CONNECTION-ORIENTED NETWORKS

SONET/SDH, ATM, MPLS and Optical Networks

Harry G. Perros

WILEY

Connection-oriented Networks

SONET/SDH, ATM, MPLS and OPTICAL NETWORKS

Harry G. Perros

John Wiley & Sons, Ltd

Connection-oriented Networks

Connection-oriented Networks

SONET/SDH, ATM, MPLS and OPTICAL NETWORKS

Harry G. Perros

John Wiley & Sons, Ltd

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears
in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data:

Perros, Harry G.
Connection-oriented networks : SONET/SDH, ATM, MPLS, and optical networks
/ Harry G. Perros.
p. cm.
Includes bibliographical references and index.
ISBN 0-470-02163-2 (cloth)
1. Telecommunication systems. 2. Computer networks. I. Title.
TK5101.P3853 2005
621.382′1 – dc22
2004026216

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-02163-2

Typeset in 10/12pt Times by Laserwords Private Limited, Chennai, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire
This book is printed on acid-free paper responsibly manufactured from sustainable forestry
in which at least two trees are planted for each one used for paper production.

To

Helen, Nick, and Mikey!

About the Author

Harry G. Perros is a Professor of Computer Science, an Alumni Distinguished Graduate Professor, and the Program Coordinator of the Master of Science degree in Computer Networks at NC State University.

He received the B.Sc. degree in Mathematics in 1970 from Athens University, Greece, the M.Sc. degree in Operational Research with Computing from Leeds University, England, in 1971, and the Ph.D. degree in Operations Research from Trinity College Dublin, Ireland, in 1975. He has held visiting faculty positions at INRIA, Rocquencourt, France (1979), NORTEL, Research Triangle Park, North Carolina (1988-89 and 1995-96) and University of Paris 6, France (1995-96, 2000, and 2002).

He has published numerous papers in the area of performance modeling of computer and communication systems, and he has organized several national and international conferences. He has also published two print books: *Queueing Networks with Blocking: Exact and Approximate Solutions*, Oxford Press 1994, *An Introduction to ATM Networks*, Wiley 2002, and an e-book *Computer Simulation Techniques – The Definitive Introduction,* 2002 (available through his Web site).

In 1995, he founded the IFIP Working Group 6.3 on the *Performance of Communication Systems*, and he was the chairman from 1995 to 2002. As of 2004, he is the chairman of the IFIP Working Group 6.10 on *Optical Networks*. He is also a member of IFIP Working Groups 6.2, and 7.3, and an IEEE Senior Member. In addition, he is an associate Editor for the *Performance Evaluation* Journal, and the *Telecommunications Systems* Journal.

His current research interests are in the area of optical networks.

In his free time he likes to go sailing on the *Aegean*, a Pearson 31!

Contents

Preface

This book explores a number of connection-oriented packet-switching networks and circuit-switching networks. These networks, though seemingly different, share common networking principles, and in some cases, one more recent network builds upon an older one.

The first connection-oriented network is probably the familiar and ubiquitous telephone network. This is a circuit-switching network, whereby a connection is established between the two parties by allocating a channel on each transmission link along the path. The concept of connection, as used in the telephone system, has been emulated for a long time in computer packet-switching networks. In view of this, such networks are known as connection-oriented packet-switching networks.

In this book, we explore two connection-oriented packet-switching networks: *ATM networks* and *multi-protocol label switched (MPLS) networks*. ATM is a legacy network that was developed in the late 1980s and early 1990s. It is used in the backbone to transport IP traffic, in access networks (such as ADSL-based networks and passive optical networks), and in cellular telephony. The MPLS architecture is an extension of ATM, and must be used to introduce *quality of service (QoS)* in IP networks.

Two circuit-switching networks – SONET/SDH and Optical Wavelength Routing networks – are also presented in this book. SONET/SDH has been around for a long time, whereas optical wavelength routing networks are relatively new. SONET/SDH is the underlying transport network of the telephone system and is used in all modern packet-switching networks, such as IP and ATM. Wavelength routing networks are also circuit-switching networks since the transmission of data is done using optical circuit-switching connections, known as *lightpaths*. We also present a new optical networking scheme, which has not yet been standardized, known as *optical burst switching (OBS),* which can be seen as lying between packet switching and circuit switching.

Finally, the book contains a chapter on access networks, such as ADSL-based networks, cable modems, and ATM passive optical networks, and a chapter on voice over ATM and voice over MPLS.

The book is primarily intended as a textbook in a second course on computer networks at the graduate level or senior undergraduate level. It can also serve as a reference for field networking engineers who would like to learn more about connection-oriented packet-switching networks and circuit-switching networks. The only prerequisite for this book is a basic knowledge of computer networking principles. The book does not deal explicitly with IP networks, and so it is not necessary to have a detailed knowledge of the IP network in order to understand the material presented here.

The book consists of twelve chapters, covering the following topics:

- Chapter 1 – Introduction
- Chapter 2 – SONET/SDH
- Chapters 3, 4 and 5 – ATM networks
- Chapters 6 and 7 – MPLS
- Chapters 8, 9 and 10 – Optical networks
- Chapter 11 – Access networks
- Chapter 12 – Voice over ATM and MPLS.

How current are the specifications?

Most of this book was written during 2003 and 2004, and therefore the specifications presented in the book pertain to that timeframe. Since networking technology is continuously evolving, consulting the standard committees' Web sites for updates is strongly encouraged.

A note to the students

This book grew out of teaching a course on connection-oriented networks and a course on optical networks for the degree of *Master of Science in Computer Networks* at NC State University. I like to tell my students jokingly that if they want to get an A they have to read the book five times. If they read it four times, then they will end up with a B, and if they read it three times they will end up with a C, and so on – which always spurs some lively discussion! However, there is some truth in this statement, since the book deals with descriptive material, which has been developed over several years by different standards bodies. As a result, the networking concepts are convoluted and not easy to understand in one or two readings. A good way to test your understanding of a particular networking scheme is to ask yourself a question, and then try to answer it. If you can answer it immediately without hesitation, then you know it. Otherwise, you need to go back for another reading!

A note to the instructor

At the end of each chapter, a Problems section provides self-review exercises. Also, at the end of some chapters there is a simulation project designed to reinforce some of the intricacies of the networks presented in this book. Specifically, the following three simulation projects have been included:

- *Chapter 3: AAL 2*
- *Chapter 4: ATM traffic characterization of an MPEG video source*
- *Chapter 9: Calculation of call blocking probabilities in a wavelength routing network*

Each simulation project contains enough information so that familiarity with discrete-event simulation techniques is not required. More information on basic discrete-event simulation techniques can be found in many simulation books, including my e-book *Computer Simulation Techniques – The Definitive Introduction,* available free of charge from my Web page http://www.csc.ncsu.edu/faculty/perros//index.html.

The solution to the problems and the code and results for the simulation projects can be found in a solution manual, available from Wiley's Web site: http://www.wiley.com/go/connection-oriented. A Powerpoint presentation for each chapter is also available from the Wiley Web site.

Acknowledgments

I would like to thank Xenia Mountrouidou for proofreading the manuscript; Laura Holley for copyediting the manuscript; and Venkat Jonnadula for helping me to prepare the solutions manual. And to Birgit Gruber, Commissioning Editor, and Sally Mortimore, Executive Commissioning Editor, at Wiley – thanks for your faith in me!

Harry Perros

List of Abbreviations

2F-BLSR	two-fiber bidirectional line switched ring
2F-OBLSR	two-fiber optical bidirectional link sharing ring
2F-UPSR	two-fiber unidirectional path switched ring
4F-BLSR	four-fiber bidirectional line switched ring
4F-OBLSR	four-fiber optical bidirectional link sharing ring
A2oMPLS	AAL 2 over MPLS
AAL	ATM adaptation layer
ABR	available bit rate
ABT	ATM block transfer
ACR	allowable cell rate
ADM	add/drop multiplexer
ADPCM	adaptive pulse code modulation
ADSL	asymmetric digital subscriber line
AFI	authority and format identifier
ANP	AAL 2 negotiation procedure
ANSI	American National Standards Institute
APON	ATM passive optical networks
APS	automatic protection switching
ARIMA	autoregressive integrated moving average
ARP	address resolution protocol
ATM	asynchronous transfer mode
ATU-C	ADSL transceiver unit at the central office
ATU-R	ADSL transceiver unit at the remote terminal
BAS	broadband access server
BCD	binary coded decimal
BE	best effort service
BECN	backward explicit congestion notification
B-frame	bidirectional-coded frame
BGP	border gateway protocol
BIP	bit interleaved parity
B-ISDN	broadband integrated services data network
BLSR	bidirectional line switched ring
BPSR	bidirectional path switched ring
BT	burst tolerance

CAC	call admission control
CAS	channel-associated signaling
CBR	constant bit rate
CBS	committed burst size
CCITT	International Telegraph and Telephone Consultative Committee
CCR	current cell rate
CCS	common channel signaling
CDR	committed data rate
CDVT	cell delay variation tolerance
CER	cell error rate
CES	circuit emulation service
cHEC	core head error control
CI	connection identifier
CID	channel identifier
CIDR	classless inter-domain routing
CLEC	competitive local exchange carrier
CLP	cell loss priority bit
CLR	cell loss rate
CM	cable modem
CMR	cell misinsertion rate
CMTS	cable modem termination system
CO	central office
CoS	class of service
CPCS	common part convergence sublayer
CPS	common part sublayer
CRC	cyclic redundant check
CR-LDP	constraint routing-label distribution protocol
CS	convergence sublayer
CSI	convergence sublayer indication
CTD	cell transfer delay
CVoDSL	channelized voice over DSL
DBCE	dynamic bandwidth circuit emulation services
DBR	deterministic bit rate
DCC	data country code
DCE	data communication equipment
DCS	digital cross connect system
DDS1	digital subscriber signaling system no. 1
DDS2	digital subscriber signaling system no. 2
diffserv	differentiated service
DMCR	desirable minimum cell rate
DMT	discrete multi-tone
DOCSIS	data-over-cable service interim specification
DoS	data over SONET/SDH
DS	delay sensitive service
DSL	digital subscriber loop
DSLAM	ADSL access multiplexer
DSP	domain-specific part

DSS1	digital subscriber signaling system no.1
DTE	data terminal equipment
DTMF	dual-tone multi-frequency
DWDM	dense DWDM
EADPCM	embedded adaptive pulse code modulation
EBS	excess burst size
ECON	enterprise system connect
EDF	early deadline first
EDFA	Erbium-doped fiber amplifier
EDU	encoding data units
EFCN	explicit forward congestion notification
EIA	Electronics Industries Association
E-NNI	external network node interface
ER	explicit rate
ER-TLV	explicit route TLV
ESI	end system identifier
EXI	extension header identifier
FAS	frame alignment signal
FCS	frame check sequence
FDL	fiber delay lines
FDM	frequency division multiplexing
FDMA	frequency division multiple access
FEC	forwarding equivalent class
FF	fixed filter style
FIB	forwarding information base
FICON	fiber connection
FRP	fast reservation protocol
FSAN	full service access networks
FSC	fiber-switch capable interface
FTN	FEC-to-NHLFE map
FTTB	fiber to the basement
FTTB/C	fiber to the basement/curb
FTTC	fiber to the curb
FTTCab	fiber to the cabinet
FTTH	fiber to the home
GbE	gigabit Ethernet
GCRA	generic cell rate algorithm
GFC	generic flow control
GFP	generic framing procedure
GFP-F	frame-mapped GFP
GFP-T	transparent-mapped GFP
GFR	guaranteed frame rate
GMPLS	generalized MPLS
GOP	group of pictures
G-PID	generalized payload identifier
HDLC	high-level data link control
HDSL	high data rate DSL

HEC	header error control
HFC	hybrid fiber coaxial
HO-DSP	high-order DSP
IBP	interrupted Bernoulli process
ICD	international code designator
ICMP	Internet control message protocol
IDI	initial domain identifier
IDP	initial domain part
IDSL	ISDN DSL
IE	information elements
IEC	International Electronical Commission
IEEE	Institute of Electrical and Electronics Engineering
IETF	Internet Engineering Task Force
IFP	Interrupted fluid process
I-frame	intra-coded frame
ILEC	incumbent local exchange carrier
InATMARP	inverse ATMARP
I-NNI	internal network-node interface
intserv	integrated services
IP	Internet protocol
IPCC	IP control channel
IPP	interrupted Poisson process
ISDN	integrated service data network
ISO	International Organization of Standards
ISOC	Internet Society
ISP	Internet service provider
ISUP	integrated service user part
ITU	International Telecommunication Union
IWF	interworking function
JET	just enough time
JIT	just in time
L2TP	layer 2 tunnel protocol
LAC	L2TP access concentrator
laser	light amplification by stimulated emission of radiation
LCAS	link capacity adjustment scheme
LD-CELP	low delay code excited linear prediction
LDP	label distribution protocol
LES	loop emulation service
LFIB	label forward information base
LI	length indicator
LIS	logical IP subnet
LMP	link management protocol
LOH	line overhead
LSA	link-state advertisements
LSC	lambda switch capable interface
LSP	label switched path
LSR	label switching router

LWPF	low water peak fiber
MBS	maximum burst size
MCR	minimum cell rate
MEMS	micro electronic mechanical systems
MFAS	multi-frame alignment signal
MFS	maximum frame size
MIN	multistage interconnection network
MMBP	Markov modulated Bernoulli process
MMPP	Markov modulated Poisson process
MPLS	multi-protocol label switching
MSC	message switching center
MTP	message transfer part
MTU	maximum transfer unit
NAS	network access server
NDF	negative dispersion fiber
NHLFE	next hop label forwarding entry
N-ISDN	narrowband ISDN
NNI	network node interface
nrtPS	non-real-time polling service
NRT-SBR	non-real-time statistical bit rate
NRT-VBR	non-real-time variable bit rate
NSAP	network service access point
NSP	network service provider
NTR	network timing reference
NZDF	non-zero dispersion fiber
OADM	optical add/drop multiplexer
OAM	operations, administration, maintenance
OBS	optical burst switching
OC	optical carrier
Och	optical channel
ODU	Och data unit
OIF	Optical Internetworking Forum
OLSA	optical LSA
OLT	optical line terminator
OMS	optical multiplex section
ONT	optical network terminator
ONU	optical network unit
OPS	optical packet switching
OPU	Och payload unit
OQoS	optical QoS
OSF	offset field
OSI	open system interconnection reference model
OSPF	open shortest path first
OTN	optical transport network
OTS	optical transmission section
OUPSR	optical unidirectional path sharing ring
OUT	Och transport unit

OXC	optical cross connect
PBS	peak burst size
PBX	private branch exchange
PCM	pulse code modulation
PCR	peak cell rate
PDH	plesiochronous digital hierarchy
PDR	peak data rate
PDU	protocol data unit
PE	provider edge
PFI	payload FCS indicator
P-frame	predictive-coded frame
PIM	protocol independent multicast
PLI	payload length indicator
PLOAM	physical layer OAM
PMD	physical medium dependent sublayer
PNNI	private network-network interface or private network node interface
POF	plastic optical fibers
POH	payload overhead
PON	passive optical network
POP	point of presence
PoS	packet over SONET
PPD	partial packet discard
PPP	point-to-point protocol
PPT	packet payload
PSC	packet-switch capable interface
PSTN	public switched telephone network
PTI	payload type indicator
PVC	permanent virtual connection
QAM	quadrature amplitude modulation
QoS	quality of service
RADIUS	remote authentication dial in user service
RARP	reverse address resolution protocol
RDN	routing data node
RFC	request for comments
RM	resource management
ROC	regional operations center
RRO	RECORD_ROUTE object
RSVP	resource reservation protocol
RSVP-TE	resource reservation protocol – traffic engineering
rtPS	real-time polling service
RT-SBR	real-time statistical bit rate
RT-VBR	real-time variable bit rate
SAAL	signaling AAL
SAN	storage area networks
SAR	segmentation-and-reassembly sublayer
SB-ADPCM	sub-band adaptive pulse code modulation
SBR	statistical bit rate

SCCP	signaling connection control part
SCR	sustained cell rate
SDH	synchronous digital hierarchy
SDSL	symmetric DSL
SDT	structured data transfer
SDU	service data unit
SE	shared explicit style
SECBR	severely errored cell block ratio
SEG-SSCS	segmentation and reassembly SSCS
SEL	selector
SID	silence insertion description
SL	signaling link
SNR	signal-to-noise ratio
SOA	semiconductor optical amplifier
SOH	section overhead
SONET	synchronous optical network
SP	signaling point
SPE	synchronous payload envelope
SRLG	shared risk link group
SRTS	synchronous residual time stamp
SS6	signaling system no. 6
SS7	signaling system no. 7
SSADT	service-specific assured data transfer sublayer
SSCF	service-specific connection function
SSCOP	service-specific connection oriented protocol
SSCS	service-specific convergence sublayer
SSMF	standard single-mode fiber
SSSAR	service-specific segmentation and reassembly sublayer
SSTED	service-specific transmission error detection sublayer
STF	start field
STM	synchronous transfer mode
STP	signaling transfer point
STS	synchronous transport signal
SVC	switched virtual connection
TAW	tell and wait
TC	transmission convergence sublayer
TCAP	transaction capabilities application part
TE	terminal equipment
TG	trunk group
tHEC	type head error control
TLV	type-length-value
TNA	transport network administrative
TNE	terminal network element
TOH	transport overhead
TS	throughput sensitive service
TTL	time to live
TUP	telephone user part

UBR	unspecified bit rate
UGS	unsolicited grant service
ULSR	unidirectional line switched ring
UNI	user network interface
UPI	user payload identifier
UPSR	unidirectional path switched ring
USG-AD	unsolicited grant service with activity detection
UUI	user-to-user indication
VCC	virtual channel connection
VCEL	vertical cavity surface emitting laser
VCI	virtual channel identifier
VDSL	very high data rate DSL
VoIP	voice over IP
VoMPLS	voice over MPLS
VPI	virtual path identifier
VPN	virtual private networks
VPC	virtual path connections
VTG	virtual tributary group
WAN	wide area networks
WDM	wavelength division multiplexing
WF	wildcard-filter style
xDSL	x-type digital subscriber line

1

Introduction

This book deals with several different circuit-switching networks and connection-oriented packet-switching networks. These networks, although seemingly different, have all been built around the notion of a *connection*. That is, a connection has to first be set up between two users before they can communicate. Such a connection is set up by allocating network resources to it. The nature of these resources, as will be seen in this book, depends on the type of the network.

