (generalized MPLS) development, 857–860
 - infrastructure of, 4–6
 - versus IP and IP/ATM internetworks, 273
 - IP-based management tools for, 790–791
 - and Ipsilon's IP switching, 265–266
 - label stack operation, 314
 - multiservice network potential of, 862–863
 - next generation multiservice network infrastructure, 855
 - optical networking for scalability, 855–857
 - origins of, 263–274
 - over ATM, 315–317
 - over Frame Relay, 317–318
 - support for Diffserv, 595–596
 - support for link layer protocols, 498–503
 - and traffic engineering of IP networks, 275–276
 - vendor-proprietary ICMP extensions for, 791–792
 - voice trunking, 449–450
- MPLS OAM status and direction overview of, 819–820
- protection switching and fast rerouting, 820
- MPLS over FR encapsulation formats, diagram of, 318
- MPLS PWE3 support for voice, video, and WAN data, diagram of, 356
- MPLS support of IP, optimization of, 312
- MPLS tunneling
 - versus IP-based tunneling, 557–558
 - multiservice type of, 276–277
 - using network-based IP VPNs with, 276
- MPOA (Multiprotocol over ATM) lessons learned from, 541–542
- network components, 539
- overview of, 537–542
- server and client usage of emulated LANs, 541
- virtual routers in, 538–539
- MRCMS (Multirate Circuit-Mode Bearer Service), explanation of, 99
- Mrm, default value for, 663
- MTTR (mean time to repair), role in private-line networks, 101
- MTU (Maximum Transfer Unit), role in MPLS shim headers, 314
- MTU path discovery
 - over AAL5, 560–561
 - over MPLS, 561–562
- multicast-capable subnets, explanation of, 208
- multilayered model of ITU TMN, diagram of, 768
- Multilink Frame Relay, explanation of, 157–159
- multiplexer, explanation of, 57–58
- multiplexing
 - definition of, 53
 - examples of, 57–62

methods of, 54, 510–511
 rates of, 97
 role in traffic engineering, 701
 usage of PDH in, 105
 multiplexing voice conversations
 statistically, role in TCP
 performance, 746–747
 multipoint connections in physical
 layer of OSIRM, explanation of, 81
 multipoint-to-point forwarding, role
 in ARIS and MPLS, 271
 multipoint-to-point topology,
 explanation of, 43–44
 multipoint topology, explanation of,
 42–44
 multiprotocol encapsulation over
 AAL5, explanation of, 506–511
 multiprotocol label switching, role
 in MPLS, 272–274
 multiservice backbone network
 infrastructure of ATM, future
 directions and applications for,
 852–853
 multiservice networking, future
 possibilities for, 861–863
 multiservice tunneling over MPLS,
 dynamics of, 276–277, 346
 multistrand cabling, role in
 DTE-to-DCE connections, 49

▼ N

N in Bernoulli process,
 meaning of, 678
 N-ISDN B-channel and D-channel
 services, diagram of, 115
 N-ISDN bearer connections,
 establishment and release of, 117
 N-ISDN call establishment, data
 transfer, and release phases
 diagram, 118
 N-ISDN (Narrowband-ISDN)
 and B-ISDN, 255–256
 basics and history of, 113–118
 protocol layers in, 89–90
 reference configuration, 114
 in VoATM trunk signaling, 451
 role of IWF in ATM control
 plane, 362–363
 N-ISDN PRI, explanation of, 116
 N times, role in SONET STSs, 107
 NANPs (North American
 Numbering Plans) in ATM
 control plane addresses,
 explanation of, 380
 NAU (network accessible unit)
 services, role in SNA, 87
 NBMA (nonbroadcast multiple
 access) subnet, explanation of, 209
 NCCI (Network Call Correlation
 Identifier), role in PNNI and
 AINI, 436
 NCP (Network Control Protocol) of
 PPP, explanation of, 180–181
 NDCs (National Destination Codes)
 in ATM control plane addresses,
 explanation of, 380

network addressing philosophies,
 types of, 418–419
 network administration, OAM&P
 philosophy of, 760–761
 network-based IP VPNs,
 considerations and trade-offs
 with, 556–557
 network costs, explanation of, 211
 network design and modeling tools
 design scenario
 specification, 754
 displaying and comparing
 results, 755
 GUI (graphical user interface),
 753–754
 modeling network-specific
 capabilities, 755
 network design quadrants, diagram
 of, 752
 network engineering, role in
 congestion management, 652–653
 network interworking, explanation
 of, 348–351
 network layer (L3) of OSIRM,
 dynamics of, 83
 network layer, protocol model of, 76
 network maintenance, OAM&P
 philosophy of, 762
 network management architectures
 ATM Forum, 772
 centralized versus distributed
 model, 764–765
 ITU-T TINA, 769–771
 ITU TMN, 766–769
 OSI functional model, 765–766
 network management protocols
 BSC (Binary Synchronous
 Communications), 51
 choosing, 782
 CMIP (Common Management
 Interface Protocol), 780–781
 computing relative efficiency of, 833
 explanation of, 70
 for multipoint and broadcast
 topologies, 42
 proprietary types of, 781–782
 for ring topology, 45
 SNMP (Simple Network
 Management Protocol),
 776–779
 network operations, OAM&P
 philosophy of, 762
 network planning and design
 process
 analyzing and simulating
 candidate networks and
 technology, 751
 approaches and modeling
 philosophy, 749–750
 guidelines for, 752
 measuring traffic and
 performance data, 750–751
 network provisioning, OAM&P
 philosophy of, 761–762
 network service paradigms
 CLNS (Connectionless Network
 Services), 92–94

connection-oriented versus
 connectionless services, 94
 CONS (connection-oriented
 services), 91–92, 94
 network service providers, role in
 creating standards, 31
 network topologies, types of, 40–47
 networking
 geographical aspects of, 18
 origins of, 15–16
 networks
 address assignment and
 resolution in, 210–211
 explanation of, 209
 reconfiguration of, 211–212
 restoration in, 211–212
 routing in, 211–212
 Next Header field of IPv6, purpose
 of, 185–186
 NEXT-HOP path for NLRI
 information, role in BGP, 406
 NHLFE (next hop label forwarding
 entry), explanation of, 310–311
 NICs (network interface cards),
 using with emulated LANs, 512
 NLRI (Network Layer Reachability
 Information), role in BGP, 406
 NNI and UNI, diagram of, 259
 NNI (Network-to-Network
 Interface) signaling protocol
 explanation of, 296–297
 role in ATM and MPLS control
 plane, 359–360
 nodes
 connecting in dual star
 topology, 44
 connecting in mesh topology, 46
 connecting in multipoint
 topology, 42
 connecting in point-to-point
 topology, 41–42
 connecting in ring topology, 45
 definition of, 40
 purpose of, 4
 role in CLNS, 92–93
 role in frequency/wavelength
 switching, 66–67
 nonadditive link attributes, role in
 PNNI, 428
 nonassociated signaling channel
 configuration, role in ATM
 control plane protocols, 363
 nonbursty sources, explanation
 of, 700
 nonconforming cell flow, diagram
 of, 615
 nonconforming packet flow,
 diagram of, 622
 nonpreemptive prioritized queuing,
 explanation of, 278
 Normal/Gaussian distribution, role
 in communications engineering,
 678–679
 notifications, role in LDP
 protocol, 398
 Np in Normal/Gaussian
 distribution, meaning of, 678