The notion of a connection is also prevalent in the IP network and IP-related protocols. For instance, a TCP connection has to be set up before two TCP users can communicate. This type of connection, however, is not the same as the connection in circuit-switching networks and connection-oriented packet-switching networks. For instance, let us consider an IP network that runs over Ethernet. In this case, when two peer TCP protocols set up a connection, the IP routers and the Ethernet switches are not aware of this connection and so do not allocate any resources to it.

In this chapter, we first describe the concept of a connection as used in this book, and then give examples of connections from the circuit-switching and connection-oriented packet-switching networks described in this book. Subsequently, we describe the organization of this book and the scope and objectives of each chapter. Finally, we present some of the well-known national and international standards committees involved with the standardization process of networking architectures and protocols.

1.1 COMMUNICATION NETWORKS

Communication networks can be classified into the following two broad categories: *switched communication networks* and *broadcast communication networks*. As shown in Figure 1.1, switched communication networks are further classified into *circuit-switching networks* and *packet-switching networks*. Circuit switching and packet switching are two different technologies that evolved over a long time. Examples of circuit-switching networks are the telephone network and the wavelength routing optical network. Examples of packet-switching networks are the IP network, ATM, frame relay, and MPLS networks. Examples of broadcast communication networks are packet radio networks, satellite networks, and multi-access local networks (such as the Ethernet).

Packet-switching networks are further classified as *connection-oriented networks* and *connectionless networks*. Examples of connection-oriented networks are: X.25, ATM, frame relay, and MPLS. The prime example of a connectionless network is the ubiquitous IP network.

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

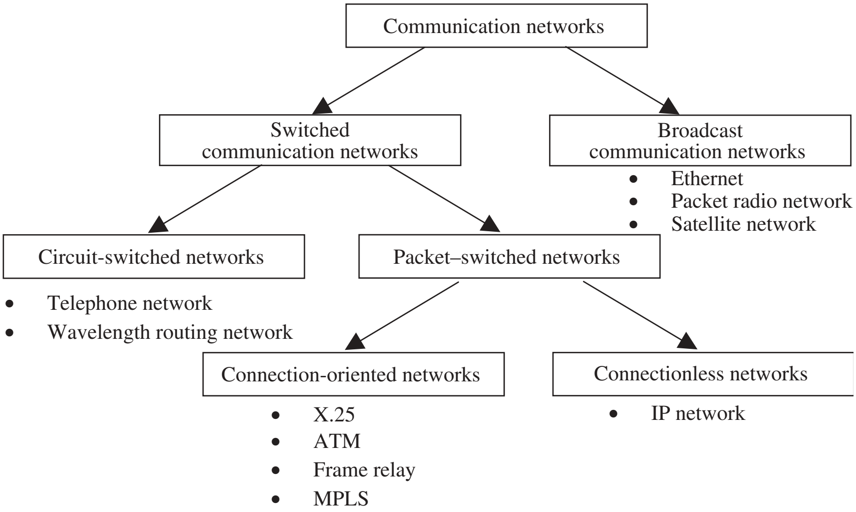


Figure 1.1 A classification of communication networks.

In a circuit-switching network, in order for two users to communicate, a *circuit* or a *connection* has to be first established by the network. Specifically, three phases are involved: *circuit establishment, data transfer,* and *circuit disconnect*. These three phases take place, for instance, when we make a phone call. Circuit establishment takes place when we dial a number. At that moment, the telephone network attempts to establish a connection to the phone of the called party. This involves finding a path to the called party, allocating a channel on each transmission link along the path, and alerting the called party. The data transfer phase follows, during which we converse with the person we called. Finally, the circuit disconnect phase takes place when we hang up. At that moment, the network tears down the connection and releases the allocated channel on each link on the path. The connection is dedicated to the two users for the duration of the call, even when no data is being sent. That is, the channel allocated on each transmission link along the path from our phone to the one we called is not shared with any other phone calls. Also, in order for the call to be established, both stations must be available at the same time.

Circuit switching is a good solution for voice, since it involves exchanging a relatively continuous flow of data. However, it is not a good solution for the transmission of *bursty* data; that is, data that continuously alternates between an active period and a silent period. Transmission of data only takes place when the source is in the active period. Such intermittent data transmission is typical in high-speed networks, and leads to low utilization of the circuit-switching connection.

In packet-switching networks, information is sent in packets, which are passed through the network from node to node until they reach their destination. Error and flow control procedures can be built into the network to assure reliable service. In packet switching, two different techniques (*virtual circuits* and *datagrams*) can be used.

A virtual circuit imitates circuit switching and it involves the same three phases: call setup, transfer of packets, and call termination. In call setup, a connection path is established between the sender and the receiver prior to the transmission of packets. This path, which all packets will follow, goes through the nodes of the packet-switching network. Each packet consists of a header and a payload. The header contains various fields, of which one or more are used to identify the connection that the packet is associated with. This information is used to switch the packet through the network. Unlike circuit switching, however, the channel capacity allocated on each transmission link is not dedicated to the virtual circuit. Rather, the transmission link is shared by all of the virtual circuits that pass through it. Error control assures that all packets are delivered correctly in sequence. Packet-switching networks that employ the virtual circuit switching technique are known as *connection-oriented networks*. Examples of such networks are: ATM, frame relay, and MPLS.

In datagrams, no connection is set up between the two users, and a user can transmit packets whenever the user wants to. Each packet consists of a header and a payload. The header typically contains a number of different fields, including the source address and the destination address. Each packet is individually routed through the network using its destination address. Since the packets are transferred separately, two successive packets transmitted from the same sender to the same receiver could conceivably follow different routes through the network. Since each packet is routed through the network individually, a datagram service can react to congestion easier. Packet-switching networks that employ datagrams are known as *connectionless networks*. The IP network is a prime example of a connectionless packet-switching network. A packet-switching network can be either connection-oriented or connectionless.

A broadcast network has a single communication channel that is shared by all of the stations. There are no switching nodes, as in circuit switching or packet switching. Data transmitted by one station is received by many, and often all, stations. An access control technique is used to regulate the order in which stations transmit. Packet radio networks and satellite networks are examples of a broadcast network. The most widespread example of a broadcast network is the Ethernet. (Currently, the 1-gigabit and 10-gigabit Ethernet is not used as a broadcast network. Ethernet is simply used for unidirectional point-to-point transmission between two users.)

1.2 EXAMPLES OF CONNECTIONS

As mentioned in the previous section, in a circuit-switching network and in a connection-oriented packet-switching network, a connection between two users has to be first set up before they can communicate. The connection is set up by allocating network resources to it. This is in contrast to the connectionless IP network, where a computer can transmit data at any time without setting up a connection to the destination computer.

Note that connections are used in the IP network and IP-related protocols. However, these connections are logical ones between two peer protocols and do not involve allocation of resources in the IP network. For instance, consider an IP network that runs over Ethernet. In this case, when two peer TCP protocols set up a connection, the IP routers and the Ethernet switches are neither aware of this connection nor do they allocate any resources to it. This of course is not the case when IP runs over a connection-oriented packet-switching network such as ATM, as will be seen in Chapter 3. Also, in the case where IP is used with a diffserv, network resource allocation does take place, but just for an aggregate of connections.

In this section, we give examples of connections of the packet-switching and circuit-switching networks described in this book. Specifically, we describe an ATM connection, an MPLS connection, a telephone connection, and a wavelength routing optical network connection.

1.2.1 An ATM Connection

Before we describe an ATM connection, we detour to examine briefly how packets are switched in a connectionless IP network.

Let us assume that Computer A sends IP packets to Computer B, as shown in Figure 1.2. Each IP packet consists of a header and a payload, and the header contains the IP destination address of Computer B. When a packet arrives at IP router 1, the header is examined and the destination address is used in a forwarding routing table in order to find out the next IP router to which the IP packet has to be forwarded. In our example, the next hop router is IP router 2. IP packets arriving at IP router 2 are processed the same way. That is, the destination address is looked up in the router's forwarding routing table in order to identify the next hop. IP router 2 will read that the destination address of the IP packets sent from Computer A is a local address, and it will simply send the IP packets to Computer B. The forwarding routing table in each IP router is constructed using a routing protocol, such as the *open shortest path first (OSPF).*

Let us now contrast the IP procedure for routing IP packets with the scheme used in ATM networks to switch ATM packets (commonly known as *ATM cells*). As will be seen in Chapter 3, an ATM cell has a fixed size of 53 bytes. Of those, 5 bytes are used for the header and the remaining 48 for the payload. For a user to transmit traffic to a destination user over an ATM network, user A first has to request the establishment of a connection, as shown in the example in Figure 1.3. User A sends a SETUP message to ATM switch 1 (to which it is directly connected). The switch calculates a path to the destination ATM user, and then decides whether the path has enough free capacity to accept this new connection. If it does, then the switch forwards the SETUP message to the next switch on the path (switch 2), which in turn has to decide whether to accept the connection, based on how much free capacity it has. If it decides that it can accept the new connection, it forwards the SETUP message to the next switch on the path (switch 3), which forwards the SETUP request to user B. The connection is established when user B returns a CONNECT message, which is propagated all the way back to user A.

The decision as to whether a switch can accept a new connection is crucial to the efficient operation of the network. Each ATM switch tracks all of the connections carried through its switch fabric, the amount of traffic transmitted over each connection, and the *quality of service (QoS)* requested by each connection. The decision to accept a new connection comes down to whether the prospective traffic can be switched according

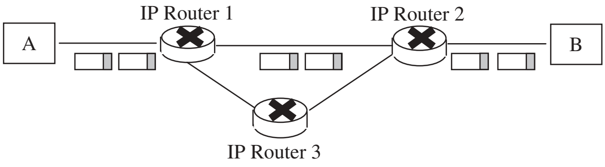


Figure 1.2 Routing IP packets.

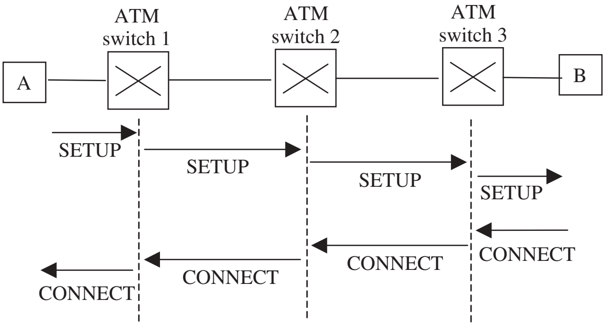


Figure 1.3 Successful establishment of an ATM connection.

to the requested QoS, without affecting the QoS of other existing connections. When a connection is accepted, the switch allocates bandwidth on the outgoing link for the connection. It stops accepting new connections when it runs out of bandwidth, or when it reaches a certain percentage of utilization.

The user starts transmitting ATM cells once it receives the CONNECT message. The ATM cells carry two fields in the header – the *virtual path identifier (VPI)* and the *virtual connection identifier (VCI)* – which are used to identify the connection. The ATM switch uses the combined VPI/VCI value to pass a cell through its switch fabric. Specifically, as in the case of an IP router, an ATM switch maintains a table that specifies the next hop for each VPI/VCI value. When a cell arrives at a switch, the virtual path and virtual connection identifiers check the table for the next ATM switch. The cell is then switched through the switch fabric to the output port that connects to the next ATM switch. The ATM table is considerably smaller than an IP forwarding routing table, since it only contains the existing ATM connections, rather than an entire set of IP addresses.

When user A completes its transmission to B, it tears down the connection by sending a RELEASE message to ATM switch 1. This message is propagated through the switches along the path, and each switch releases the bandwidth it had allocated to the connection.

As we can see, transmitting packets through the IP network is a lot simpler than transmitting cells through an ATM network, since it is not necessary to establish a connection first. On the other hand, by establishing a connection in an ATM network, the network can provide QoS guarantees that are not possible in an IP network.

1.2.2 An MPLS Connection

MPLS introduces a connection-oriented structure into the otherwise connectionless IP network. An MPLS-ready IP router does not forward IP packets based on the destination address in the header. Rather, it forwards them based on a label that is very similar in functionality to the VPI/VCI value carried in the header of an ATM cell.

Let us consider an MPLS-enabled IP network that runs over Ethernet. In this case, a special MPLS header, sandwiched between the IP header and the LLC header, is used. The MPLS header contains a label that is a short, fixed-length connection identifier. The MPLS-ready IP router, known as a *label switched router (LSR),* maintains a table of labels. When an IP packet arrives at the LSR, the label carried in the MPLS header is cross-referenced to the table of labels to find the next hop. The IP packet is then switched to the destination output port of the LSR that connects to the next hop LSR. The table contains labels for only the existing connections, and therefore it is not as large as the forwarding routing table in an IP router.

The procedure is similar to ATM. In order for a user to transmit over an MPLS-enabled IP network, it has to first request the establishment of a connection. This is done using a signaling protocol, such CR-LDP or RSVP-TE. The connection is known in MPLS as a *label switched path (LSP).* As in the case of ATM, an LSR is aware of all of the connections that pass through its switch fabric; therefore, it can decide whether to accept a new connection or not based on the amount of traffic that will be transmitted and the requested QoS. The LSR allocates a portion of its bandwidth to a new connection, and it stops accepting new connections when it either runs out of bandwidth or reaches a certain percentage of utilization.

1.2.3 A Telephone Connection

The telephone network is probably the oldest connection-oriented network. A telephone switch, known as the *central office,* serves many thousands of subscribers. Each subscriber is directly connected to a central office via a dedicated twisted pair line, known as a *local loop*. Central offices are interconnected via *time-division multiplexing (TDM)* links, such as SONET/SDH links and PDH links (i.e., T1, E1, T3, and E3).

Figure 1.4 shows two telephones interconnected via two central offices. For presentation purposes, let us assume that the two central offices are connected via a T1 line. Transmission on a T1 line is organized into frames, with each frame containing 24 time slots. Each time slot is 8 bits long and carries a single voice call. The frame repeats every 128 μsec, meaning that a particular time slot occurs once every 128 μsec (i.e. 8000 times per second). Since it carries 8 bits at a time, the total bit rate of a time slot as it continuously repeats frame after frame is 64 Kbps.

Transmission on a T1 line is unidirectional; that is, data is routed from central office 1 to central office 2. For a bidirectional transmission between the two central offices, two separate T1 lines – each transmitting in the opposite direction – are needed.

In order for subscriber A to talk to subscriber B, a connection has to be first established. This connection is set up by the telephone network when A picks up the receiver and dials the number for the called party. A signaling protocol is used to set up a connection that runs through the central offices that are along the path from subscriber A to subscriber B. The connection involves:

(1) a dedicated line from subscriber A to central office 1;
(2) a time slot (e.g. time slot i) on the T1 line from central office 1 to central office 2; and
(3) a dedicated subscriber line from central office 2 to subscriber B.

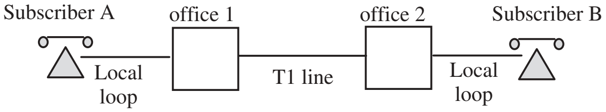


Figure 1.4 A simple telephone network.

In the opposite direction, it involves:

(1) a dedicated line from subscriber B to central office 2;
(2) time slot *i* on the T1 line from central office 2 to central office 1; and
(3) a dedicated subscriber line from central office 1 to subscriber A.

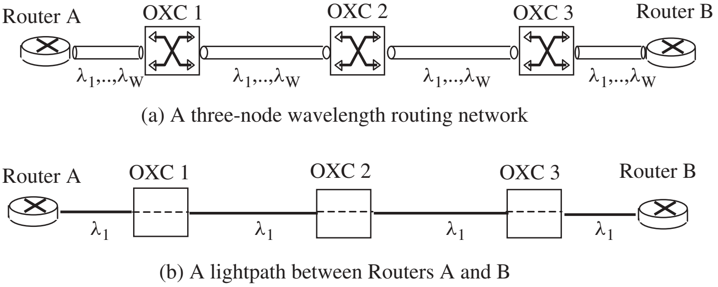
These resources are allocated to the phone call between subscriber A and subscriber B until one of them hangs up. A telephone connection is known as a *circuit*; thus, the telephone network is a circuit-switching network.