NP/QoS parameter estimation, role in ATM performance
specification and measurement, 814–819

NPC (Network Parameter Control), role in ATM policing, 613

Nrm, default value for, 663

NRM (normal response mode), role in HDLC, 126

nrt-VBR (non-real-time variable bit rate), explanation of, 587

NSAP (Network Service Access Point) format for ATM
addressing, explanation of, 379

NSNs (Nationally Significant Numbers) in ATM control plane addresses, explanation of, 380

NT (network terminal points)
in ISDN, 89
in N-ISDN, 114

NTSC (National Television Standards Committee) video coding standard, explanation of, 467

nx64 Kbps bearer service, role in BRI and PRI, 116

nxDS0
explanation of, 106
role in BRI and PRI, 116–117

Nyquist sampling theorem, explanation of, 97

0

OA&M (Operations, Administration, and Maintenance) of Frame Relay, explanation of, 161–164

OAM&P architectures, centralized versus distributed types of, 764–765

OAM&P (Operations, Administration, Maintenance, and Provisioning), explanation of, 760

OAM&P philosophy
administration, 760–761
ATM challenges to, 763
maintenance, 762
MPLS challenges to, 763
operations, 762
provisioning, 761–762

OAM&P process flow, diagram of, 761

object model of network management, SNMP as, 776–777

OC-N (optical carriers) in SONET, explanation of, 107–108

OIDs (Object Identifiers), role in SNMP, 778

one-way delay, role in IPPM QoS, 578

one-way delay variation, role in IPPM QoS, 578

one-way packet loss metric, role in IPPM QoS, 578

OOK (on/off keying) signals
dynamics of, 680

role in pulse shaping, 680
role in QAM, 683

open-loop congestion control, dynamics of, 645–646, 650

operations systems functions, role in TMN, 768

optical fiber, transparency of, 59

optical multiplexing and switching systems, example of, 66–67

optimistic queuing models, explanation of, 707

ordered label distribution, diagram of, 396

ordered versus independent LSP control, role in MPLS, 394–396

ORIGIN path attribute for NLRI information, role in BGP, 406

OSF (Offset Field), role in AAL2, 335

OSI layers
mapping B-ISDN layers and sublayers to, 284
mapping generic devices to, 84

OSIRM (Open Systems Interconnection Reference Model)
application layer (L7), 84
data link layer (L2), 82–83
diagram of, 78
explanation of, 77–81
network layer (L3), 83
physical layer (L1), 81–82
presentation layer (L6), 84
role of LLC in, 82
role of MAC in, 82
session layer (L5), 84
transport layer (L4), 83
versus X.25 packet switching, 123–124

OSPF (Open Shortest Path First) routing protocol, explanation of, 241

OSPF-TE modifications, role in IGP, 408–409

out-of-rate RM cells, purpose of, 664

outer labels, role in VoMPLS trunking, 458

output buffer overflow probability, dynamics of, 714–716

outside links, role in PNNI, 421

overflow probability objectives, requirement of, 715

oversubscription of CIR, role in Frame Relay, 145–146

overtrunking ration, role in Erlang model for blocked calls cleared, 711

P

p notation in queuing system models, meaning of, 705

p (peak rate) parameter, role in IP traffic descriptors, 582

P-pictures, usage in MPEG-2, 469

P (provider) devices, role in VPNs, 546

packet classifiers in RSVP, purpose of, 187

packet forwarding
explanation of, 235–237
role in scalability analysis, 839–840

packet layer format and protocol, explanations of, 131–132

packet layer of X-series standards, explanation of, 123

packet layer, receive and send sequence numbers used by, 132

packet networks and VoATM, explanation of, 467–471

packet-oriented traffic parameters, graph of, 582

packet performance, effect of link speed on, 277–280

packet schedulers, role in RSVP, 187

packet size
versus protocol efficiency, 830
versus relative frequency of IP packets, 832

packet switching
early reasons for, 71
explanation of, 122
genealogy of, 74–75
history of, 70–75
principles of, 71–73

packet video efficiency analysis, dynamics of, 834–835

packet voice networking
control plane signaling protocols in, 446
encoding standards, 446–447
explanation of, 444–445
general network architecture, 445–446
media gateway functions, 446–447
quality considerations, 448–449

packetized voice efficiency, dynamics of, 828–829

packets, role in address or label switching, 65

PAD character, role in BSC, 51–52

PAD field
in AAL2, 336
in AAL3/4, 338–339
in AAL5, 342, 343
in AAL5 multiplexing, 344–345