1.2.4 A Wavelength Routing Optical Network Connection

Optical networks are based on the *wavelength division multiplexing (WDM)* technology, which combines multiple wavelengths onto the same optical fiber. A *wavelength* is a frequency on which a data stream can be modulated. Each wavelength, therefore, is a separate transmission channel. Transmission over a WDM fiber requires W-independent transmitters. Each transmitter is a light source (e.g. a laser), and is independently modulated with a data stream. The output of each transmitter is an optical signal on a unique wavelength: $\lambda_i, i = 1, 2, \ldots, W$. The optical signals from the W transmitters are combined into a single optical signal at a wavelength multiplexer and transmitted out onto a single optical fiber. At the receiving end, the combined optical signal is demultiplexed into the W individual signals, and each one is then directed to the appropriate receiver, where it is terminated and converted to the electric domain.

A *wavelength routing optical network* consists of *optical cross-connects (OXCs)* interconnected with WDM fibers. An OXC is an $N \times N$ optical switch, with N input fibers and N output fibers. Each fiber carries W wavelengths. The OXC can switch *optically*; that is, all of the incoming wavelengths of its input fibers are switched to the outgoing wavelengths of its output fibers without having to convert the optical signal to an electrical signal. For instance, the OXC can switch the optical signal on incoming wavelength λ_i of input fiber k to the outgoing wavelength λ_i of output fiber m.

A wavelength routing network is a circuit-switching network. That is, in order for a user to transmit data to a destination user, a connection has to be first set up. This connection is a circuit-switching connection, established by using a wavelength on each hop along the connection's path. For example, let us consider that two IP routers (router A and router B) are connected via a three-node wavelength routing network (see Figure 1.5(a)). The



(a) A three-node wavelength routing network

(b) A lightpath between Routers A and B

Figure 1.5 A lightpath.

link from router A to OXC 1, OXC 1 to OXC 2, OXC 2 to OXC 3, and OXC 3 to router B is assumed to be a single fiber with W wavelengths, referred to as $\lambda_1, \lambda_2, \ldots, \lambda_W$. Data is transmitted only in one direction: from router A to router B. Another set of fibers (not shown in Figure 1.5(a)) has to be used in order to transmit data in the opposite direction (i.e. from router B to router A).

Assume that IP router A wants to transmit data to IP router B. Using a signaling protocol, A requests the establishment of a connection to B. The connection between routers A and B is established by allocating the same wavelength, say wavelength λ_1, on all of the links along the path from A to B (i.e., links A to OXC 1, OXC 1 to OXC 2, OXC 2 to OXC 3, and OXC 3 to B). Also, each OXC is instructed to switch λ_1 through its switch fabric transparently. As a result, an optical path is formed from router A to B, over which data is transmitted optically. This optical path is called a *lightpath*, and it connects routers A and B in a unidirectional way from A to B. In order for B to communicate with A, a separate lightpath has to be established in the opposite way over a different set of fibers which are set up to transmit in the opposite direction.

1.3 ORGANIZATION OF THE BOOK

In this book, we explore two connection-oriented packet-switching networks: ATM networks and MPLS-enabled networks. ATM is a legacy network that was developed in the late 1980s and early 1990s. It is used in the backbone to transport IP traffic, in access networks such as ADSL-based networks and *ATM passive optical networks (APON),* and in cellular telephony. The MPLS architecture can be seen as an extension of ATM, and it can be used to introduce QoS in IP networks.

Two circuit-switching networks – SONET/SDH and *optical wavelength routing networks* – are also presented in this book. SONET/SDH has been around for along time, whereas optical wavelength routing networks are relatively new. SONET/SDH is the underlying transport network of the telephone system. It is also used in all modern packet-switching networks, such as IP and ATM. Wavelength routing networks are also circuit-switching networks since the transmission of data is done using optical circuit-switching connections, known as *lightpaths*. We also present a new optical networking scheme, which has not yet been standardized, known as *optical burst switching (OBS)*. This type of optical network can be seen as lying between packet switching and circuit switching.

Finally, the book contains a chapter on access networks, such as ADSL-based networks, cable modems, and passive optical networks, and a chapter on voice over ATM and voice over MPLS.

The book consists of twelve chapters, which cover the following topics:

- Chapter 1: Introduction
- Chapter 2: SONET/SDH
- Chapters 3, 4, and 5: ATM networks
- Chapters 6 and 7: MPLS
- Chapters 8, 9, and 10: Optical networks
- Chapter 11: Access networks
- Chapter 12: Voice over ATM and MPLS

Below, we briefly examine the content of each chapter.

Chapter 2: SONET/SDH and the Generic Frame Procedure (GFP)

In this chapter, we focus on the SONET/SDH transport technology. We first start with a description of T1 and E1, and then we present in detail the SONET/SDH hierarchy, the SONET STS-1 frame structure, overheads, payload, and the SONET STS-3 frame structure.

Subsequently, we describe the SONET/SDH devices and SONET/SDH rings. One of the main features of a SONET/SDH rings is that they are *self-healing*. That is, a SONET/SDH ring can automatically recover when a fiber link fails. Link failure can result from a fiber being accidentally cut, or the optical components that are used to transmit on a fiber fail, or the SONET/SDH switch fails. We describe various architectures for self-healing rings, such as two-fiber and four-fiber protection schemes.

We conclude this chapter with a description of the *generic framing procedure (GFP)* and *data over SONET/SDH (DoS)*. GFP is a lightweight adaptation scheme that permits the transmission of different types of traffic over SONET/SDH and, in the future, over G.709. DoS is a network architecture that uses GFP (together with two other mechanisms) to provide an efficient transport of integrated data services over SONET/SDH.

Chapter 3: ATM networks

The *asynchronous transfer mode (ATM)* architecture was standardized in 1987 by ITU-T as the preferred architecture for the *broadband integrated services data network (B-ISDN)*. ATM is a mature technology that is primarily used in the backbone. For instance, it is widely used in the backbone of *internet service providers (ISPs)* and it has been deployed to provide point-to-point and point-to-multipoint video connections. It is also used in cellular telephony to carry multiple voice connections using the *ATM adaptation layer 2 (AAL 2)*. It is also used for *circuit emulation,* a service that emulates a point-to-point T1/E1 circuit over an ATM network. ATM is also used in access networks such as ADSL-based residential access networks and ATM passive optical networks. ATM is not visible to the networking users, as is the IP/TCP protocol, and because of this, it is often mistaken as a network that it is no longer in use!

The ATM architecture was a novel departure from previous networking architectures; it has built-in mechanisms that permit the transport of different types of traffic with different QoS. Until the advent of *multi-protocol label switching (MPLS)* architecture in the late 1990s, ATM was the only networking technology that provided QoS. From the educational point of view, it is a good idea to develop a working knowledge of ATM and its congestion control schemes before proceeding to MPLS in Chapter 6.

This chapter is organized as follows. We first present the main features of the ATM architecture, such as the structure of the header of the ATM cell, the ATM protocol stack, and the physical layer. Then we briefly describe the ATM shared memory switch architecture, which is the dominant switch architecture, and various scheduling algorithms used to determine the order in which ATM cells are transmitted out. Subsequently, we describe the three *ATM adaptation layers (AAL): AAL 1*, *AAL 2*, and *AAL 5*. We conclude the chapter with a description of *classical IP and ARP over ATM*, a technique standardized by IETF to transport IP over ATM.

Chapter 4: Congestion control in ATM networks

Congestion control is a very important component of ATM networks, as it permits an ATM network operator to carry as much traffic as possible so that revenues can be maximized without affecting the QoS offered to the users.

Two different classes of congestion control schemes have been developed. These schemes are the *preventive congestion control scheme* and *reactive congestion control scheme.* Predictably, the preventive congestion control scheme aims to take a proactive approach to congestion. This is done using the following two procedures: *call (or connection) admission control (CAC)* and *bandwidth enforcement.* CAC is exercised at the connection level and is used to decide whether to accept or reject a new connection. Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the source transmitting on this connection is within its negotiated traffic parameters.

Reactive congestion control is based on a totally different philosophy than preventive congestion control. In reactive congestion control, the network uses feedback messages to control the amount of traffic that an end device transmits so that congestion does not arise.

In this chapter, we first present the parameters used to characterize ATM traffic, the QoS parameters, and the ATM QoS categories. Then, we describe in detail various preventive and the reactive congestion control schemes.

Chapter 5: Signaling in ATM networks

In ATM networks, there are two types of connections: *permanent virtual connections (PVC)* and *switched virtual connections (SVC).* PVCs are established off-line using network management procedures, whereas SVCs are established dynamically in real-time using signaling procedures. In this chapter, we explore the signaling protocol Q.2931 used to set up an SVC. This protocol is used exclusively between a user and the ATM switch to which it is attached. Q.2931 runs on top of a specialized AAL, known as the *signaling AAL (SAAL).* A special sublayer of this AAL is the *service-specific connection oriented protocol (SSCOP).* We first describe the main features of SAAL and SSCOP, and present the various ATM addressing schemes. Then, we discuss the signaling messages and procedures used by Q.2931.

Chapter 6: The multi-protocol label switching architecture

In this chapter, we describe the basic features of the *multi-protocol label switching (MPLS)* architecture. MPLS introduces a connection-oriented structure into the otherwise connectionless IP network. MPLS circumvents the CPU-intensive table look-up in the forwarding routing table necessary to determine the next hop router of an IP packet. Also, it can be used to introduce *quality of service (QoS)* in the IP network. Interestingly enough, since the introduction of MPLS, several CPU-efficient algorithms for carrying out table look-ups in the forwarding routing table were developed. The importance of MPLS, however, was by no means diminished since it is regarded as a solution to introducing QoS in the IP networks.

Chapter 7: Label distribution protocols

MPLS requires a signaling protocol for the reliable establishment of a *label switched path (LSP).* MPLS does not require the use of a single signaling protocol, and in view of this, various protocols have been proposed, of which the *label distribution protocol (LDP)* and the *resource reservation protocol – traffic engineering (RSVP–TE)* are the most popular. Typically, an LSR will run both LDP and RSVP-TE. The two label distribution protocols are not compatible, however. In order to establish a label switched path, one of the two protocols has to be used. In this chapter, we describe LDP, its extension *constraint-based routing label distribution protocol (CR-LDP),* RSVP and RSVP-TE.

Chapter 8: Optical fibers and components

This chapter deals with the physical layer of *wavelength division multiplexing (WDM)* optical networks. We first give a general overview of WDM optical networks. We then proceed to describe how light is transmitted through an optical fiber. Specifically, we discuss the *index of refraction, step-index* and *graded-index* optical fibers, *multi-mode* and *single mode* optical fibers, and various optical effects that occur when light is transmitted through an optical fiber, known as *impairments*. Finally, we conclude this chapter by describing some of the components used in WDM optical networks, such as lasers, optical amplifiers, 2 × 2 couplers and star couplers, and *optical cross-connects (OXCs).*

We note that this chapter is not entirely within the scope of this book, which focuses on layers higher than the physical layer. However, due to the novelty of optical networks, it is important to have some knowledge of the underlying WDM technology. It is not necessary to read this chapter in detail in order to understand the subsequent chapters on optical networks; the key sections to study are the introductory section and the section on components.

Chapter 9: Wavelength routing optical networks

In this chapter, we explore different aspects of a wavelength routing optical networks. We first start with a description of the main features of a wavelength routing network and introduce the ever important concept of a lightpath and the concept of *traffic grooming,* which permits multiple users to share the same lightpath. We also present protection and restoration schemes used to provide carrier grade reliability.

Information on a lightpath is typically transmitted using SONET/SDH framing. Ethernet frames can also be transmitted over an optical network. In the future, it is expected that information will be transmitted over the optical network using the new ITU-T G.709 standard, part of which is described in this chapter. G.709, also known as the *digital wrapper*, permits the transmission of IP packets, Ethernet frames, ATM cells, and SONET/SDH synchronous data.

The rest of the chapter is dedicated to the control plane for wavelength routing networks. We present different types of control plane architectures, and then describe the *generalized MPLS (GMPLS)* architecture and the OIF *user network interface (UNI).* GMPLS is an extension of MPLS and it was designed with a view to applying the MPLS label-switching techniques to *time-division multiplexing (TDM)* networks and wavelength routing networks in addition to packet-switching networks. The OIF UNI specifies signaling procedures for clients to automatically create, delete, and query the status of a connection over a user network interface.

Chapter 10: Optical Burst Switching (OBS)

In a wavelength routing optical network, a connection has to be set up before data will be transmitted. The resources remain allocated to this connection even when there is no traffic transmitted. In view of this, connection utilization can be low when the traffic is bursty. In this chapter, we examine a different optical networking scheme, which is better suited for the transmission of bursty traffic. Because the data is transmitted in bursts, this scheme is known as *optical burst switching (OBS).*

OBS has not yet been standardized, but it is regarded as a viable solution to the problem of transmitting bursty traffic over an optical network. In an OBS network, the user data is collected at the edge of the network, then sorted by destination address, and then grouped into bursts of variable size. Prior to transmitting a burst, a control packet is sent into the optical network in order to set up a bufferless optical connection all of the way to the destination. After a delay, the data burst is transmitted optically without waiting for a positively acknowledgment from the destination node. The connection is set up uniquely for the transmission of a single burst, and is torn down after the burst has been transmitted. That is, a new connection has to be set up each time a burst has to be transmitted through the optical network.

In this chapter, we first present briefly the main features of *optical packet switching*, a technology that preceded OBS. Then, we describe in detail the main features of OBS and present the *Jumpstart* signaling protocol. This is a proof-of-concept protocol developed to demonstrate the viability of OBS.

Chapter 11: Access networks

An access network is a packet-switching network that provides high-speed Internet connectivity to homes. Access networks will also provide additional services, such as *voice over IP (VoIP), voice over ATM (VoATM),* and video on demand. Access networks have different features and requirements than LANs, MANs, and WANs. Currently, there are two different access networks; one is provided over the telephone twisted pair, and the other over the cable network. New access networks, such as the *ATM passive optical network (APON)* and Ethernet-based and wireless-based access networks, are beginning to emerge.

Telephone operators provide currently high-speed access to the Internet over the twisted pair in addition to basic telephone services. Video on demand and voice over IP or ATM will also be provided in the future. A family of modems known as *x-type digital subscriber line (xDSL)* has been developed to provide high-speed access to the Internet over the telephone line. Of the xDSL modems, the *asymmetric DSL (ADSL)* is the most popular one.

Cable operators provide currently access to the Internet, voice over IP, and video on demand over their cable network in addition to the distribution of TV channels. The cable-based access network uses the *data-over-cable service interface specification (DOCSIS).*

The *ATM passive optical network (APON)* is a cost-effective alternative to the telephone-based and cable-based access networks. An APON uses an optical distribution network that consists of optical fibers and passive splitters. It can be used to provide high-speed Internet connection, voice over IP, voice over ATM, and video on demand services.

In this chapter, we describe ADSL-based access networks, cable-based access networks, and the APON. The ADSL-based access network and the APON have been designed to support ATM and consequently they are connection-oriented networks. The cable-based access network supports the IP network. Although the cable-based access network is not a connection-oriented network, it has been included in this chapter for completeness and because of its importance in the access network market.

Chapter 12: Voice over ATM and MPLS

Voice over packet solutions have been developed for the IP network, ATM, frame relay, and MPLS. In this chapter, we explore the topic of voice over ATM and voice over MPLS. Both ATM and MPLS are suitable technologies for voice over packet, since they can provide QoS, a necessary requirement for real-time traffic such as voice.

The ATM Forum has defined several specifications for transporting voice over ATM. These standards can be organized into two groups. The first group of specifications, referred to as *ATM trunking for voice,* deals with the transport of voice over ATM between two telephone networks. The second group of specifications deals with how to provide voice over ATM to a user at a desktop or to a user over ADSL. In this chapter, we describe two of the ATM trunking for voice specifications (*circuit emulation services [CES]* and *ATM trunking using AAL 2 for narrowband services*). Circuit emulation services emulate a TDM link, such as a T1 or E1 link, over an ATM network. The ATM trunking using AAL 2 for narrowband services specification is used to transport voice traffic between two distant private or public telephone networks.

The MPLS and Frame Relay Alliance has so far defined two different specifications for voice over MPLS. These two specifications use ATM's AAL 1 and AAL 2 protocols. The first specification deals with circuit emulation services over MPLS, and it makes use of AAL 1. The second specification deals with the transport of voice over MPLS and it uses AAL 2. Both specifications are described in this chapter.

1.4 STANDARDS COMMITTEES

Standards allow vendors to develop equipment to a common set of specifications. Providers and end-users can also influence the standards so that vendor equipment conforms to certain characteristics. Because of the standardization process, one can purchase equipment from different vendors without being bound to the offerings of a single vendor.

There are two types of standards: *de facto* and *de jure*. De facto standards are those that were first developed by a vendor or a consortium, and then were accepted by the standards bodies. De jure standards are those generated through consensus within national or international standards bodies. ATM and MPLS, for instance, are the result of the latter type of standardization.

Several national and international standards bodies are involved with the telecommunications standardization process, including:

- International Telecommunication Union (ITU)
- International Organization for Standardization (ISO)
- American National Standards Institute (ANSI)
- Institute of Electrical and Electronics Engineering (IEEE)
- Internet Engineering Task Force (IETF)
- ATM Forum
- MPLS and Frame Relay Alliance
- Optical Internetworking Forum (OIF)
- DSL Forum

These standards bodies are described below.