parallelogram symbol, role in depicting multiplexing, 57–58

parent peer groups, role in PNNI, 423

parity check schemes, dynamics of, 689–690

Path messages in RSVP-TE, explanation of, 401

path traces, role in PNNI and AINI, 436–437

path-vector routing paradigm, disadvantage of, 413

path vector routing protocols, usage of, 212

Payload Length field of IPv6, purpose of, 185–186

- payload, role in address or label switching, 64
- PBS (peak burst size), role in IP traffic conformance, 585
- PBXs (private branch exchanges), role in leased lines, 100–101
- PCM (pulse code modulation), role in TDM, 55, 97
- PCR (peak cell rate)
 - default value of, 663
 - in ATM traffic descriptors, 579–580
 - in GFR traffic contract, 598
- PDB (per-domain behavior), role in Diffserv, 570
- PDH (Plesiochronous Digital Hierarchy)
 - in VoMPLS, 104–106
 - versus SONET/SDH, 107
- PDLs (PAD length fields), role in VoMPLS, 458–459
- PDU formats for SMDS/802.6 protocol, explanation of, 199–202
- PDUs (protocol data units), role in protocol layers, 76–77, 79–80
- PE (provider edge) devices
 - role in PWE3, 350–351
 - role in VPNs, 546
- PE routers, role in aggregated routing network-based VPNs using tunnels, 552
- peak and average rates in voice conversations, dynamics of, 700
- peak rate connection admission control, diagram of, 636
- peak rate shaper, example of, 629
- peak-to-peak CDV, explanation of, 576–577
- performance and traffic data, measuring, 750–751
- performance management of networks, explanation of, 765
- PES (Packetized Elementary Stream), role in MPEG-2, 468
- pessimistic queuing models, explanation of, 707
- PGs (peer groups), role in PNNI, 421
- phasing in TCP, occurrence of, 195
- PHB (Per Hop Behavior)
 - role in classes of service, 594–595
 - role in Diffserv, 189–190, 570
- PHY (PHYSical) protocol layer in FDDI protocol stack
 - in ATM, 285–295
 - purpose of, 226
 - role in IEEE 802.X series, 87–88
- physical circuits, diagram of, 298
- physical layer (L1)
 - of OSIRM, 81–82
 - sublayers of, 282
 - of X-series standards, 123
- Physical Layer MIB in ILMI, contents of, 784
- physical layer, protocol model of, 76
 - Physical Medium-Dependent sublayer of ATM physical layer, explanation of, 285–287
 - physical star topology, usage of, 44–45
 - physical topology, definition of, 40
 - PING IP-based management tool, using with MPLS, 790–791
 - pipe model, role in network-based IP VPNs, 548
 - pipes, role in VP and VC switching and cross-connection, 300
 - PIR (peak information rate), role in IP traffic conformance, 585
 - playback buffers, overrun and underrun scenarios, 738–740
 - PLCP (Physical Layer Convergence Protocol), advisory about, 290
 - PLP (Packet Layer Protocol) X-series standard, explanation of, 123, 131–132
 - PM (performance measurement), role in ATM, 810–814
 - PM procedure, explanation of, 812–814
 - PMD (Physical Medium Dependent) sublayer of FDDI protocol stack, 226
 - of physical layer, 282, 285–287
 - PNNI and AINI routing and signaling capabilities
 - DBR (Domain-Based Rerouting), 437
 - GAT and NCCI, 436
 - path and connection trace, 436–437
 - PNNI hierarchy
 - diagram of, 423
 - example of, 425–427
 - PNNI (Private Network-to-Network Interface) protocol
 - versus AINI, 434
 - architecture and requirements, 417–418
 - congestion of resources, 438
 - crankback procedure, 430–431
 - DTLs (Designated Transit Lists), 429–431
 - dynamic building of hierarchy, 422–424
 - explanation of, 241, 416–417
 - explicitly routed calls, 438
 - functionality of, 417–418
 - GCAC (Generic Connection Admission Control)
 - algorithm, 428–429
 - link parameters, 428
 - lowest hierarchical level, 421–422
 - minimum operable subset, 432–433
 - network addressing philosophy, 418–419
 - quality and bandwidth, 427–431
 - reachability and scope, 427
 - routing hierarchy and topology aggregation, 420–427
 - security services IE, 438
 - signaling versus routing, 419–420
 - source route estimates and refinements, 428
 - SPVCs (switched PVCs), 431–432
 - terminology, 421–422
 - topology aggregation and complex node representation, 424–425
 - point-to-multipoint
 - ABR (Available Bit Rate), 666–667
 - call procedures in UNI 4.1 signaling messages, 372
 - SVC (Switched Virtual Connection), 378
 - switching functions, 56–57
 - VCs in IP multicast over ATM, 543
 - point-to-point connections
 - in ATM signaling, 374–375
 - in physical layer of OSIRM, 81
 - point-to-point topology
 - explanation of, 41–42
 - switching functions of, 56
 - Poisson arrivals and Markov processes, role in queuing theory, 702–705
 - policing
 - definition of, 613
 - role in priority discard thresholds, 631
 - and tagging in congestion avoidance, 654
 - POS (Packet over SONET), role in FAST, 496
 - PPP datagram with HDLC framing, diagram of, 181
 - PPP (Point-to-Point) protocol
 - authentication feature of, 182
 - development of, 180
 - PPVPN (Provider-to-Provider Virtual Private Network)
 - diagram of, 546–547
 - explanation and diagram of, 356–357
 - and MIBs, 789–790
 - preassigned reserved header values, meaning of, 304–305
 - presentation layer (L6) of OSIRM, dynamics of, 84
 - PRI (Primary Rate Interface), role in N-ISDN, 113–114
 - primary subframes, role in VoMPLS trunking, 458
 - priority control, performance
 - implications of, 632–633
 - priority discard thresholds, role in QoS, 631–632
 - priority queuing operation, diagram of, 630
 - priority queuing performance, capability of, 728–730
 - private-line networks
 - advantages and disadvantages of, 101–102
 - diagram of, 101
 - leased-line characteristics of, 100

permanent versus switched circuits in, 103–104

private lines and data communications, explanation of, 47–50

probability density function, role in QoS, 574–575

probability theory in communications engineering, randomness in communications networks, 677

profiles, role in SSCS and VoATM trunk signaling, 456

protection switching and fast rerouting in MPLS, 820

in private-line networks, 102

protocol efficiency comparison scorecard, 836

versus packet size, 830

protocol encapsulation, explanation of, 506–508

protocol field in IPv4, purpose of, 182–183

protocol layering concepts diagram of, 79

explanation of, 75–77

importance of, 77

protocol planes in ISDN, dynamics of, 90

protocol tunneling, explanation of, 346–347

protocols

- BSC (Binary Synchronous Communications), 51
- choosing, 782

CMIP (Common Management Interface Protocol), 780–781

computing relative efficiency of, 833

explanation of, 70

for multipoint and broadcast topologies, 42

proprietary types of, 781–782

for ring topology, 45

SNMP (Simple Network Management Protocol), 776–779

protocols carrying IP packets, overhead diagram, 832

provisioning, OAM&P philosophy of, 761–762

proxy signaling, role in ATM control plane protocols, 365

PS (program stream), role in MPEG-2, 468

PT (Payload Type) field, meaning of, 306–307

PTI (Payload Type Indicator) bit, role in ATM over MPLS network interworking, 349

PTSEs (PNNI Topology State Elements), purpose of, 422–423, 426

PTSPs (PNNI Topology State Packets), purpose of, 422, 424

pulse shaping, role in communications engineering, 681

pulse signals, dynamics of, 680–681

PU (physical units), role in SNA, 87

PVC status signaling and Frame Relay, explanation of, 152–155

PVCs (Permanent Virtual Connections)

- establishing with CONS, 91–92
- explanation of, 359
- links as, 40
- versus SVCs, 173

PWE2 and service emulation, role in link layer protocols, 499

PWE3 (Pseudo Wire Emulation Edge-to-Edge)

- diagram of support for voice, video, and WAN data, 356
- explanation of, 350–351
- and MIBs, 789–790

▼ Q

Q3 interface in ITU TMN, purpose of, 767

QAM (Quadrature Amplitude Modulation), role in communications engineering, 681–685

QFC (Quantum Flow Control), role in closed-loop flow control, 656

QoS and service categories of ATM layer, explanation of, 306–308

QoS classes for ITU-T ATM, dynamics of, 589–591

QoS parameters, list of, 307

QoS (Quality of Service)

- ATM cell transfer outcomes, 575–576
- ATM parameter definitions, 576–577
- ATM parameters, 573–577
- complexity analysis considerations, 844–845
- delivering, 630–634
- diagram of trade-offs with scheduling and discard strategies, 633
- explanation and requirements, 571–573
- in IP networks, 186–190
- performance implications of priority control, 632–633
- prioritized queuing and scheduling, 630–631
- priority discard thresholds, 631–632
- role in ATM VPs and label stacked MPLS LSPs, 639
- video considerations, 471
- weighted scheduling algorithms, 633–634

queuing and scheduling, role in QoS, 630–631

queuing power for congestion control schemes, graph of, 649

queuing theory

- Poisson arrivals and Markov processes, 701–705
- source model parameters, 699–702
- system models, 705–707

▼ R

random arrival processes, role in queuing theory, 702–705

random trials and Bernoulli processes, role in communications engineering, 678

randomness in communications networks, role in communications engineering, 677

rate-based versus credit-based scheme in closed-flow control, 659–666

RCCs (Routing Control Channels), role in PNNI, 422

RD (Routing Domain) in ATM control plane, explanation and diagram of, 381–382

RDF (Rate Decrease Factor), default value for, 663

RDI and AIS theory and operation, role in ATM OAM fault management, 800–803

RED (Random Early Detection) algorithm

- role in congestion control, 650
- using with TCP, 195

relaying and encoding, recurring trends in, 14–15

RELEASE and RELEASE COMPLETE messages in ATM signaling, 375

in Frame Relay SVC operation, 168–169

in N-ISDN D-channel switching, 118

RELEASE messages, role in point-to-multipoint connections, 377

reliability, explanation of and concerns about, 846–847

repeaters, estimating costs in HDLSLs, 50

Request bit in DQDB ACF, purpose of, 203

resource allocation, role in congestion management, 652–653

resource management

- admission control, 634–638
- MPLS admission control, 638

RESPONSE messages in SNMP, purpose of, 778

Resv messages, role in RSVP-TE, 401

retransmission protocols, analysis of, 735–736

RF-SSCS PDU formats, diagram of, 479

RFCs (Requests For Comments), purpose of, 28–29. *See* IETF RFC entries

RHC (remaining hop count), role in AINI and B-ISUP, 439

RIB (routing information base), role in MPLS forwarding operations, 309

RIF (Rate Increase Factor), default value for, 663

ring-switching schemes

- BLSR (bidirectional line-switched ring), 109
- BPSR (bidirectional path-switched ring), 109
- ULSR (unidirectional line-switched ring), 109
- UPSR (unidirectional path-switched ring), 109

ring topology, explanation of, 45–46

RM (resource management) cells, role in ABR parameters, 663–664

root-initiated point-to-multipoint connection establishment, diagram of, 376

round-trip delay, role in IPPM QoS, 578

route targets, role in aggregated routing network-based VPNs using tunnels, 553

routed protocols, features of, 507–509

router packet forwarding function protocol context, diagram of, 236

routers

- definition of, 208
- disadvantages of, 247–248

routing

- basic terminology for, 208–210
- versus bridging, 248
- concepts, systems, and protocols for, 235–247
- determining necessity of, 242
- and LISs, 242–245
- routing algorithms, purpose of, 211
- routing and bridging system design, guidelines for, 247–249
- routing functions, role in network layer of OSIRM, 83
- routing interface, functions, and architecture of Internet: diagram of, 239
- routing protocols. *See also* link-state routing protocols
 - functional context, 236
 - usage of, 237–238
- routing, role in TCP/IP, 178–179

RPC (Remote Procedure Call), role in TCP/IP, 178

RS-232 standard, using with DTE-to-DCE connection, 49

RST bit in TCP, purpose of, 191

RSVP functions in hosts and routers, diagram of, 188

RSVP (Resource Reservation Protocol)

- and Intserv (Integrated Services) in QoS in IP networks, 187–189
- and IP traffic descriptors, 582
- role in MPLS control plane, 361

- role in token bucket algorithm, 623–624
- using with RTP, 197

RSVP-TE (RSVP Traffic Engineering)

- downstream on demand ordered control explicit routing, 402–403
- priority, preemption, and resource affinity, 404
- refresh overhead reduction, 403–404
- reservation setup messages, 401–402
- role in LDP, 400–404
- tear down, error, and Hello messages, 402
- troubleshooting scaling problems with, 403–404

rt-VBR (real-time variable bit rate)

- class of service
- explanation of, 586
- usage with VoATM and N-ISDN, 456–457

RTD (round-trip delay), role in TCP/IP performance in congested scenario, 745

RTP packet header format, diagram of, 197

RTP (Real-Time Transport Protocol), explanation of, 196–197

run-length coding, role in data compression, 695

▼ S

s notation in queuing system models, meaning of, 705–706

S/T (user access) reference points, role in N-ISDN, 114

SAAL (Signaling AAL) model, explanation and diagram of, 365–366

SAPs (Service Access Points)

- role in IEEE 802.X series, 88
- role in protocol layers, 77

SAR (Segmentation and Reassembly) sublayer

- in AAL3/4, 339–340
- explanation of, 282
- in unstructured mode CS of AAL1, 329–330

scalability analysis

- addressing and hierarchy, 837–838
- capacity bottlenecks, 842
- connection-oriented versus connectionless paradigms, 840
- interface and speed support, 841
- packet forwarding and Moore's Law, 839–840
- user and routing table growth support, 838–839

scalability, explanation of, 210

scaling problems in RSVP, troubleshooting, 403–404

scheduling, modifying for GFR, 602

SCR (Sustainable Cell Rate), role in ATM traffic descriptors, 579, 581

SDH and ATM management plane reference architecture, diagram of, 797

SDH STM-M speed hierarchy, table of, 108

SDH (Synchronous Digital Hierarchy)

- basic structure of, 111
- basic transmission rate for, 111
- mapping plesiosynchronous signals into, 112
- and SONET, 106–113

SDLC (Synchronous Data Link Control)

- origins of, 125
- role in packet switching, 74

SDM (space division multiplexing), explanation of, 54, 61

SDSL (Single Line Digital Subscriber Line), explanation of, 293

SDT CS (Structured Data Transfer Convergence Sublayer)