1.4.1 The International Telecommunication Union (ITU)

ITU is a United Nations specialized agency whose job is to standardize international telecommunications. ITU consists of the following three main sections: the *ITU Radiocommunications Sector (ITU-R),* the *ITU Telecommunications Standardization Sector (ITU-T),* and the *ITU Development Sector (ITU-D).*

The ITU-T's objective is the telecommunications standardization on a worldwide basis. This is achieved by studying technical, operating and traffic questions, and adopting recommendations on them. ITU-T was created in March 1993, and it replaced the former well-known standards committee *International Telegraph and Telephone Consultative Committee*, whose origins are over 100 years old. This committee was commonly referred to as CCITT, which are the initials of its name in French.

ITU-T is formed by representatives from standards organizations, service providers, and more recently by representatives from vendors and end users. Contributions to standards are generated by companies, and are first submitted to national technical coordination groups, resulting to national standards. These national coordinating bodies can also pass on contributions to regional organizations or directly to ITU-T, resulting in regional or world standards. ITU more recently started recommending and referencing standards adopted by the other groups, instead of rewriting them.

ITU-T is organized in 15 technical study groups. At present, more than 2500 *recommendations (standards)* or some 55,000 pages are in force. They are nonbinding standards agreed by consensus in the technical study groups. Although, nonbinding, they are generally complied with due to their high quality and also because they guarantee the interconnectivity of networks, and enable telecommunications services to be provided on a worldwide scale.

ITU-T standards are published as *recommendations*, and are organized into series. Each series of recommendations is referred to by a letter of the alphabet. Some of the well-known recommendations are the I, Q, and X recommendations. Recommendations I are related to integrated services digital networks. For instance, I.361 describes the B-ISDN ATM layer specification, I.370 deals with congestion management in frame relay, and I.362.2 describes the ATM Adaptation Layer 2. Recommendations Q are related to switching and signaling. For instance, Q.2931 describes the signaling procedures used to establish a point-to-point ATM switched virtual connection over the private UNI, and Q.2971 describes the signaling procedures used to establish a point-to-multipoint ATM switched virtual connection over the private UNI. Recommendations X are related to data networks and open system communication. For instance, X.700 describes the management framework for the OSI basic reference model, and X.25 deals with the interface between a DTE and a DCE terminal operating in a packet mode and connected to a public data networks by dedicated circuit.

1.4.2 The International Organization for Standardization (ISO)

ISO is a worldwide federation of national standards bodies from some 130 countries, one from each country. It is a nongovernmental organization established in 1947. Its mission is to promote the development of standardization and related activities in the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological and economic activity.

It is interesting to note, that the name ISO does not stand for the initials of the name of this organization, which would have *ISO* been IOS! In fact, it is an acronym derived from the Greek *isos*, which means *equal*. From *equal* to *standard* was the line of thinking that led to the choice of ISO. In addition, the acronym *ISO* is used around the world to denote the organization, thus avoiding a plethora of acronyms resulting from the translation of *International Organization for Standards* into the different national languages of the ISO members, such as IOS in English, and OIN in French (from Organization International de Normalization).

ISO standards cover all technical fields. Well-known examples of ISO standards are: the ISO film speed code, the standardized format of telephone and banking cards, ISO 9000 which provides a framework for quality management and quality assurance, paper sizes, safety wire ropes, ISO metric screw threads, and the ISO international codes for country names, currencies and languages. In telecommunications, the *open system interconnection (OSI)* reference model is a well-known ISO standard.

ISO has cooperated with the *International Electronical Commission (IEC)* to develop standards in computer networks. IEC emphasizes hardware while ISO emphasizes software. In 1987, the two groups formed the *Joint Technical Committee 1 (JTC 1)*. This committee developed documents that became ISO and IEC standards in the area of information technology.

1.4.3 The American National Standards Institute (ANSI)

ANSI is a nongovernmental organization and it was formed in 1918 to act as a cross between a standards setting body and a coordinating body for US organizations that develop standards. ANSI represents the US in international standards bodies such as ITU-T and ISO. ANSI is not restricted to information technology. In 1960, ANSI formed X3, a committee responsible for developing standards within the information processing area in the US. X3 is made up of twenty-five technical committees; X3S3 is the one that is responsible for data communications. The main telecommunications standards organization within ANSI is the T1 secretariat, sponsored by the *Exchange Carriers Standards Association*. ANSI is focused on standards above the physical layer. Hardware oriented standards are the work of the *Electronics Industries Association (EIA)* in the US.

1.4.4 The Institute of Electrical and Electronics Engineering (IEEE)

IEEE is the largest technical professional society in the world, and it has been active in developing standards in the area of electrical engineering and computing through its *IEEE Standards Association (IEEE-SA)*. This is an international organization with a complete portfolio of standards. The IEEE-SA has two governing bodies: the Board of Governors and the Standards Board. The Board of Governors is responsible for the policy, financial oversight, and strategic direction of the Association. The Standards Board has the charge to implement and manage the standards process, such as approving projects.

One of the most well-known IEEE standards body in the networking community is the *LAN/MAN standards committee*, or otherwise known as the *IEEE project 802*. They are responsible for several well-known standards, such as the 802.11b Wireless LAN, the CSMA/CD, token bus, token ring, and the *logical link control (LLC)* layer.

1.4.5 The Internet Engineering Task Force (IETF)

The IETF is part of a hierarchical structure that consists of the following four groups: the *Internet Society (ISOC)* and its Board of Trustees, the *Internet Architecture Board (IAB),* the *Internet Engineering Steering Group (IESG),* and the *Internet Engineering Task Force (IETF)* itself.

The ISOC is a professional society that is concerned with the growth and evolution of the Internet worldwide. The IAB is a technical advisory group of the ISOC, and its charter is to provide oversight of the Internet and its protocols, and to resolves appeals regarding the decisions of the IESG. The IESG is responsible for technical management of IETF activities and the Internet standards process. It administers the standardization process according to the rules and procedures that have been ratified by the ISOC Trustees.

The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is divided into several functional areas: applications, Internet, IP: next generation, network management, operational requirements, routing, security, transport, and user services. Each area has several working groups. A working group is made-up of a group of people who work under a charter in order to achieve a certain goal. Most working groups have a finite lifetime, and a working group is dissolved once it has achieved its goal. Each functional area has one or two area directors, who are members of IESG. Much of the work of IETF is handled via mailing lists, which anyone can join.

The IETF standards are known as *request for comments (RFC),* and each of them is associated with a different number. For instance, RFC 791 describes the *Internet protocol (IP),* and RFC 793 the *transmission control protocol (TCP).* Originally, an RFC was just what the name implies: a request for comments. Early RFCs were messages between the ARPANET architects about how to resolve certain procedures. Over the years, however, RFCs became more formal, and were cited as standards, even when they were not. Within the RFCs, there are two subseries: *for your information (FYI) RFCs* and *standard (STD) RFCs*. The FYI RFC subseries was created to document overviews and topics which are introductory in nature. The STD RFC subseries was created to identify those RFCs that are, in fact, Internet standards.

Another type of Internet document is the *Internet-draft*. These are work-in progress documents of the IETF, submitted by any group or individual. These documents are valid for six months, after which they might be updated, replaced, or become obsolete.

The ISOC has also chartered the *Internet Assigned Numbers Authority (IANA)* as the central coordinator for the assignment of “unique parameters” on the Internet including IP addresses.

1.4.6 The ATM Forum

During the late 80s, many vendors felt that the ATM standardization process in ITU-T was too slow. The ATM Forum was created in 1991 with the objective of accelerating the use of ATM products and services in the private domain through a rapid development of specifications. The ATM Forum is an international nonprofit organization, and it has generated very strong interest within the communications industry.

The ATM Forum consists of the *Technical Committee*, the *Marketing Awareness Committee*, and the *User Committee*.

The ATM Forum Technical Committee works with other worldwide standards bodies selecting appropriate standards, resolving differences among standards, and recommending new standards when existing ones are absent or inappropriate. It was created as one, single worldwide committee in order to promote a single set of specifications for ATM products and services. It consists of several working groups, which investigate different areas of ATM technology.

The activities of the ATM Marketing Awareness Committee include emerging marketing opportunities and educational services.

The ATM Forum User Committee, formed in 1993, consists of organizations which focus on planning, implementation, management or operational use of ATM based networks, and network applications. This committee interacts regularly with the Marketing Awareness Committees and the Technical Committee in order to ensure that ATM technical specifications meet real-world end-user needs.

1.4.7 The MPLS and Frame Relay Alliance

This alliance was created in April 2003 by merging two different organizations (the *Frame Relay Forum* and the *MPLS Forum*). The Frame Relay Forum was formed in 1991. It was an association of vendors, carriers, users, and consultants who were committed to implementing frame relay in accordance with national and international standards. The MPLS Forum was founded in March 2000 with the objective of advancing the deployment of MPLS.

The MPLS and Frame Relay Alliance is an industry-wide association of networking and telecommunication companies focused on advancing the deployment of multi-vendor multiservice label switching networks and associated applications. The Forum provides a meeting ground for companies that are creating MPLS and frame relay products and deploying MPLS and frame relay networks and services.

Through the efforts of its three working committees, the Alliance encourages: (a) input to the development of standards throughout the various industry standards groups; (b) the creation of Implementation Agreements, based upon appropriate standards, on how to build and deliver MPLS and frame relay networks and services; (c) the definition of interoperability test suites and coordination of interoperability events to demonstrate the readiness of MPLS for network deployments; (d) the creation and delivery of educational programs to educate the industry about MPLS and frame relay technologies, services and solutions; and (e) building the awareness of MPLS as a technology ready for wide-scale deployment within service provider networks to deliver profitable services to the end-user community.

1.4.8 The Optical Internetworking Forum (OIF)

OIF was founded in 1988. It is the only industry group uniting representatives from data and optical networks. Its goal is the development and deployment of interoperable products and services for data switching and routing using optical networking technologies. OIF consists of working groups that focus on different topics, such as architecture; physical and link layer; *operations administration, maintenance and provisioning (OAM&P);* signaling; carrier-related requirements; and interoperability.

OIF encourages cooperation among telecom industry participants, such as equipment manufacturers, telecom service providers, and end users, and it promotes global development of optical internetworking products. It also encourages input to appropriate national and international standards bodies, and identifies, selects, and augments as appropriate and publishes optical internetworking specifications drawn from national and international standards.

1.4.9 The DSL Forum

The ADSL Forum was established in 1994; in 1999, it was renamed to the DSL Forum. The goals of the Forum is to provide a complete portfolio of digital subscriber line technologies designed to deliver ubiquitous broadband services for a wide range of situations and applications.

The Forum consists of the technical working groups: architecture and transport, autoconfiguration, operations and network management, testing and interoperability, and *voice over DSL (VoDSL).* It also consists of the marketing working groups: ambassador program, deployment council, marketing interoperability and qualification, summit and best practices, and tradeshows.

Outcomes of the DSL Forum's work are published as technical reports and are available through the Forum's Web site.

PROBLEMS

1. Visit the Web sites of some of the Standards bodies. Familiarize yourself with their organizational structure and the type of standards that are available on these Web sites.
2. Let us assume that the routing tables in an IP administrative domain remain unchanged for a long time. In this case, the path between any two computers, say A and B, within the same domain will not change. Therefore, all of the packets sent from A to B will follow the same path. Can we say that during that time, the IP network behaves like a connection-oriented packet-switching network? Why?
3. Explain why bursty data transmission over a circuit-switching network leads to a poor utilization of the connection.

2

SONET/SDH and the Generic Frame Procedure (GFP)

So far, we have witnessed the development of three generations of digital transport technologies for telephony: PDH, SONET/SDH, and G.709. The first generation of digital transport technology was the *plesiochronous digital hierarchy (PDH)*, of which the North American standard T1 and the ITU-T equivalent standard E1 are probably the most well-known transport schemes. T1/E1 was first deployed in the early 1960s to transport voice traffic.

The *synchronous optical network (SONET)* was proposed by Bellcore (now Telecordia) in 1985, and it can be seen as the second generation of digital transport networks. SONET was designed to multiplex PDH signals and transmit them optically between equipment made by different manufacturers. SONET was not designed, however, to address the needs of the European community, which used the ITU-T plesiochronous digital hierarchy. In view of this, ITU-T adopted the *synchronous digital hierarchy (SDH)* as the international standard. SONET is compliant with SDH. SONET and SDH were also defined to carry ATM cells and PPP and HDLC frames.

The third generation digital transport network is the ITU-T standard G.709, otherwise known as the *digital wrapper*. This is a new standard that takes advantage of the *wavelength division multiplexing (WDM)* technology. It can carry IP packets, ATM cells, Ethernet frames, and SONET/SDH synchronous traffic.

In this chapter, we focus on the SONET/SDH transport technology. The G.709 standard is described in Section 9.3, since the reader is required to have knowledge of the WDM optical technology. We first start with a description of T1 and E1, and then we present in detail the SONET/SDH hierarchy, the SONET STS-1 frame structure, overheads, payload, and the SONET STS-3 frame structure. Subsequently, we describe the SONET/SDH devices and SONET/SDH rings.

One of the main features of SONET/SDH rings is that they are *self-healing*. That is, a SONET/SDH ring can recover automatically when a fiber link fails. This failure can occur when a fiber is accidentally cut, when the optical components used to transmit on a fiber fail, or the SONET/SDH switch fails. We will describe various architectures for self-healing rings, such as two-fiber and four-fiber protection schemes.

We conclude this chapter with a description of the *generic framing procedure (GFP)* and *data over SONET/SDH (DoS)*. GFP is a lightweight adaptation scheme that permits the transmission of different types of traffic over SONET/SDH and, in the future, over

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

G.709. DoS is a network architecture that uses GFP (together with two other mechanisms) to provide an efficient transport of integrated data services over SONET/SDH.

2.1 T1/E1

Time-division multiplexing permits a data link to be used by many sender/receiver pairs (see Figure 2.1). A multiplexer combines the digital signals from *N* incoming links into a single composite digital signal, which is transmitted to the demultiplexer over a link. The demultiplexer then breaks out the composite signal into the *N* individual digital signals and distributes them to their corresponding output links. In the multiplexer, there is a small buffer for each input link that holds incoming data. The *N* buffers are scanned sequentially and each buffer is emptied out at the rate at which the data arrives.

The transmission of the multiplexed signal between the multiplexer and the demultiplexer is organized into frames. Each frame contains a fixed number of time slots, and each time slot is preassigned to a specific input link. The duration of a time slot is either a bit or a byte. If the buffer of an input link has no data, then its associated time slot is transmitted empty. The data rate of the link between the multiplexer and the demultiplexer that carries the multiplexed data streams is at least equal to the sum of the data rates of the incoming links. A time slot dedicated to an input link repeats continuously frame after frame, thus forming a *channel* or a *trunk*.

TDM is used in the telephone system. The voice analog signals are digitized at the end office using the *pulse code modulation (PCM)* technique. That is, the voice signal is sampled 8000 times per second (i.e. every 125 μsec), and the amplitude of the signal is approximated by an 8-bit number, thus producing a 64-Kbps stream. At the destination end office, the original voice signal is reconstructed from this stream. Because of this sampling mechanism, most time intervals within the telephone system are multiples of 125 μsec.

The North American standard that specifies how to multiplex several voice calls onto a single link is known as the *digital signal level* standard, or the *DS* standard. This is a generic digital standard, independent of the medium over which it is transmitted. The DS standard specifies a hierarchy of different data rates (see Table 2.1). The nomenclature of this hierarchy is *DS* followed by the level of multiplexing. For instance, *DS0* refers to a single voice channel corresponding to 64 Kbps, while *DS1* multiplexes 24 voice channels and has a data rate of 1.544 Mbps. The higher levels in the hierarchy are integer multiples of the DS1 data rate. The letter *C* stands for concatenation. For instance, the concatenated signal *DS1C* consists of two DS1 signals pasted together for transmission purposes.

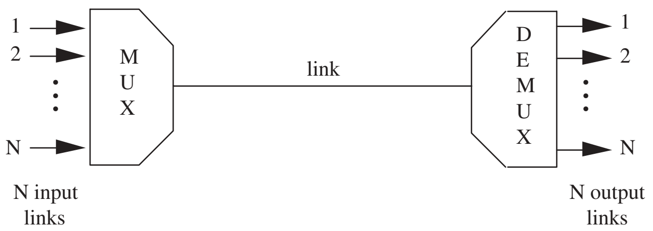


Figure 2.1 Synchronous time-division multiplexing (TDM).

Table 2.1 The North American hierarchy.

Digital signal number	Voice channels	Data rate (Mbps)
DS0	1	0.064
DS1	24	1.544
DS1C	48	3.152
DS2	96	6.312
DS3	672	44.736
DS3C	1344	91.053
DS4	4032	274.176

Table 2.2 The international (ITU-T) hierarchy.

Level number	Voice channels	Data rate (Mbps)
0	1	0.064
1	30	2.048
2	120	8.448
3	480	34.368
4	1920	139.264
5	7680	565.148

The DS standard is a North American standard. The international hierarchy standardized by ITU-T is different, and consists of various levels of multiplexing (see Table 2.2). For instance, Level 1 multiplexes 30 voice channels and has a data rate of 2.048 Mbps; Level 2 multiplexes 120 voice channels and has a data rate of 8.448 Mbps; and so on.

The DS and ITU-T hierarchies are both known as the *plesiochronous digital hierarchy (PDH). Plesiochronous* means *nearly synchronous;* it is derived from the Greek words *plesion,* which means *nearly the same*, and *chronos*, which means *time*.

The digital signal is carried over a *carrier system*, or simply a *carrier*. A carrier consists of a transmission component, an interface component, and a termination component. The T carrier system is used in North America to carry the DS signal, and the E carrier system is used to carry the international digital hierarchy. T1 carries the DS1 signal, T2 the DS2 signal, T3 the DS3 signal, and so on. Similarly, E1 carries the Level 1 signal; E2 carries the Level 2 signal; and so on. Typically, the T and DS nomenclatures are used interchangeably. For instance, one does not distinguish between a T1 line and the DS1 signal. The same applies for the international hierarchy.

In addition to the T and E carrier systems, there is the J system used in Japan; this system is very similar to the T carrier system.

The DS1 format has twenty-four 8-bit time slots and one 1-bit time slot (slot F) for frame synchronization (see Figure 2.2). On the F channel, the frame synchronization pattern 1010101... is transmitted. Each of the twenty-four time slots carries a single

F	Time slot 1	Time slot 2	Time slot 3	. . .	Time slot 24

Figure 2.2 The DS1 format.

64-Kbps voice. For five successive frames, an 8-bit PCM sample is used. In the sixth frame, a 7-bit sample is used, and the eighth extra bit is *robbed* for signaling. The total transmission rate of the DS1 format is $24 \times 8 + 1 = 193$ bits per 125 μsec corresponding to 1.544 Mbps, with each voice channel carrying a 64-Kbps voice.

In the international hierarchy, the Level 1 format for voice consists of thirty-two 8-bit time slots, resulting to a total transmission rate of 2.048 Mbps. Of these time slots, thirty are used for voice and the remaining two are used for synchronization and control.

2.1.1 Fractional T1/E1

Fractional T1 or *fractional E1* allows a user to purchase only a fraction of the T1 or E1 capacity. Fractional T1 services are offered on an $N \times 64$ Kbps or an $N \times 56$ Kbps basis, where $N = 2$, 4, 6, 8, or 12. For example, if $N = 2$, then only two time slots are used per frame, which corresponds to a channel with a total bandwidth of 128 (2×64) Kbps. With fractional T1/E1, users can reduce costs because they pay only for the number of time slots that matches their bandwidth requirements.

2.1.2 Unchannelized Framed Signal

In an unchannelized framed signal, the time slot boundaries are ignored by the sending and receiving equipment. For instance, in the unchannelized T1 framed signal, all 192 bits are used to transport data followed by the 193rd framing bit. This approach permits more flexibility in transmitting at different rates. This scheme is implemented using proprietary solutions. It is also possible to use the entire frame including the framing bit in an unchannelized manner.

2.2 SONET/SDH

The *synchronous optical network (SONET)* was first proposed by Bellcore (now Telecordia) in 1985, and was further developed and standardized by ANSI's T1X1 committee. SONET was designed to multiplex DSn signals and to transmit them optically between equipment made by different manufacturers. SONET was not designed, however, to address the needs of the European community, which used the ITU-T PDH signals. In view of this, ITU-T adopted the *synchronous digital hierarchy (SDH)* as the international standard, which enables the efficient multiplexing of 34.368-Mbps PDH signals (ITU-T's Level 3). SONET is compliant with SDH. SONET and SDH were also defined to carry ATM cells and PPP and HDLC frames.

The information transmitted by SONET/SDH is organized into frames. These frames are transmitted continuously one after the other. Each frame consists of a collection of overhead fields and a payload. SONET/SDH equipment constructs these frames in the electrical domain and then transmits them out optically. At the receiving end, the

SONET/SDH equipment receives the optical signal and converts it to the electrical domain in order to process the frames. The electrical side of the SONET signal is known as the *synchronous transport signal (STS)*, and the electrical side of the SDH is known as the *synchronous transport module (STM)*. The optical side of a SONET/SDH signal is known as the *optical carrier (OC)*.

SONET's basic rate is 51.84 Mbps, whereas the SDH's basic rate is 155.52 Mbps. SONET's basic rate enables the efficient transport of a DS3 signal; SDH's basic rate enables the efficient multiplexing of ITU-T's Level 3 signals. A hierarchy of different *levels* can be constructed (see Table 2.3). The first column gives the *optical carrier level (OC-N);* the next two columns give the equivalent STS and STM levels, respectively. *N* can take a value between 1 and 255. The data rate, overhead rate, and payload rate associated with each level is shown in columns 4, 5, and 6, respectively. In SONET, a data rate is typically referred to by its optical level; in SDH, it is typically referred to by its electrical level. For instance, a *155.520 Mbps data rate* is referred to in SONET as *OC-3*, whereas it is referred to in SDH as *STM-1*.

Not all levels are economically viable; existing products (such as OC-3/STM-1, OC-12/STM-4, and OC-48/STM-16) are indicated by bold formatting. As can be seen, each level can be obtained from the previous one by multiplying it by four.

SONET/SDH is *channelized*. For example, in SONET, STS-3 is constructed by multiplexing three STS-1 basic rate streams; STS-12 is constructed by multiplexing twelve STS-1 streams; and so on. Likewise, in SDH, STM-4 is constructed by multiplexing four STM-1 basic rate streams; STM-16 is constructed by multiplexing 16 STM-1 streams; and so on. As indicated in the last row of Table 2.3, the STM level for OC-N is obtained by dividing *N* by three.

The data rate of STS-1 is 51.840 Mbps, of which 1.728 Mbps is used for overheads and the remaining 50.112 Mbps for payload information. The data rate for STS-3 is obtained by multiplying the corresponding data rate of STS-1 by three. Likewise, STS-3's overhead and payload data rates can be obtained by multiplying those from STS-1 by three. As

Table 2.3 SONET/SDH hierarchy.

Optical level	SONET level (electrical)	SDH level (electrical)	Data rate (Mbps)	Overhead rate (Mbps)	Payload rate (Mbps)
OC-1	STS-1	–	51.840	1.728	50.112
OC-3	**STS-3**	**STM-1**	**155.520**	**5.184**	**150.336**
OC-9	STS-9	STM-3	466.560	15.552	451.008
OC-12	**STS-12**	**STM-4**	**622.080**	**20.736**	**601.344**
OC-18	STS-18	STM-6	933.120	31.104	902.016
OC-24	STS-24	STM-8	1244.160	41.472	1202.688
OC-36	STS-36	STM-12	1866.240	62.208	1804.932
OC-48	**STS-48**	**STM-16**	**2488.320**	**82.944**	**2405.376**
OC-96	STS-96	STM-32	4976.640	165.888	4810.752
OC-192	**STS-192**	**STM-64**	**9953.280**	**331.776**	**9621.504**
OC-768	**STS-768**	**STM-256**	**39813.120**	**1327.104**	**38486.016**
OC-N	STS-N	STM-N/3	N*51.840	N*1.728	N*50.112

we go up the hierarchy, the percent of the overhead remains constant: it corresponds to 3.33% of the data rate or to 3.45% of the payload rate.

As mentioned above, a channelized STS-N consists of *N* STS-1s. Each STS-1, as demonstrated below, is constructed using a specific combination of DSn and E1 signals. In addition to this channelized structure, it is possible to simply fill the STS-3 payload with ATM cells or IP packets packed in PPP or HDLC frames. The resulting structure is called *concatenated*, and is indicated by adding a *c* after the optical level name (e.g. *OC-3c* and *OC-12c*) and after the electrical level name (*STS-3c* and *STS-12c*). Similar concatenation takes place in SDH. Concatenated SONET/SDH links are commonly used to interconnect ATM switches. In fact, the ATM architecture was originally defined to run at OC-3c. They are also used in *packet over SONET (PoS)*. In PoS, IP routers are interconnected with SONET/SDH links and the IP packets are transported directly over SONET/SDH after they have been encapsulated in PPP or HDLC frames.

2.3 THE SONET STS-1 FRAME STRUCTURE

The SONET STS-1 frame consists of 810 bytes and is transmitted 8000 times per second (i.e., every 125 μsec). This gives a total data rate of 8000 × 810 × 8 bits/sec (i.e., 51.84 Mbps). A single byte (commonly referred to as a *time slot*) in the frame is transmitted 8000 times per second, thus giving a data rate of 64 Kbps, as in the case of a time slot in the DS1 signal. This permits SONET to transmit uncompressed PCM voice.

The SONET STS-1 frame is graphically displayed in a matrix form, consisting of nine rows and 90 columns (see Figure 2.3). Each cell in the matrix corresponds to a byte. Starting from byte 1, the frame is transmitted out row by row. The frame consists of the overhead section and the payload section. The overhead section, called the *transport overhead (TOH)*, occupies the first three columns. The payload section occupies the remaining 87 columns (i.e. columns 4 to 90), and it carries the *synchronous payload envelope (SPE)*.

	1	**2**	**3**	**4**	**5**	**6**	**...**	**90**
1	1	2	3	4	5	6	...	90
2	91	92	93	94	95	96	...	180
3	181	182	183	184	185	186	...	270
4	271	272	273	274	275	276	...	360
5	361	362	363	364	365	366	...	450
6	451	452	453	454	455	456	...	560
7	561	562	563	564	565	566	...	630
8	631	632	633	634	635	636	...	720
9	721	722	723	724	725	726	...	810

Figure 2.3 The SONET STS-1 frame structure.

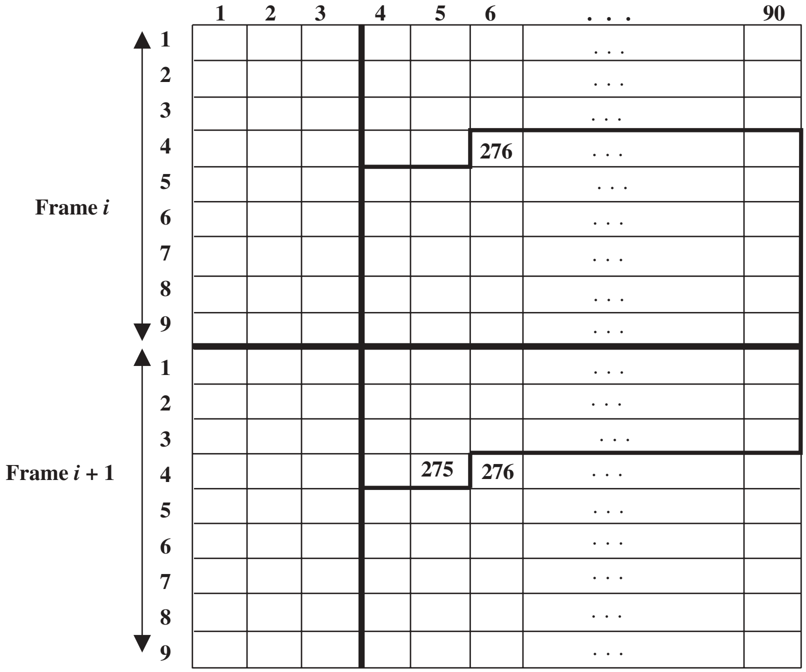


Figure 2.4 An example of the start of the SPE.

The SPE carries a payload of user data and some additional overhead, referred to as the *payload overhead (POH).*

The user is not typically aligned with the transmitted SONET frames. Consequently, the SPE might become available for transmission during the time that a SONET frame is being transmitted out. This problem can be alleviated by buffering the SPE until the beginning of the next SONET frame, and then aligning it with the first byte (row 1, column 4) of the SONET payload. As a result, the SPE will occupy the entire 87 columns of the payload section. This solution requires buffers to hold the SPE until the beginning of the next frame. As the SONET speed increases, the buffering requirements also increase.

An alternative solution to buffering the SPE is transmitting it at the moment that it becomes available. This means that the SPE can start anywhere within the SONET payload, which necessitates the need for a pointer to point to the beginning of the SPE. An example of the start of the SPE is shown in Figure 2.4; the SPE begins on byte 276 (fourth row, sixth column) of frame i, and ends at byte 275 (fourth row, fifth column) of the next frame $i + 1$. The next SPE starts immediately, on byte 276 of frame $i + 1$, and so on. In general, SONET assumes that the SPE can be floated within the payload of the frame, and it provides a pointer in the overhead section for locating its beginning.

The *transport overhead (TOH)* consists of the *section overhead (SOH)* and the *line overhead (LOH)*. Also, as mentioned above, there is a path overhead embedded in the SPE. These overheads are described below.

2.3.1 The Section, Line, and Path Overheads

Let us consider a simple SONET network consisting of SONET devices A1 to A12, A, B, and B1 to B12 (see Figure 2.5). A_i, $i = 1, 2, \ldots, 12$ collects user information and transmits it out to A in STS-1 frames. A multiplexes the twelve STS-1 incoming streams to an STS-12 stream, which is then transmitted to B over two regenerators. B demultiplexes the STS-12 stream to 12 STS-1 individual streams and delivers them to the B_i, $i = 1, 2, \ldots$, 12 devices so that the STS-1 stream from A_i is delivered to B_i, $i = 1, 2, \ldots$, 12. The same happens in the opposite direction, but for the sake of simplicity, we only consider the transmission from left to right.

The quality of the optical signal deteriorates as it travels through the optical fiber, and so has to be periodically regenerated. The distance that an optical signal can travel without requiring regeneration increases continuously as the optical transmission technology evolves. Regeneration is done in the electrical domain. That is, the optical signal at a regenerator undergoes conversion from the optical to the electrical domain; then it is processed; and then it is converted back to the optical domain and transmitted out. Electrical regeneration will eventually be replaced by optical regeneration. In SONET, it is assumed that the regeneration of the optical signal is done in the electrical domain.

In SONET, a single link with a SONET device or a regenerator on either side of it is known as a *section*. A link between two SONET devices (which might include regenerators) is known as a *line*. The sections and lines are indicated in our example in Figure 2.5. The section overhead in the SONET frame is associated with the transport of STS-1 frames over a section, and the line overhead is associated with the transport of SPEs over a line.

SONET is organized into a stack of four layers, all imbedded within the physical layer. The SONET stacks for the SONET network given in Figure 2.5 are shown in Figure 2.6. The lowest layer is the *photonic layer*, which deals with the optical transmission of the STS frames. The next layer up is the *section layer*, which manages the transport of STS frames over the photonic layer. It handles section error monitoring, framing, and signal scrambling; it also employs the section overheads. The *line layer* handles the transport of the SPEs over a line. Its functions include multiplexing and synchronization; it also employs the line overhead. Finally, the *path layer* processes the end-to-end transmission between the points where the SPE originates and terminates. Recall from the example in

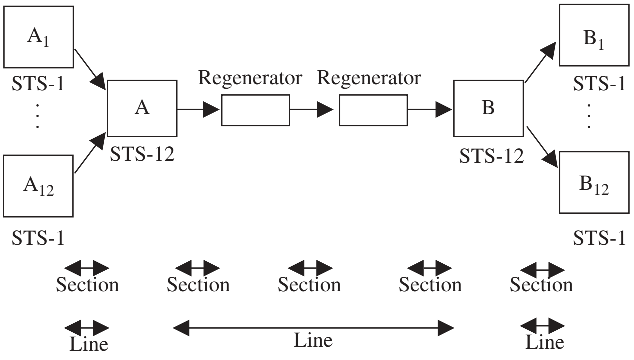


Figure 2.5 A simple SONET network.

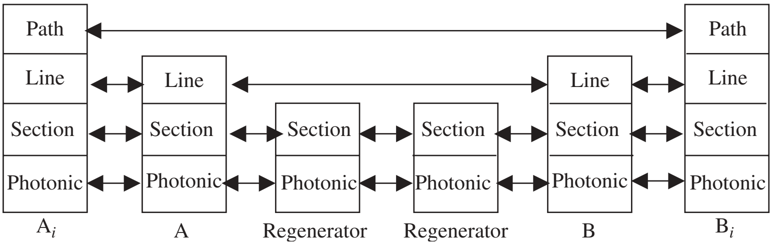


Figure 2.6 The SONET stacks.

Figure 2.5 that each A_i creates SPEs that are delivered to its destination B_i, $i = 1, 2, \ldots$, 12. A path overhead is provided within each SPE that permits the destination B_i to extract the user information. In view of this, there is an association between these two devices at the path layer.

In SONET, the device that can process the section overhead in known as the *section terminating equipment (STE)*. The device that can process the line overhead is known as the *line terminating equipment (LTE)*, and the device that can process the path overhead is known as the *path terminating equipment (PTE)*. See from the examples in Figure 2.5 and Figure 2.6 that device A_i, $i = 1, 2, \ldots$, 12, is an STE, an LTE, and a PTE. Device A, meanwhile, is an STE and an LTE, whereas a regenerator is an STE.

2.3.2 The STS-1 Section, Line, and Path Overheads

As shown in Figure 2.7, the *section overhead (SOH)* bytes occupy the first three rows of the first three columns, and the *line overhead (LOH)* bytes occupy the bottom six rows of the first three columns.