- explanation of, 327–328
- role in AA1 clock recovery methods, 330

SE (shared explicit) style, role in RSVP-TE, 402

SEAL (Simple Efficient Adaptation Layer), development of, 340

SECBR (severely errored cell block ratio), role in ATM QoS, 576–577

security, interpretations of, 847–848

security management of networks, explanation of, 765

segment flow, role in ATM OAM

- flow reference architecture, 796

SEL (Selector) byte in ATM control plane, explanation and diagram of, 381–382

selective discard, role in congestion recovery, 667–668

selective-reject retransmission strategy, role in loss, 736

self-similar Internet traffic, role in MMPP, 705

sender templates, role in RSVP-TE, 401

Sequence Number field in TCP, purpose of, 190

service categories, explanation of, 307–308

service classes

- ABR (Available Bit Rate), 587
- CBR (Constant Bit Rate), 586
- Diffserv PHB (per-hop behavior), 594–595
- GFR (Guaranteed Frame Rate), 588
- ITU-T ATM QoS classes, 588–591
- mapping between ATM Forum and ITU-T QoS definitions, 591–594

- MPLS support for Diffserv, 595–596
- nrt-VBR (non-real-time variable bit rate), 587
- rt-VBR (real-time variable bit rate), 586
- UBR (Unspecified Bit Rate), 587
- service emulation and PWE3, role in link layer protocols, 499
- service interworking
 - ATM support for, 476
 - attributes for, 476
 - diagram of, 475
- Service Registry MIB in ILMI, contents of, 785
- session attributes, role in RSVP-TE, 404
- session ID in RSVP-TE, explanation of, 401–402
- session layer entities, connecting with transport layer of OSIRM, 83
- session layer (L5) of OSIRM, dynamics of, 84
- sessions
 - LDP protocol, 399
 - role in SNA, 86–87
- SET messages in SNMP, purpose of, 778
- SETUP messages
 - in ATM signaling, 374
 - in Frame Relay SVC operation, 168
 - in N-ISDN D-channel switching, 118–119
 - in point-to-multipoint connections, 376
 - in UNI 4.1 signaling, 372
- severely errored cell block ratio
 - ATM performance parameter, definition of, 816
- Shannon's channel capacity, dynamics of, 686–687
- shaping
 - diagram of, 627
 - ensuring conformance with, 624–629
 - methods for conformance, 625–626
- shared versus dedicated buffer performance, chart of, 717
- shim headers, explanation of, 312–315
- shortest path and constraint-based routing in MPLS, diagram of, 390–391
- shortest-path routing algorithm
 - example of, 410
 - role in IGP traffic engineering, 407–408
- signal modulation, dynamics of, 681–682
- signaling channel configurations for ATM control plane protocols, explanation of, 363–365
- signaling protocols, role in ATM and MPLS control plane, 359–360
- signaling terminology versus telephone calls, 93
- simplex communications, explanation of, 47–48
- simplex mode of physical layer of OSIRM, explanation of, 81
- SIR (Sustained Information Rate), role in SMDS, 205
- sites in VPNs, explanation of, 545–546
- SLA (service level agreements) in Frame Relay, reference model diagram, 160
- sliding and jumping window policing, role in leaky bucket policing, 616–619
- sliding windows, role in X.25 packet switching, 136–137
- SLIP (Serial Line IP), explanation of, 180
- Slow Start TCP algorithm, example of, 192
- SMDS access over ATM, diagram of logical configuration for, 488
- SMDS L3_PDU header, contents of, 200–201
- SMDS over DQDB operation, example of, 204
- SMDS (Switched Multimegabit Data Service)
 - 802.6 PDU formats, 199–202
 - 802.6 protocol structure, 199
 - addressing plan for, 200
 - ATM access to, 488–489
 - and DQDB operation, 202–204
 - origins of, 74, 198–199
 - service aspects of, 205–206
- SMT (Station Management) function in FDDI protocol stack, purpose of, 226
- SN (Sequence Number) fields in AAL1 SAR, explanation of, 324–325
- SNA (Systems Network Architecture)
 - dynamics of, 86–87
 - role in packet switching, 74
 - role of sessions in, 86–87
 - and SDLC (Synchronous Data Link Control) in multipoint topology, 43
- SNMP (Simple Network Management Protocol)
 - versus CMIP, 780
 - message types, 778
 - object model of network management, 776–777
 - in TCP/IP, 178
 - versions 2 and 3, 778–779
- SNP (Sequence Number Protection) field in AAL1 SAR, explanation of, 324–325
- software and hardware in emulated LANs, dynamics of, 511–513
- SOH character, role in BSC, 51–52
- SONET frame format, explanation of, 110–112
- SONET overhead, amount of, 826
- SONET/SDH digital hierarchy
 - payload and overhead rates, table of, 109
- SONET/SDH multiplexing structure, diagram of, 113
- SONET/SDH systems
 - advisory about deployment of, 110
 - explanation of, 106–113
 - versus PDH, 107
 - role in private-line networks, 101–102
- SONET/SDH systems, diagram of architecture layers in, 107
- SONET STS-3c direct TC mapping of ATM cells, explanation of, 288
- SONET STS-N/OC-N speed hierarchy, table of, 108
- SONET (Synchronous Optical Network), speeds of, 52
- SONET VT1.5 format, diagram of, 112
- Source ID field, role in ATM OAM, 803
- source model parameters in queuing theory, role in traffic engineering, 699–702
- source route bridging, explanation of, 232
- sources, role in ABR, 660
- space division switching
 - example of, 62–63
 - explanation of, 56
- spanning tree and LANE, explanation of, 518
- specification process, diagram of, 32–34
- SPF (Shortest Path First) algorithm, usage of, 241
- splitters, using with xDSL schemes, 294
- SPVC (switched PVC), role in Frame Relay, 148
- SPVCs (switched PVCs)
 - explanation of, 359
 - role in PNNI, 431–432
- SREJ (Selective Reject) control protocol, usage of, 135
- SRP (Source Routing Protocol), explanation of, 233–234
- SRTS (Synchronous Residual Time Stamp), role in AAL1 clock recovery methods, 330–331
- SS7 (System No. 7) signals
 - explanation of, 118
 - role in B-ISDN, 255–256
- SSCF (Service Specific Coordination Function), role in ATM control plane protocols, 365
- SSCOP (Service Specific Connection-Oriented Protocol), role in ATM control plane, 366–368
- SSCPs (system services control points), role in SNA, 87
- SSCS (Server-Specific Convergence Sublayer)
 - role in AAL, 321

role in AAL3/4, 337
 role in VoATM trunk signaling, 452–453
 typical function of, 456
 using with AAL2 and broadband local loop emulation, 461