	Column			
Row	1	2	3	
1	A1	A2	J0	SOH
2	B1	E1	F1	
3	D1	D2	D3	
4	H1	H2	H3	LOH
5	B2	K1	K2	
6	D4	D5	D6	
7	D7	D8	D9	
8	D10	D11	D12	
9	Z1	Z2	E2	

Figure 2.7 The section and line overhead bytes.

The following bytes in the section overhead have been defined:

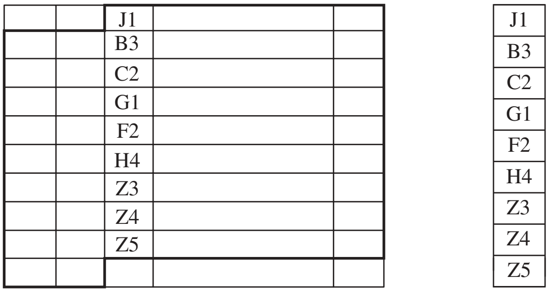
- *A1 and A2*: These two bytes are called the *framing byte*s and are used for frame alignment. They are populated with the value 1111 0110 0010 1000 (i.e. 0xF628), which uniquely identifies the beginning of an STS-frame.
- *J0*: This is called the *section trace* byte and is used to trace the STS-1 frame back to its originating equipment.
- *B1*: This byte is the *bit interleaved parity* byte, commonly referred to as *BIP-8*. It is used to perform an even parity check on the previous STS-1 frame after the frame has been scrambled. The parity is inserted in the BIP-8 field of the current frame before it is scrambled, and is calculated as follows. The first bit of the BIP-8 field performs an even parity check on all of the first-position bits of all of the bytes in the previous frame after it has been scrambled. That is, the sum of the first-position bits in all of the bytes of the previous frame, after it has been scrambled, plus the first bit in the BIP-8 field, has to be an even number. The second bit performs an even parity check on all of the second-position bits of all of the bytes in the previous frame after it has been scrambled, and so on.
- *E1*: This byte provides a 64-Kbps channel that can be used for voice communications by field engineers.
- *F1*: This byte is reserved for use by the network operator.
- *D1, D2, and D3*: These bytes form a data communication channel of 192 Kbps, which is used for network management operations.

The following bytes in the *line overhead (LOH)* have been defined:

- *H1 and H2*: These two bytes are known as the *pointer byte*s. They contain a pointer that points to the beginning of the SPE within the STS-1 frame. The pointer gives the offset in bytes between the H1 and H2 bytes and the beginning of the SPE.
- *H3*: This byte is known as the *pointer action* byte; it is used to compensate for small timing differences that might exist between SONET devices.
- *B2*: This is similar to the B1 byte in the section overhead, and is used to carry the BIP-8 parity check performed on the line overhead section and the payload section. That is, it is performed on the entire STS-1 frame, except for the section overhead bytes.
- *K1 and K2*: These two bytes are used in automatic protection switching.
- *D4 to D12*: These bytes form a data communication channel of 576 Kbps, which is used for network management.
- *Z1 and Z2*: These two bytes have been partially defined.
- *E2*: This byte is similar to the E1 byte in the section overheads.

The *path overhead (POH)*, as mentioned previously, is embedded in the SPE and it occupies the first column (see Figure 2.8). The following bytes have been defined:

- *J1*: This byte is similar to J0 in the section overhead.
- *B3*: This byte is similar to B1 used in the section overhead, and to B2 used in the line overhead. It is used to carry the BIP-8 parity check performed on the payload section. That is, it is performed on the entire STS-1 frame, except for the section and line overhead bytes.



(a) Location of POH (b) The POH bytes

Figure 2.8 The path overhead (POH) bytes.

- *C2*: This byte is known as the *path signal label*. It indicates the type of user information carried in the SPE, such as *virtual tributaries(VT)*, asynchronous DS3, ATM cells, HDLC-over-SONET, and PPP over SONET.
- *G1*: This is known as the *path status* byte; it carries status and diagnostic signals.
- *F2*: This byte is reserved for use by the network operator.
- *H4*: This byte is known as the *multi-frame indicator* and is used to identify payloads carried within the same frame.
- *Z3 and Z4*: These are reserved for future use.
- *Z5*: This byte is used for tandem monitoring. A *tandem* is a telephone switch that is used in the backbone telephone network. It switches traffic between other telephone switches and it does not serve subscribers directly.

2.3.3 The STS-1 Payload

As we have seen above, the STS-1 SPE carries a payload of user data and the path overhead. The payload has been defined to carry multiple *subrate* data streams – that is, streams that are transmitted at rates below that of STS-1 (such as DS1, DS2 and E1 signals). Such a subrate stream is known as a *virtual tributary*. The payload can also carry an entire DS3 signal. In addition to PDH-type signals, the STS-1 payload has been defined to carry ATM cells and IP packets encapsulated in PPP frames. In this section, we examine the different STS-1 payloads that can be constructed.

a) Virtual tributaries

The STS-1 payload is divided into seven groups, known as *virtual tributary groups (VTG)*. Each VTG consists of 108 bytes, which are contained in 12 columns. (Assume that each column always has nine rows.) The seven VTGs take up a total of $12 \times 7 = 84$ columns of the SPE. One of the remaining three columns is used to carry the POH, and the other two are reserved for future use (that is, they go unused). Each VTG can carry a number of virtual tributaries. The following virtual tributaries have been defined:

- *VT1.5*: This virtual tributary carries one DS1 signal, which in itself carries 24 DS0 signals, at 64 Kbps each. VT1.5 is contained in three columns; that is, it takes up 27 bytes. Four VT1.5s can be transported in a single VTG.

- *VT2*: This virtual tributary carries an E1 signal of 2.048 Mbps, which consists of thirty-two 8-bit time slots, of which thirty are used for voice and the remaining two are used for synchronization and control. VT2 is contained in four columns; that is, it takes up 36 bytes. Three VT2s can be carried in a single VTG.
- *VT3*: This virtual tributary transports the unchannelized version of the DS1 signal, where the time slot boundaries are ignored by the sending and receiving equipment. All 192 bits are used to transport data, followed by the 193rd framing bit. It is also possible to use the entire frame (including the framing bit) in an unchannelized manner. The unchannelized version is known as the *concatenated channel*; it is indicated by the capital *C* (as opposed to the lowercase *c* used in SONET). VT3 is contained in six columns; that is, it takes up 54 bytes. This means that a VTG can carry two VT3s.
- *VT6*: This virtual tributary transports a DS2 signal, which carries 96 voice channels. VT6 is contained in twelve columns; that is, it takes up 108 bytes. A VTG can carry exactly one VT6.

The STS-1 payload can only carry one type of virtual tributary. That is, the seven VTGs can only carry VT1.5s, VT2s, VT3s, or VT6s. This means that the STS-1 payload can carry a total of twenty-eight DS1s, twenty-one E1s, fourteen DS1Cs, or seven DS2s. More sophisticated payload construction permits mixing different types of virtual tributaries. For instance, the payload can carry two VTGs (each with one VT6 tributary) and five VTGs (each with four VT1.5s).

b) Asynchronous DS3

The DS3 signal multiplexes 672 voice channels and has a transmission rate of 44.736 Mbps. In the unchannelized format, it is used to carry a continuous bit stream. Both the channelized and unchannelized DS3 are carried in STS-1, and the DS3 signal occupies the SPE's entire payload.

c) ATM cells

ATM cells are directly mapped into the STS-1 SPE so that the ATM bytes coincide with the SONET bytes. The total number of bytes available for user data in the STS-1 frame is: $87 \times 9 = 783$. Of these 783 bytes, 9 bytes are used for the path overhead, leaving 774 bytes for user data. An ATM cell consists of 53 bytes; thus, $774/53 = 14.6$ ATM cells can be stored in an SPE. In other words, the SPE cannot contain an integer number of ATM cells. As a result, an ATM cell might straddle two successive SPEs, with part of it in one SPE, and the rest of it in the next SPE. The ATM cell can be cut off at any byte, whether it is in the header or the payload. ATM cells can also skip past the path overhead bytes.

An example of how ATM cells are mapped in the STS-1 SPE is shown in Figure 2.9. For presentation purposes, we only show one SPE that straddles over two STS-1 frames. The SPE begins on column 10, which means that the path overhead bytes occupy column 10. Assume that the first cell begins immediately after the first path overhead byte on column 11. (The cells are shaded and are not drawn to proportion.) ATM cell 1 occupies row 1, columns 11 to 63. ATM cell 2 occupies the remaining bytes (from row 1, column 64 to row 2, column 27). The path overhead byte on the second row is skipped when the cell is mapped into the SPE. ATM cell 3 is mapped on row 2, columns 28 to 80.

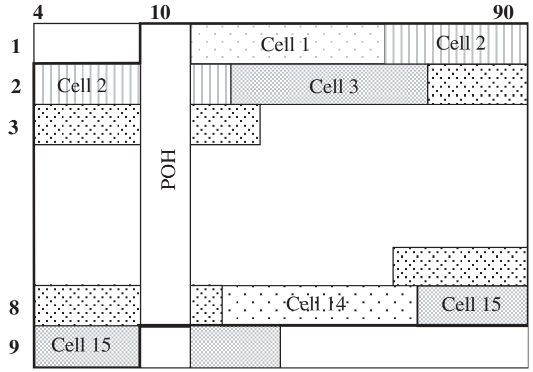


Figure 2.9 Mapping ATM cells in the STS-1 SPE.

ATM cell 14 is mapped on row 8, columns 15 to 67. ATM cell 15 is mapped from row 8, column 67 to row 9, column 31. As we can see, it skips the ninth path overhead byte, and it runs into the next SPE.

ATM users do not always transmit continuously. During the time that ATM cells are not generated, idle cells are inserted so as to maintain the continuous bit stream expected from SONET. This is done in the *transmission convergence (TC)* sublayer of the ATM physical layer. These idle cells can be identified uniquely since their header is marked as: VPI = 0, VCI = 0, PTI = 0, and CLP = 0.

ATM cells are similarly mapped within the SPE of an STS-3c, STS-12c, etc. The basic rate that has been defined for the transport of ATM cells is the STS-3c.

d) Packet over SONET (PoS)

IP packets can be directly carried over SONET links. This scheme is known as *packet over SONET (PoS)*, and is used to interconnect IP routers. IP packets are first encapsulated in HDLC; the resulting frames are mapped, row by row, into the SPE payload, as in the

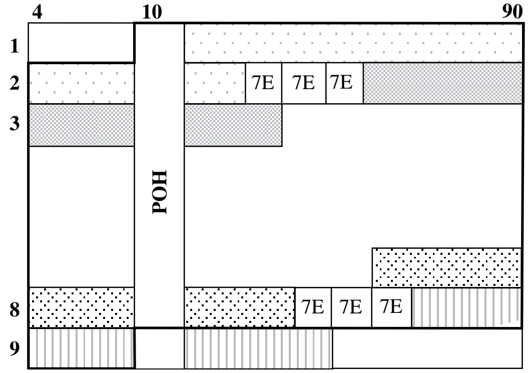


Figure 2.10 Packet over SONET (PoS).

case above for ATM cells. IP packets can also be encapsulated in PPP frames instead of HDLC. The PPP frames are delineated with the HDLC flag 01111110. As is the case with ATM, a frame can straddle over two adjacent SPEs. The interframe fill 7E is used to maintain a continuous bit stream when there are no IP packets to transmit. An example of mapping either HDLC or PPP frames in the SPE payload is shown in Figure 2.10 (note that the frames are not drawn to proportion).

2.4 THE SONET STS-3 FRAME STRUCTURE

The SONET STS-3 frame consists of 2430 bytes and is transmitted 8000 times per second (i.e. once every 125 μsec). This gives a total data rate of 8000 × 2430 × 8 bits/sec (i.e. 155.520 Mbps). The frame is displayed in a matrix form consisting of nine rows and 270 columns, wherein each matrix cell corresponds to a byte.

The channelized STS-3 frame is constructed by multiplexing byte-wise three channelized STS-1 frames. That is, bytes $1, 4, 7, \ldots 268$ of the STS-3 frame contains bytes $1, 2, 3, \ldots 90$ of the first STS-1 frame. Likewise, bytes $2, 5, 8, \ldots 269$ of the STS-3 frame contains bytes $1, 2, 3, \ldots 90$ of the second STS-1 frame, and bytes $3, 6, 9, \ldots 270$ of the STS-3 frame contains bytes $1, 2, 3, \ldots 90$ of the third STS-1 frame. This byte-wise multiplexing causes the columns of the three STS-1 frames to be interleaved in the STS-3 frame (see Figure 2.11). That is, columns $1, 4, 7, \ldots 268$ contain the first STS-1 frame; columns $2, 5, 8, \ldots 269$ contain the second STS-1 frame; and columns $3, 6, 9, \ldots 270$ contain the third STS-1 frame. As a result of this multiplexing, the first nine columns of the STS-3 frame contain the overhead part, and the remaining columns contain the payload part. Error checking and some overhead bytes are for the entire STS-3 frame, and are only meaningful in the overhead bytes of the first STS-1 frame (the corresponding overhead bytes in the other STS-1 frames are ignored).

The STS-3c (OC-3c) was designed originally for the transport of ATM cells. The STS-3c frame consists of nine columns of overhead and 261 columns of payload. ATM cells or IP packets are packed into a single SPE in the same manner as described above in Section 2.3.3.

Higher levels of channelized STS-N are constructed in the same manner as STS-3 by multiplexing N STS-1s.

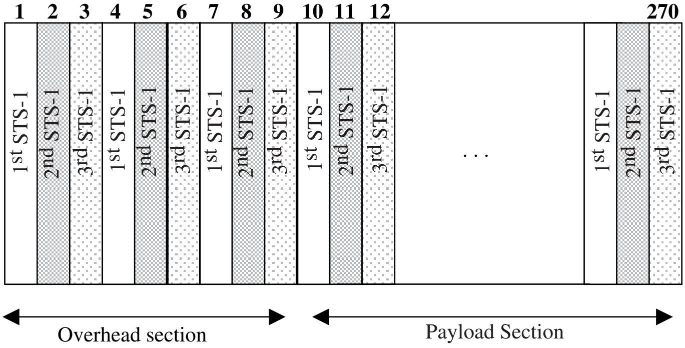


Figure 2.11 The channelized STS-3 frame.

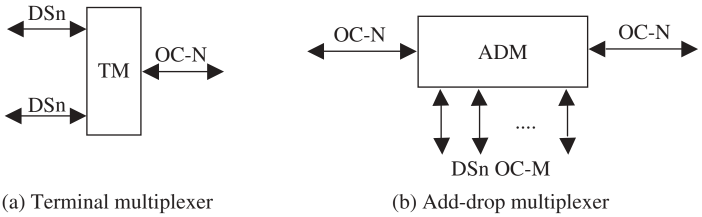
2.5 SONET/SDH DEVICES

Several different types of SONET/SDH devices exist. The SONET/SDH *terminal multiplexer (TM)* device multiplexes a number of low-speed signals into a higher-speed signal. It also works in the opposite direction as a demultiplexer. That is, it demultiplexes a high-speed signal into a number of lower-speed signals. For instance, a SONET TM device multiplexes a number of DSn signals into a single OC-N signal (see Figure 2.12(a)). It can also demultiplex an OC-N signal into a number of DSn signals. It consists of a controller, low-speed interfaces for DSn, an OC-N interface, and a *time slot interchanger (TSI)*, which is used to map the incoming time slots of the DSn signals into the STS-N SPE.

The SONET/SDH *add/drop multiplexer (ADM)* is a more complex version of the TM device. As shown in Figure 2.12(b), a SONET ADM receives an OC-N signal from which it can demultiplex and terminate any number of DSn and/or OC-M signals, where M < N. At the same time, it can add new DSn and OC-M signals into the OC-N signal. Specifically, the incoming OC-N signal is first converted into the electrical domain, and then the payload is extracted from each incoming frame. Recall that the payload consists of a fixed number of bytes, or time slots. These time slots carry different virtual tributaries, such as DSn and OC-M signals, some of which are *dropped* (i.e., terminated). That is, the information is first extracted from the appropriate time slots that carry these virtual tributaries, and then it is transmitted to local users through the ADM's low-speed DSn and OC-M interfaces. (An OC-M interface is typically connected to a TM device.) This termination process frees up a number of time slots in the frame, which, along with other unused time slots, can be used to carry traffic that it is locally generated. That is, DSn and OC-M signals received from its low-speed DSn and OC-M interfaces can be *added* into the payload of the frame using these unused time slots. The final payload is transmitted out at the same SONET level as the incoming OC-N signal.

SONET or SDH ADM devices are typically interconnected to form a SONET or an SDH ring. SONET/SDH rings are *self-healing*; that is, they can automatically recover from link failures. Self-healing rings consist of two or four fibers, and are discussed in the following section.

In Figure 2.13, we show a SONET ring interconnecting four ADM devices. For presentation purposes, we assume that these four ADM devices are connected by a single fiber and that the direction of transmission is clockwise. Each ADM device serves a number of local TM devices and other users. User A is connected to TM 1, which in turn is connected to ADM 1. User A has established a connection to user B, who is attached to ADM 3 via TM 2. In Figure 2.13, this connection is signified by the dotted line. Let us assume that user A transmits a DS1 signal. This is multiplexed with other DS1 signals in



(a) Terminal multiplexer (b) Add-drop multiplexer

Figure 2.12 The SONET TM and ADM.

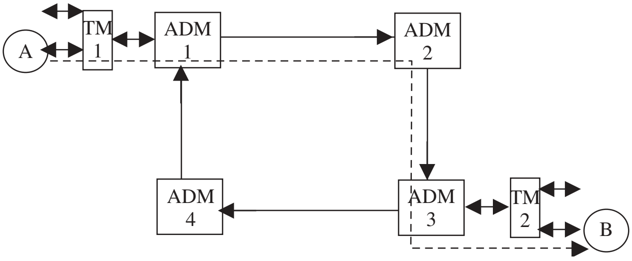


Figure 2.13 A SONET ring.

TM 1, and the output is transmitted to ADM 1. Let us assume that the output signal of TM 1 is an OC-3 signal, and that the speed of the ring is OC-12. ADM 1 adds the OC-3 signal it receives from TM 1 into the STS-12 payload and transmits it out to the next ADM. The OC-12 signal is transmitted to ADM 2, where it is terminated and converted to the electrical domain. ADM 2 adds and drops various signals, and then transmits the resulting STS-12 frames to ADM 3. At ADM 3, the DS1 signal belonging to A is dropped from the payload and transmitted with other signals to TM 2. TM 2 then demultiplexes the signals and transmits A's DS1 signal to B.

The connection from to A to B is a good example of a circuit-switching connection. It is set up manually using network management software and by appropriately configuring each SONET device along the path. The connection is *permanent*, in the sense that it lasts for a long time. The connection is up all of the time, independently of whether A is transmitting continuously to B. A similar connection might also exist from B to A.

SONET/SDH rings are interconnected to cover a wide geographical area via *digital cross connect systems (DCS)*. A DCS is a more complex version of an ADM device. As shown in Figure 2.12, an ADM device receives an OC-N signal from the incoming fiber of the working ring. It then transmits out a new OC-N signal on the outgoing fiber of the ring. A DCS node has a similar functionality, but it is connected to multiple incoming and outgoing OC-N interfaces. For each incoming OC-N signal, it can drop and add any number of DSn and/or OC-M signals, M < N, as in the case of an ADM device. Additionally, it can switch DSn and/or OC-M signals from an incoming interface to any outgoing interface.

Figure 2.14 shows a DCS node interconnecting two rings (Ring 1 and Ring 2). The DCS node receives STS-N frames from Ring 1. For each frame, the DCS node then drops predefined virtual tributaries. It then adds new virtual tributaries – those that are from the local SONET devices (i.e., that are directly attached to the DCS), and those that are from

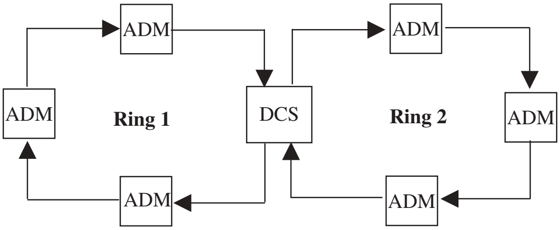


Figure 2.14 A digital cross connect (DCS) node.

Ring 2. The resulting STS-N frames are transmitted out to the adjacent ADM device on Ring 1. The dropped virtual tributaries are either delivered to the local SONET devices or are switched to Ring 2. Likewise, the DCS receives STS-N frames from Ring 2 – from which it drops some virtual tributaries and adds new ones generated from local SONET devices that are attached to the DCS – and from Ring 1. The resulting STS-N frames are transmitted out to the adjacent ADM device on Ring 2. The dropped virtual tributaries are either delivered to the local SONET devices or are switched to Ring 1.

A DCS node is equipped with a switch fabric so that it can switch virtual tributaries from one input interface to an output interface. Specifically, the switch fabric can switch the data carried in one or more time slots of each incoming frame from any input interface to the same number of time slots, but not necessarily in the same position, of the outgoing frame of any output interface. It serves all of the input interfaces simultaneously.

SONET/SDH rings are typically deployed in a metropolitan area, either as *metro edge rings* or as *metro core rings*. A metro edge ring is used to transport traffic between customers and a *hub*, which is a SONET/SDH node that is attached to a metro core ring. Typical customers include: ADSL-based access networks, cable-based access networks, small telephone switches *(private branch exchange*, or *PBX), storage access networks (SAN)*, and enterprise networks. A metro core ring interconnects metro edge rings, large telephone switches, and ISP *points of presence (POP)*. It also sends and receives traffic to and from larger regional and long-haul networks. Traffic demands on a metro core ring are dynamic, unlike a metro edge ring, which has fairly static traffic patterns. Metro core rings are interconnected using DCS nodes to form a mesh network.

2.6 SELF-HEALING SONET/SDH RINGS

SONET/SDH rings have been specially architected so that they are highly reliable. Specifically, they are available 99.999% of the time, which translates to an average downtime for the network of only six minutes per year! One of the main causes for a ring to go down is failure of a fiber link. This can happen when the fiber is accidentally cutoff (backhoe fade), or when the transmission or receiver equipment on the fiber link fail. Also, link failure can occur when a SONET/SDH device fails, although this happens very rarely since these devices have a high degree of redundancy. Fiber cuts due to digging in an area where fiber cables pass through, however, are quite common. SONET/SDH rings are self-healing, so that the ring's services can be automatically restored following a link failure or a degradation in the network signal. This is done using the *automatic protection switching (APS)* protocol. The time to restore the services has to be less than 50 msec.

In this section, we first describe protection schemes for point-to-point SONET/SDH links, and then we describe several self-healing SONET/SDH ring architectures.

The simplest SONET/SDH network is a point-to-point fiber link that connects two SONET/SDH devices. Link protection can be done in a *dedicated* $1 + 1$ manner, or in a *shared 1:1* or a $1 : N$ manner. In the $1 + 1$ scheme, the two devices are connected with two different fibers (see Figure 2.15). One is designated as a *working fiber*, and the other as a *protection fiber*. The SONET/SDH signal is split and then transmitted simultaneously over both fibers. The destination selects the best of the two signals based on their quality. If one fiber fails, the destination continues to receive the signal from the other fiber. The working and protection fibers have to be *diversely routed*. That is, the two fibers use separate conduits and different physical routes. Often, for economic reasons, the two

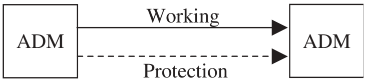
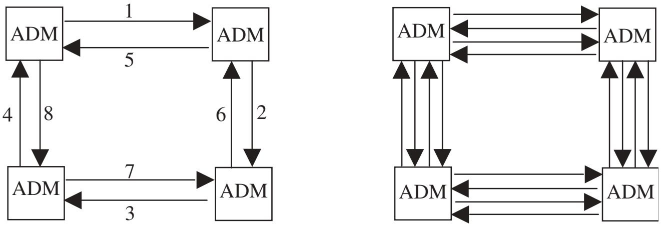


Figure 2.15 The 1 + 1 protection scheme.



(a) Two-fiber ring

(b) Four-fiber ring

Figure 2.16 SONET/SDH rings.

fibers use different conduits, but they use the same physical path. In this case, we say that they are *structurally diverse*.

In the 1:1 scheme, there are still two diversely routed fibers: a *working fiber* and a *protection fiber*. The signal is transmitted over the working fiber. If this fiber fails, then the source and destination both switch to the protection fiber. The 1:N scheme is a generalization of the 1:1 scheme, whereby N working fibers are protected by a single protection fiber. Since there is one protection fiber, only one working fiber can be protected at any time. Once a working fiber has been repaired, the signal is switched back, either automatically or manually, from the protection fiber to the working fiber.

Self-healing SONET/SDH ring architectures are distinguished by the following three features:

- *Number of fibers*: A SONET/SDH ring can consist of either two or four fibers (see Figure 2.16). In the two-fiber ring, fibers 1, 2, 3, and 4 are used to form the *working ring*, and fibers 5, 6, 7, and 8 are used to form the *protection ring*. Transmission on the working ring is clockwise; on the protection ring, it is counter-clockwise (as indicated by the arrows in Figure 2.16). In another variation of the two-fiber ring, each set of fibers (i.e. fibers 1, 2, 3, 4 and fibers 5, 6, 7, 8) form a ring that can function as both a working ring and a protection ring. In this case, the capacity of each fiber is divided into two equal parts: one for working traffic and the other for protection traffic. In a four-fiber SONET/SDH ring, there are two working rings and two protection rings (one per working ring).

  As in the case of a point-to-point SONET/SDH fiber link, the working and protection rings are *route diverse*. That is, the fibers between two adjacent SONET/SDH devices use different conduits and different physical routes. The working and protection rings can also be structurally diverse, which is typically more economical. In this case, the fibers between two adjacent SONET/SDH devices use different conduits, but they follow the same physical path.

- *Direction of transmission*: A SONET/SDH ring can be *unidirectional* or *bidirectional*. In a unidirectional ring, signals are only transmitted in one direction of the ring; in a bidirectional ring, signals are transmitted in both directions.
- *Line or path switching*: Protection on a SONET/SDH ring can be at the level of a *line* or a *path*. Recall from Section 2.3.1 that a *line* is a link between two SONET/SDH devices and might include regenerators. A *path* is an end-to-end connection between the point where the SPE originates and the point where it terminates. (Note that Section 9.2 refers to *line protection* as *link protection*.) *Line switching* restores all of the traffic that pass through a failed link, and *path switching* restores some of the connections that are affected by a link failure.

Based on these three features, we have the following two-fiber or four-fiber ring architectures: *unidirectional line switched ring (ULSR), bidirectional line switched ring (BLSR), unidirectional path switched ring (UPSR)*, and *bidirectional path switched ring (BPSR)*. Of these rings, the following three are currently used: *two-fiber unidirectional path switched ring (2F-UPSR), two-fiber bidirectional line switched ring (2F-BLSR)*, and *four-fiber bidirectional line switched ring (4F-BLSR)* These three ring architectures are discussed below in detail.

2.6.1 Two-fiber Unidirectional Path Switched Ring (2F-UPSR)

This ring architecture, as its name implies, consists of two fibers with unidirectional transmission and path switching. Figure 2.17 shows an example of this ring architecture type. The working ring consists of fibers 1, 2, 3, and 4; the protection ring consists of fibers 5, 6, 7, and 8. The ring is unidirectional, meaning that traffic is transmitted in the same direction. That is, A transmits to B over fiber 1 of the working ring, and B transmits over fibers 2, 3, and 4 of the working ring. Protection is provided at the path level using a scheme similar to the $1 + 1$ described above. That is, the signal transmitted by A is split into two; one copy is transmitted over the working fiber (fiber 1), and the other copy is transmitted over the protection fibers (fibers 8, 7, and 6). During normal operation, B receives two identical signals from A and selects the one with the best quality. If fiber 1 fails, then B will continue to receive A's signal over the protection path. The same applies if there is a node failure.

This is a simple ring architecture; it is used as a metro edge ring to interconnect PBXs and access networks to a metro core ring. Typical transmission speeds are OC-3/STM-1 and OC-12/STM-4. The disadvantage of this ring architecture is that the maximum amount of traffic it can carry is equal to the traffic it can carry over a single fiber.

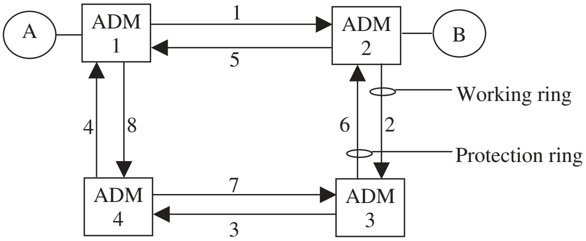


Figure 2.17 An example of a 2F-UPSR.

2.6.2 Two-fiber Bidirectional Line Switched Ring (2F-BLSR)

This is a two-fiber ring with bidirectional transmission and line switching. It is used in metro core rings. As shown in Figure 2.18, fibers 1, 2, 3, 4, 5 and 6 form a ring (Ring 1), on which transmission is clockwise. Fibers 7, 8, 9, 10, 11, and 12 meanwhile form another ring (Ring 2), on which transmission is counter-clockwise. Unlike the 2F-UPSR, both Rings 1 and 2 carry working and protection traffic. This is done by dividing the capacity of each fiber on Rings 1 and 2 into two parts. One part is used to carry working traffic, and the other part to carry protection traffic. For instance, let us assume that the transmission speed of each fiber is OC-12/STM-4. Then, two OC-3/STM-1s are allocated to working traffic and the other two to protection traffic. Since only two OC-3/STM-1s can be used for working traffic, the maximum capacity that the 2F-BLSR can carry over both Rings 1 and 2 is OC-12/STM-4. The capacity allocated to protection traffic on either Rings 1 and 2 can be used to carry low priority traffic. This traffic can be preempted in case of failure of a fiber.

The ring is bidirectional, which means that a user can transmit in either direction. That is, it can transmit on either Ring 1 or Ring 2, depending on the route of the shortest path to the destination. In view of this, under normal operation, A transmits to B over the working part of fibers 1 and 2 of Ring 1, and B transmits to A over the working part of fibers 8 and 7 of Ring 2.

Assume that fiber 2 fails. Since the ring provides line switching, all of the traffic that goes over fiber 2 will be automatically switched to the protection part of Ring 2. That is, all of the traffic will be rerouted to ADM 3 over the protection part of Ring 2 using fibers 7, 12, 11, 10, and 9. From there, the traffic for each connection will continue on following the original path of the connection. For instance, let us consider a connection from A to C, as indicated in Figure 2.18 by the solid line. When fiber 2 fails, the traffic from A will be rerouted, as shown in Figure 2.18 by the dotted line. At ADM 3, the traffic will continue along the same path as the original connection. That is, it will be routed to back to ADM 4 over fiber 3.

If both fibers 2 and 8 fail, then the traffic transmitted on fiber 2 from ADM 2 will be switched to the protection part of Ring 2, and the traffic transmitted from ADM 3 on fiber 8 will be switched to the protection part of Ring 1.

2.6.3 Four-fiber Bidirectional Line Switched Ring (4F-BLSR)

This is a four-fiber ring with bidirectional transmission and line switching. There are four fibers between two adjacent SONET/SDH devices; they form two working rings

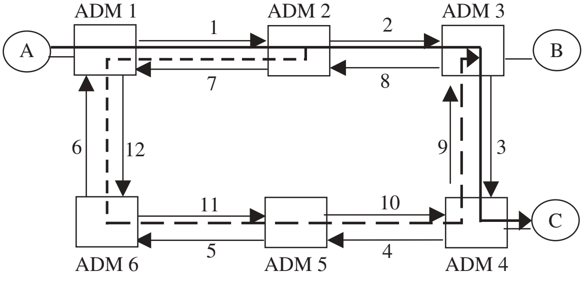


Figure 2.18 An example of a 2F-BLSR.

and two protection rings (see Figure 2.19). The two working rings transmit in opposite directions, one clockwise and the other counter-clockwise. Each working ring is protected by a protection ring that transmits in the same direction. The protection rings can either be idle or carry low priority traffic, which can be preempted in case of a fiber failure. The advantage of this four-fiber ring over the two-fiber ring described above is that it can suffer multiple failures and still function. In view of this, it is deployed by long-distance telephone companies in regional and national rings.

The ring is bidirectional, which means that a user can transmit in either direction. That is, it can transmit on either of the working rings, depending on the shortest path route to the destination. For instance, user A transmits to B via ADM 2 over the clockwise working ring, and B transmits to A via ADM 2 over the counter-clockwise working ring.

If a working fiber fails, then the working traffic will be transferred over its protection ring. This is known as *span switching*. For instance, let us assume that the working fiber between ADM devices 2 and 3 on the clockwise protection ring fails. Then, all of the working traffic will be diverted to the protection fiber between ADM device(s) 2 and 3 of the clockwise protection ring (as shown by the solid line in Figure 2.20).

Often, the two working fibers are part of the same bundle of fibers. Therefore, both might be cut at the same time. The same applies to the protection fibers. When the two working fibers between two SONET/SDH devices are both cut, then the working traffic will be switched automatically to the protection fibers. It is also possible that the working fibers and the protection fibers might not be too far apart, and that they all might get cut at the same time. In this case, the traffic will be switched to the protection fibers, known as *ring switching* (see Figure 2.21).

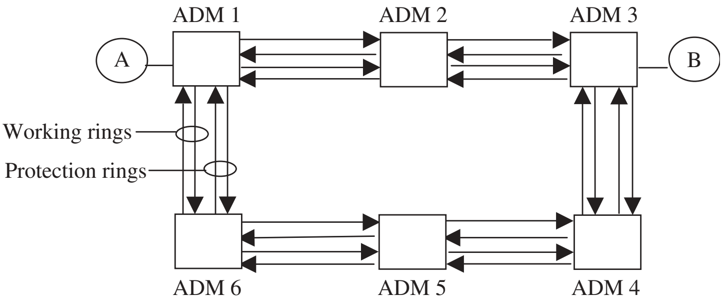
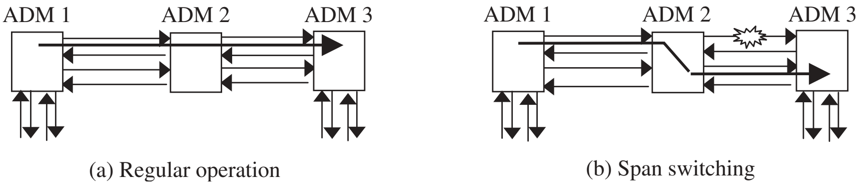


Figure 2.19 An example of a 4F-BLSR.