SSM (Single Segment Message) in AAL3/4, explanation of, 337

ST (Segment Type) field in AAL3/4, explanation of, 337

stability, explanation of and concerns about, 846–847

standardization, predicting future of, 36

standards
 approval and consensus of, 34
 business and politics factors, 35
 charters and work plans involved in, 33
 creating, 30–34
 creation process, 32–34
 drafting and review of, 33–34
 measures of success and proven approached, 35–36
 meetings and contributions involved in, 33
 user acceptance and interoperability factors, 34
 worldwide cooperation for, 22–23

standards bodies
 ANSI (American National Standards Institute), 30
 ATM Forum, 27–28
 for ATM and MPLS, explanation of, 26
 for B-ISDN and ATM, 30
 DSL Forum, 30
 ETSI (European Telecommunications Standards Institute), 30
 FRF (Frame Relay Forum), 29
 IETF (Internet Engineering Task Force), 28–29
 ITU-T (International Telecommunications Union-Telecommunications), 27
 MPLS Forum, 29–30

star topology, explanation of, 44–45

state transition rate diagram, role in MMPP, 704

statistical multiplex gain model
 role in equivalent capacity, 722–725
 sources required for achievement of, 725

statistical multiplexers, definition of, 58

STATUS ENQUIRY message in PVS status signaling, explanation of, 154, 156

STATUS message in PVS status signaling, explanation of, 154–157

STDM (statistical division multiplexing), explanation of, 60–61

STF (Start Field), role in AAL2, 335

STM-1 frames, bytes in, 111

STM-1 (synchronous transfer module), explanation of, 107–108

STM (Synchronous Transfer Mode), explanation of, 50–53

Stop gremlin, role in leaky bucket buffering, 626

store-and-forward approach, using with LAP-B protocol, 133–134

STP (Spanning Tree Protocol), explanation of, 232–233

strictly explicitly routed paths in MPLS, explanation of, 392

structured mode CES
 internetworking, diagram of, 466

STS-1 SONET SPEs, mapping VT1.5s into, 111

STS-3c direct TC mapping of ATM cells, explanation of, 288

STS-3d TC sublayer mapping, diagram of, 289

STS-N SPE frame format, explanation of, 110

STs (synchronous transfer signals), role in SONET, 107

STX (Start of Text) character, role in BSC, 51–52

subnet mask decimal and binary values, table of, 242

subnetting
 in large networks, 242
 in TCP/IP, 180–182

summarized peer groups, role in PNNI, 424–426

supervisory frame, role in HDLC, 128

supportability and operability, explanation of and concerns about, 847

SVCs (Switched Virtual Connections)
 establishing with CONS, 91–92
 explanation of, 359
 in Frame Relay, 168
 versus PVCs, 173
 role in FR/ATM service interworking, 484–486

switch buffering
 performance diagram, 714
 types of, 713

switched 56, explanation of, 98

switched services, interface for, 99

switches, functionality of, 63

switching
 definition of, 53–54
 and digitized voice transmission, 97–98
 examples of, 62–67
 in point-to-point topology, 56

symmetric soft rerouting, role in AINI, 438

SYN bit in TCP, purpose of, 191

SYN characters, role in BSC, 51–52

synchronization in TCP, occurrence of, 195

synchronous data transmission, explanation of, 50–52

syndromes, role in CRC codes, 691

▼ T

T1 and T3 signals in PDH, meanings of, 105

T32 variable, role in QoS, 574

tag edge routers, usage of, 268–269

tag switching
 diagram of, 269
 role in MPLS, 267–270

tagging
 in ATM conformance, 584
 modifying for GFR, 601–602

talker echo phenomenon, presence in packet voice networking, 448

TAS (transported address stack) specification, role in ATM control plane addressing, 384

TAT (theoretical arrival time), role in GCRA and virtual scheduling, 620

TBE (Transient Buffer Exposure), default value for, 663

TC cell rate decoupling, explanation of, 290

TC (Transmission Convergence) sublayer of ATM physical layer, explanation of, 282, 287–288

TCI (Tag Control Information) fields, role in Ethernet user priority and VLANs, 219–220

TCP/IP networking context, explanation of, 178–180

TCP/IP operation, example of, 192

TCP/IP (Transmission Control Protocol/Internet Protocol) origins of, 176–177

performance in congested scenarios, 743–746

service aspects of, 198

structure of, 177–178

TCP performance considerations in congested scenarios, 743–746

multiplexing voice
 conversations statistically, 746–747
 over ATM, 742–743
 UBR and ABR, 742–743
 voice and data integration, 745
 voice/data integration savings, 747–748
 voice traffic model, 745–746
 window size impact on throughput, 742, 744–745

TCP segment format, diagram of, 191

TCP slow start congestion window size behavior, diagram of, 194–195

TCP (Transmission Control Protocol)
 congestion avoidance in, 194

- congestion window values
 - for, 195
- dynamic windowing flow
 - control diagram, 193
- dynamics of, 190–195
- enhancements to, 195
- fast retransmit and fast recovery
 - in, 195
- synchronization in, 195
- traffic and congestion control
 - aspects of, 192–195
- TCR (Tagged Cell Rate), default value for, 663
- TDM networks, significance of
 - timing transfer in, 464
- TDM (time division multiplexing)
 - versus address multiplexing, 60
 - explanation of, 55, 56
 - origin of, 60
 - and PDH (Plesiochronous Digital Hierarchy), 104–106
 - samples per second per DS0 channel, 98
 - usage of, 58, 63
- TDS (time division switch),
 - explanation of, 63
- TE (terminal equipment)
 - in ISDN, 89
 - in N-ISDN, 114
 - providing local connections to, 48–49
- TE (traffic engineering). *See* traffic engineering
- tear messages in RSVP-TE,
 - explanation of, 402
- technology trends
 - accelerating bandwidth
 - principle, 21–22
 - distributed computer
 - communication protocols, 20
 - increased LAN and WAN
 - speeds, 21
 - modernization of transmission
 - infrastructures, 20–21
 - processor and memory costs:
 - Moore's Law, 19–20
 - worldwide cooperation for
 - standards, 22–23
- telegraph pulse and binary on/off
 - keying, role in digital signals and spectra, 680–681
- telephone calls versus signaling
 - terminology, 93
- telephone networks, circuit
 - switching in, 96
- TFIB (tag forwarding information base), role in tag switching, 269–270
- TFTP (Trivial FTP), role in
 - TCP/IP, 178
- threshold control of selective
 - discard, diagram of, 632
- throughput, role in congestion, 646–649
- throughput versus loss probability,
 - chart of, 737
- time division switching
 - example of, 63–64
 - explanation of, 56
- time stamp method delay
 - estimation, diagram of, 818
- TINA-C (Telecom Information Networking Architecture Consortium), role in network management, 769–771
- TM 4.0 specification, impact on
 - service categories, 588
- TM 4.1 service classes, PNNI 1.1
 - support for, 427
- TMN (Telecommunications Management Network),
 - dynamics of, 766–769
- token bucket algorithm, role in IP
 - and MPLS policing, 623–624
- token bucket example of IP
 - and MPLS policing
 - conforming packet flow,
 - diagram of, 621
 - nonconforming packet flow, 622
- token bucket operation,
 - diagram of, 623
- token bucket shaping, dynamics of, 627–629
- Token Buckets versus Leaky Buckets, 596
- Token Ring 802.5 versus FDDI
 - operation, 227
- Token Ring architecture
 - versus FDDI, 225
 - origin of, 89
- Token Ring configuration, diagram
 - of, 221
- Token Ring protocol, explanation
 - of, 220–222
- tokens, functionality of, 221
- tolerances, allocation of, 581
- topologies, types of, 40–47
- TOS (Type of Service) field in IPv4,
 - purpose of, 182
- Total Length field in IPv4, purpose
 - of, 182
- TPID (Tag Protocol ID), role in
 - Ethernet user priority and VLANs, 219–220
- trace features in PNNI and AINI,
 - explanation of, 436–437
- Traceroute IP-based management
 - tool, using with MPLS, 790–791
- traffic and congestion control
 - overview of, 611
 - time scales of, 611
- traffic and performance data,
 - measuring, 750–751
- traffic contracts
 - ATM equivalent terminal
 - model, 568–569
 - Diffserv per-hop and
 - per-domain behavior models, 569–570
 - explanation of, 566–567
 - generic allocation of
 - impairments model, 567
 - reference models for, 567–570
 - and resource management, 634–639
- traffic descriptors and tolerances,
 - explanations of, 581
- traffic engineering
 - accuracy in, 750
 - equivalent capacity, 720–728
 - in IGP, 407–408
 - in IP backbones, 409–411
 - and MIBs, 789
 - modeling accuracy, 699
 - source model parameters for
 - queuing theory, 699–702
 - source model traffic parameter
 - characteristics, 698–699
- traffic matrix, collecting information
 - about, 751
- traffic modeling and call attempts,
 - dynamics of, 708–709
- traffic parameters and conformance
 - definitions, 307–308
 - ATM conformance, 583–585
 - ATM traffic descriptors, 579–581
 - IP traffic conformance, 585
- traffic shaping
 - diagram of, 627
 - ensuring conformance with, 624–629
 - methods for conformance, 625–626
- transfer mode, definition of, 52
- translation mode, operation of
 - bridges in, 231
- transmission paths, links as, 40
- transparent mode, operation of
 - bridges in, 231
- transport layer (L4) of OSIRM,
 - dynamics of, 83
- TRAP messages in SNMP, purpose
 - of, 778
- Trm, default value for, 663
- trunk side, role in multiplexing, 57–58
- trunking
 - ATM support for, 476
 - attributes for, 476
 - diagram of, 475
- Tspec (Traffic Specification) for
 - RSVP, explanation of, 188–189
- TSTP (Time Stamp) field in PM,
 - meaning of, 812
- TTL (Time to Live) field in IPv4,
 - purpose of, 182
- TUC (Total User Cell) field in PM,
 - meaning of, 812
- tunnel types, choosing, 557–558
- tunneling, explanation of, 346–347
- tunnels
 - role in VPNs, 547
 - using aggregated routing
 - network-based VPNs with, 550–554
 - using virtual router
 - network-based VPNs with, 554–556
- TUs (tributary units), role in SONET
 - frame format, 112
- two-fiber rings, usage of, 109–110
- type 3 SSCS packets, role in VoATM
 - trunk signaling, 453–455

▼ U

U-plane (user plane) in Frame Relay context for, 149–150
 role in Frame Relay, 138–140

U utilization, role in statistical multiplex gain model, 724

UBR and ABR, TCP performance considerations, 742–743

UBR (Unspecified Bit Rate) class of service
 in ATM as end-to-end service, 261
 with BSC and MDCR, 603–604
 explanation of, 587
 with optional MDCR parameter, 605–607

UDP format, diagram of, 196

UDP (User Datagram Protocol)
 in SNMP, 778
 in TCP/IP, 177–178, 196

ULSR (unidirectional line-switched ring), explanation of, 109

unassigned cells, explanation of, 290

unbalanced HDLC control links, explanation of, 126

unbalanced interchanges, role in DTE-to-DCE connections, 49

uncontrolled mode, role in closed-loop flow control, 657

undetected error performance of HDLC and AAL5, dynamics of, 694–695

UNI 4.0 and ITU-T standards, explanation of, 368–370

UNI 4.1
 signaling message information elements, 372–373
 table of message types, 371

UNI and NNI
 diagram of, 259
 reference configurations, 297
 signaling-standard mapping, 434

UNI signaling 4.1 and ITU-T standards, dynamics of, 370–371

UNI (User-to-Network Interface)
 signaling protocol
 definition in ATM, 258
 explanation of, 296–297
 role in ATM and MPLS control plane, 359–360

UNI (User-to-Network Interface)
 standard for Frame Relay, diagram of, 152

unidirectional rings
 explanation of, 108–109
 role in VCs, 299

unnumbered frame, role in HDLC, 128–129

unsolicited downstream versus downstream on demand label distribution, role in MPLS, 392–393

unstructured mode CES
 internetworking function, diagram of, 465

UPC differences in leaky bucket policing, diagram of, 618

UPC (Usage Parameter Control)
 purpose of, 304
 role in ATM CAC, 637
 role in ATM policing, 613
 role in congestion recovery, 670

UPC via windowing example in leaky bucket policing, diagram of, 617

UPDATE messages, using with BGP, 412

UPSR (unidirectional path-switched ring), explanation of, 109

URG bit in TCP, purpose of, 191

users in VPNs, explanation of, 545

users, role in creating standards, 31

UUI (User-to-User Indication) field, role in AAL2, 334

▼ V

VAIs (virtual application instances), role in MPLS in IP networks, 411–412

VBR (Variable Bit Rate), role in ATM as end-to-end service, 261

VC-based multiplexing, role in multiprotocol encapsulation over AAL5, 508–510

VC links, explanation of, 299

VC merging, diagram of, 271

VC multiplexing
 versus LLC encapsulation, 510–511
 role in multiprotocol encapsulation over AAL5, 506

VCC MIB in ILMI, contents of, 785

VCC (Virtual Channel Connection), explanation of, 299

VCI values reserved by ITU-T and ATM Forum, table of, 306

VCIP (VCI Present) indicator, role in ATM over MPLS network interworking, 349

VCIs (Virtual Channel Identifiers), explanation and usage of, 299–300

VCs (virtual containers)
 in ATM, 261–262
 diagram of, 298
 explanation of, 297–302
 in SONET, 108

VDSL (Very high rate Digital Subscriber Line), explanation of, 293

vendors, role in creating standards, 30–31

Version field of IPv6, purpose of, 185–186

VF (Variance Factor) additive link attributes, role in PNNI, 428

VFLs (virtual forwarding instances)
 role in aggregated routing
 network-based VPNs using tunnels, 552

role in network-based IP VPNs, 548

role in virtual router network-based VPNs using tunnels, 554–556

VFS (virtual forwarding and switching) function, role in Ethernet over MPLS, 522–523

video and audio protocols, sensitivity to delay variation, 738

video coding standards
 bit rates and compression ratios for, 468
 NTSC (National Television Standards Committee), 467

video packets efficiency analysis, dynamics of, 834–835

video, QoS considerations, 471

video signals, protocol efficiency for transport of, 835

virtual circuits in X.25 standard, explanation of, 132

virtual routers, role in MPOA, 538, 540

virtual UNI capability, role in ATM control plane protocols, 364–365

VLANs (Virtual LANs), Ethernet user priority and, 219–220

VoATM (voice over ATM) trunking
 ATM ALL2 narrowband SSCS, 452–456
 and N-ISDN relationships to AAL and QoS, 456–457
 and packet networks, 467–471
 trunk signaling, 450–452

VOD (Voice on Demand)
 specification, explanation of, 469–470

voice
 over IP access, 828
 over IP backbone, 828
 over MPLS, 828

voice and data integration, role in TCP performance, 745

voice coding, diagram of techniques, standards, and peak bit rates for, 447

voice conversations, peak and average rates in, 700

voice/data integration savings, role in TCP performance, 747–748–9

voice over packet efficiency analysis, table of, 828

voice packetization delay, role in choosing ATM cell sizes, 278–279

voice traffic model, role in TCP performance, 745–746

voice trunking using ATM and MPLS, explanation of, 449–450

VoMPLS control subframe, sending in packets over MPLS LSPs, 459

VoMPLS primary subframes
 versus AAL2 narrowband SSCS type 1 packets, 459
 number of octets in, 459

VoMPLS (voice over MPLS)
 trunking, explanation of, 457–459

VoPacket (voice over packet), contents of, 449

VP and VC switching and cross-connection, explanation of, 300

VPC MIB in ILMI, contents of, 785

VPCs (Virtual Path Connection Identifiers), role in ATM control plane protocols, 363–364

VPCs (Virtual Path Connections), explanation of, 299–300

VPI/VCI switching for VPC and VCC, diagram of, 302

VPIs (Virtual Path Identifiers), explanation and usage of, 299–300

VPLS (Virtual LAN Service) in Ethernet over MPLS, 521–525
Internet access, 525–526

VPN IPv4 address prefix, explanation of, 553

VPN representations and configuration complexity, explanation of, 558–560

VPNs (virtual private networks) definition of, 546
role in intranets and extranets in, 546
role of enterprises in, 546
role of sites in, 545
role of users in, 545
using for Frame Relay networking, 148–149
using MPLS tunneling with, 276

VPs (virtual paths) diagram of, 298
example of, 301–302

explanation of, 297–302

VR-based VPNs, considerations and trade-offs with, 556–557

VRs (virtual routers), using with network based VPNs and tunnels, 554–556

VS/VD (Virtual Source/Virtual Destination) mode, role in closed-loop flow control, 659

VSLs (virtual switch instances), role in Ethernet over MPLS, 522

VT1.5 format, diagram of, 112

VT1.5s, mapping into STS-1 SONET SPEs, 111

VTs (virtual tributaries) in SONET, explanation and usage of, 107–108

▼ W

w notation in queuing system models, meaning of, 706

WANs (wide area networks), role of point-to-point topology in, 41

wavelength switching, explanation of, 56

WDM (wavelength division multiplexing)
example of, 66–67
explanation of, 56
versus FDM, 59
role in private-line networks, 102

web sites
Gigabit and 10 Gbps
Ethernet, 224
RED and ECN, 195

WFQ (Weighted Fair Queuing), role in QoS, 633–634

WGN (Additive White Gaussian Noise) communications channel error model, explanation of, 685

▼ X

X.25 packet layer flow control, diagram of, 136

X.25 packet switching
comparing to ISDN and Frame Relay information-frame formats, 131
example of, 133–135
versus Frame Relay, 137–138
link layer protocol, 125–131
networking context for, 124–125
notation for order of bit transmission in, 141
origins of, 122–123
versus OSIRM, 123–124
protocol structure of, 123–124
service aspects of, 137
standardization of, 122
traffic and congestion control aspects of, 135–137
types of services defined in, 132

X.25 virtual call establishment, diagram of, 133

xDSL
physical layer for ATM, 292–295
reference model, 295

INTERNATIONAL CONTACT INFORMATION

AUSTRALIA

McGraw-Hill Book Company Australia Pty. Ltd.
TEL +61-2-9415-9899
FAX +61-2-9415-5687
<http://www.mcgraw-hill.com.au>
books-it_sydney@mcgraw-hill.com

CANADA

McGraw-Hill Ryerson Ltd.
TEL +905-430-5000
FAX +905-430-5020
<http://www.mcgrawhill.ca>

GREECE, MIDDLE EAST, NORTHERN AFRICA

McGraw-Hill Hellas
TEL +30-1-656-0990-3-4
FAX +30-1-654-5525

MEXICO (Also serving Latin America)

McGraw-Hill Interamericana Editores S.A. de C.V.
TEL +525-117-1583
FAX +525-117-1589
<http://www.mcgraw-hill.com.mx>
fernando_castellanos@mcgraw-hill.com

SINGAPORE (Serving Asia)

McGraw-Hill Book Company
TEL +65-863-1580
FAX +65-862-3354
<http://www.mcgraw-hill.com.sg>
mghasia@mcgraw-hill.com

SOUTH AFRICA

McGraw-Hill South Africa
TEL +27-11-622-7512
FAX +27-11-622-9045
robbyn_swanepoel@mcgraw-hill.com

UNITED KINGDOM & EUROPE (Excluding Southern Europe)

McGraw-Hill Education Europe
TEL +44-1-628-502500
FAX +44-1-628-770224
<http://www.mcgraw-hill.co.uk>
computing_neurope@mcgraw-hill.com

ALL OTHER INQUIRIES Contact:

Osborne/McGraw-Hill
TEL +1-510-549-6600
FAX +1-510-883-7600
<http://www.osborne.com>
omg_international@mcgraw-hill.com