(a) Regular operation

(b) Span switching

Figure 2.20 Span switching between ADM devices 2 and 3.

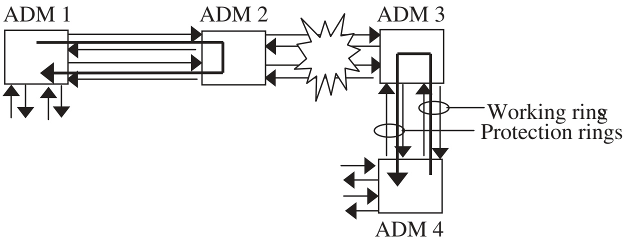
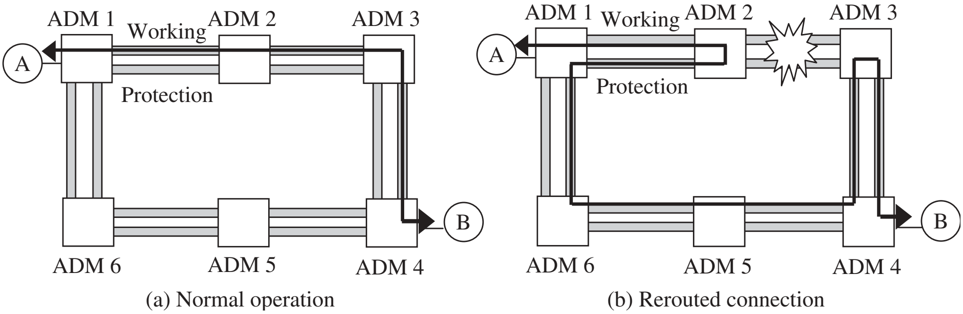


Figure 2.21 An example of ring switching.



(a) Normal operation

(b) Rerouted connection

Figure 2.22 Rerouting of a connection.

It is instructive to see how a connection is rerouted as a result of ring switching. Let us assume that users A and B have established a two-way connection (see the solid line in Figure 2.22(a)). The pair of working fibers and the pair of protection fibers is each shown as a single bundle. When all four fibers between ADM 2 and 3 are cut, the ring will automatically reroute the traffic (see Figure 2.22(b)).

2.7 THE GENERIC FRAMING PROCEDURE (GFP)

As we have seen, SONET/SDH has been optimized to carry voice traffic. It has also been defined to carry ATM traffic and *IP packets (PoS)*. GFP is a simple adaptation scheme that extends the ability of SONET/SDH to carrying different types of traffic. Specifically, it permits the transport of frame-oriented traffic, such as Ethernet and IP over PPP. It also permits continuous-bit-rate block-coded data from *storage area networks (SAN)* transported by networks, such as *fiber channel, fiber connection (FICON)*, and *enterprise system connect (ECON)*. GFP is the result of a joint standardization effort of ANSI, Committee T1X1.5, and ITU-T, and is described in ITU-T recommendation G.7041.

GFP consists of both client-independent and client-dependent aspects, as shown in Figure 2.23. The client-independent aspects apply to all GFP adapted traffic. The client-independent aspects also cover functions such as GFP frame delineation, data link synchronization and scrambling, client PDU multiplexing, and client-independent performance monitoring. The client-dependent aspects of GFP cover functions such as mapping the client PDUs into the GFP payload; client-specific performance monitoring; and *operations,*

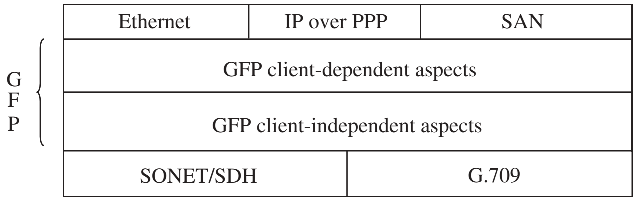


Figure 2.23 The GFP stack.

administration, and management (OA&M). The resulting GFP frames are transmitted over SONET/SDH, or over the new ITU-T optical transport G.709 (otherwise known as the *digital wrapper*). G.709 is described in detail in Section 9.3.

2.7.1 The GFP Frame Structure

Two different types of GFP frames have been defined: *GFP client frames* and *GFP control frames*. The GFP client frames can be either *client data frames* or *client management frames*. GFP client data frames are used to transport client data, and GFP client management frames are used to transport information with the management of the client signal or the GFP connection.

The GFP frame structure, as shown in Figure 2.24, consists of the *GFP core header* and the *payload area*. The GFP core header consists of the following fields:

- *Payload length indicator (PLI)*: A 2-byte field used to indicate the size of the payload area in bytes. Values between 0 and 3 are reserved for internal usage.
- *Core head error control (core HEC or cHEC)*: A 2-byte field used to protect the PLI field. It carries the *frame check sequence (FCS)* obtained using the standard CRC-16, which enables single-bit error correction and multiple-bit error detection.

The GFP payload area consists of the *payload header*, the *payload*, and an optional payload *frame check sequence (FCS)*. (See Figure 2.24.) The payload header has a variable

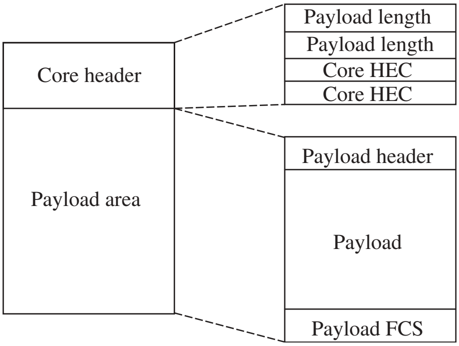


Figure 2.24 The GFP frame structure.

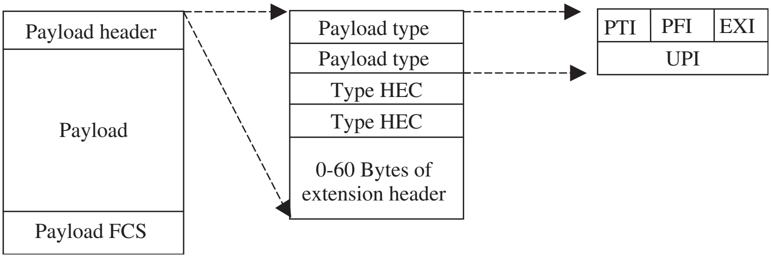


Figure 2.25 The GFP payload structure.

length between 4 bytes and 64 bytes (see Figure 2.25). The following fields have been defined:

- *Payload type*: A mandatory 2-byte field that indicates the content and format of the payload. The following subfields have been defined within the payload type:
  - ○ *Payload type identifier (PTI)*: A 3-bit subfield that identifies the type of client frame (i.e., client data frame) and client management frame.
  - ○ *Payload FCS indicator (PFI)*: A 1-bit subfield that indicates the presence or absence of the optional payload FCS.
  - ○ *Extension header identifier (EXI)*: A 4-bit subfield that identifies the type of extension header.
  - ○ *User payload identifier (UPI)*: An 8-bit field that identifies the type of payload. Defined UPI values for client data frames include:
    - ■ Frame-mapped Ethernet
    - ■ Frame-mapped PPP (including IP and MPLS)
    - ■ Transparent-mapped Fiber Channel
    - ■ Transparent-mapped FICON
    - ■ Transparent-mapped ESCON
    - ■ Transparent-mapped *Gigabit Ethernet (GbE)*
- *Type head error control (type HEC or tHEC)*: A 2-byte field that protects the payload header. It carries the FCS obtained using standard CRC-16. As with the core HEC, it enables both single-error correction and multiple-error detection.
- *Extension headers*: A flexible mechanism for header extension is supported in order to facilitate adaptation of GFP to diverse transport mechanisms.

The payload contains a GFP frame. It is a variable-length area (ranging from 0 bytes to 65,535 bytes), minus the size of the payload header and (if present) the size of the payload FCS. Finally, the GFP payload FCS consists of an optional 4-byte FCS generated using CRC-32.

2.7.2 GFP Client-independent Functions

GFP supports the following basic procedures, which are common to all payloads: frame delineation, frame multiplexing, header and payload scrambling, and client management.

Frame delineation checks the GFP frames to make sure they are extracted correctly from the bit stream that SONET/SDH delivers to the GFP client-independent layer. It is a simple procedure that makes sure that the beginning of each GFP frame is correctly identified. This procedure is identical to the cell delineation procedure used in ATM (see Section 3.4.1), and it uses the core HEC.

Frame multiplexing is used by GFP to multiplex client data frames and client management frames on a frame-by-frame basis. Client data frames have priority over client management frames. When no frames are available, GFP inserts idle frames in order to maintain a continuous bit flow. The GFP idle frame is a special 4-byte GFP control frame, consisting only of the GFP core header with the payload length indicator and the core HEC fields, set to 0.

The core header is always scrambled to ensure high bit transmission density when transmitting idle GFP frames. Payload scrambling is done using a $1 + x^{43}$ self-synchronous scrambler. As in ATM, scrambling is enabled starting at the first transmitted byte after the cHEC field, and disabled after the last transmitted byte of the GFP frame.

Client management provides a generic mechanism to propagate client-specific information, such as performance monitoring and OA&M information.

2.7.3 GFP Client-dependent Functions

The client data can be carried in GFP frames using one of the following two adaptation modes: *frame-mapped GFP (GFP-F)* or *transparent-mapped GFP (GFP-T)*. Frame-mapped GFP is applicable to most packet data types, and transparent-mapped GFP is applicable to 8B/10B coded signals. In frame-mapped GFP, each received client-frame is mapped in its entirety in a single GFP payload. Examples of such client frames include Ethernet MAC frames, PPP/IP packets, and HDLC-framed PDUs.

The transparent-mapped GFP mode is used to transport continuous bit-rate, 8B/10B block-coded client data and control information carried by networks, such as fiber channel, ESCON, FICON, and Gigabit Ethernet. (Note that in the 8B/10B encoding scheme, each group of eight bits is coded by a 10-bit code. Coding groups of bits is known as *block-coding*.) Rather than transporting data on a frame-by-frame basis, the GFP transparent-mapped mode transports data as a stream of characters. Specifically, the individual characters are decoded from their client 8B/10B block codes and then mapped into periodic fixed-length GFP frames using 64B/65B block coding. Specifically, the 10-bit codes are first decoded into their original data or control codeword value, and then the decoded characters are mapped into 64B/65B codes. A bit in the 65-bit code is used to indicate whether the 65-bit block contains only data or whether control characters are also included. Eight consecutive 65-bit blocks are grouped together into a single *super block*, and N super blocks make up a single GFP frame. This procedure reduces the 25% overhead introduced by the 8B/10B block-coding; plus, it reduces latency, which is important for storage-related applications.

2.8 DATA OVER SONET/SDH (DOS)

The *data over SONET/SDH (DoS)* network architecture provides a mechanism for the efficient transport of integrated data services. The following are some of the features of DoS:

- It provides flexible bandwidth assignment with a 50-Mbps granularity.
- No modifications are required of the intermediate nodes.

- Using GFP, it provides an efficient framing scheme with a small overhead.
- It can accommodate IP packets, Ethernet frames, and constant bit rate data and control information carried by fiber channel, ESCON, and FICON. In particular, it provides an effective mechanism to transport GbE, which has recently been widely deployed in *wide area networks (WAN).*
- Coexistence of the traditional voice services and the new data services in the same SONET/SDH frame.
- Network management through the SONET/SDH existing and quality-proven network management.

DoS uses three technologies: *generic framing procedure (GFP), virtual concatenation*, and *link capacity adjustment scheme (LCAS).* These technologies have been standardized by ITU-T. GFP was described in detail above in Section 2.7. Below, we describe the other two technologies: virtual concatenation and LCAS.

2.8.1 Virtual Concatenation

Virtual concatenation is a SONET/SDH procedure that maps an incoming traffic stream into a number of individual subrate payloads. That is, payloads with a bandwidth less than the bandwidth of a SONET/SDH link. The subrate payloads are switched through the SONET/SDH network independently of each other.

As an example, let us consider the case of transporting a GbE traffic stream over SONET. According to the SONET specifications, an OC-48c (2.488 Gbps) has to be used in order to accommodate the GbE traffic at full speed. However, about 1.488 Gbps of the OC-48c will go unused. Alternatively, we can use an OC-12c (622 Mbps), but this will require appropriately reducing the speed of GbE. The best solution is to use an OC-21c (1.088 Gbps), since this is the SONET payload with a bandwidth that is close to the speed of GbE. However, this payload is not feasible since it has not been implemented in SONET equipment. It will take a major investment to develop this new payload and deploy it into the SONET equipment.

Virtual concatenation provides an efficient and economic solution to this problem. With virtual concatenation, seven independent OC-3c (155 Mbps) subrate payloads can be used to carry the GbE traffic. These seven payloads provide a total payload with 1.088 Gbps bandwidth. The incoming GbE stream is split into seven substreams, and each substream is mapped onto one of the seven OC-3c payloads. These payloads are then switched through the SONET network as individual payloads without the intermediate nodes being aware of their relationship. Virtual concatenation is only required to be implemented at the originating node where the incoming traffic is demultiplexed into the seven subrate payloads and at the terminating node, where the payloads are multiplexed back to the original stream. The seven payloads might not necessarily be contiguous within the same OC-N payload. Also, they do not have to be transmitted within the same SONET fiber. That is, if the SONET/SDH network consists of nodes that are interconnected with f fibers, then each of these seven payloads can be transmitted over any of the f fibers.

As we saw in Section 2.3.3, when transporting IP *packets over SONET/SDH (PoS)*, the entire SONET/SDH payload has to be dedicated to IP packets. Unlike PoS, virtual concatenation permits the bandwidth of a SONET/SDH frame to be divided into several subrate payloads, each of which can carry different type of traffic (see Figure 2.26). The

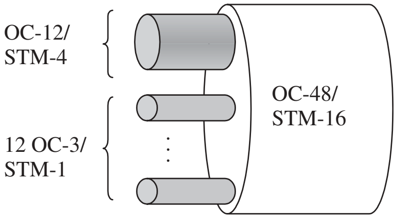


Figure 2.26 SONET/SDH virtual concatenation.

capacity of the OC-48/STM-16 payload is split into an OC-12/STM-4 subrate payload, which is used to carry voice; the rest is split into 12 OC-3/STM-1s, which can be used to carry data either individually or virtually concatenated.

2.8.2 Link Capacity Adjustment Scheme (LCAS)

The number of subrate payloads allocated to an application is typically determined in advance. However, the transmission rate of the application can indeed vary over time. In view of this, it can be useful to dynamically vary the number of subrate payloads allocated to an application. This can be done using the *link capacity adjustment scheme (LCAS)*. In LCAS, signaling messages are exchanged between the originating and terminating SONET/SDH node to determine and adjust the number of required subrate payloads. LCAS makes sure that the adjustment process is done without losing any data.

PROBLEMS

1. Consider the DS1 and E1 signals:
   a) How many voice calls are multiplexed in a DS1 signal?
   b) How many voice calls are multiplexed in an E1 signal?
   c) Why is there a difference in the number of voice calls carried in a DS1 signal and in an E1 signal?
2. Consider a time slot in the DS1 frame. Every sixth frame, the 8th bit is *robbed* and is used for signaling. What is the data rate of this signaling channel?
3. What is fractional T1?
4. In SONET, identify the differences between a section, a line, and a path.
5. In SONET, what is a virtual tributary group? What is a virtual tributary? Give an example of a virtual tributary.
6. Explain how an add/drop SONET/SDH multiplexer works.
7. Two IP routers are connected via an OC-3c link by using packet over SONET (PoS). Assume that 33% of the resulting HDLC frames are 50 bytes long, and the rest are 1500 bytes long. What is the rate of transmission of IP packets?
8. Consider the 1:1 and 1:*N* protection schemes in a point-to-point fiber link. Which of these two schemes provides better protection? Why?
9. Consider the two-fiber bidirectional line switched ring (2F-BLSR) shown in Figure 2.18. What is the total available capacity of the ring for the transmission of the working traffic from ADM 1 to ADM 2:

a) When all fibers are working?
b) When fibers 2 and 8 fail?
c) When fiber 12 fails after fibers 2 and 8 have failed?

10. Explain how the virtual concatenation scheme works. Why does it have to be implemented only at the originating and terminating nodes?

3

ATM Networks

The *Asynchronous transfer mode (ATM)* architecture was standardized by ITU-T in 1987 as the preferred architecture for the *broadband integrated services data network (B-ISDN)*. The broadband integrated services data network was conceived as a future high-speed network that would have replaced the telephone network and other data networks. It would have provided a single network for the transport of voice, video, and data. The term *asynchronous transfer mode* was chosen in contrast to *synchronous transfer mode (STM)*, which was proposed prior to the standardization of ATM and which was based on the SONET/SDH hierarchy. The term *transfer mode* means a telecommunication technique for transferring information.

Despite the enormous technological advancements in networking, the integration of voice, video, and data on to the same network is still elusive. Currently, ATM is a mature technology that is primarily used in the backbone. For instance, it is widely used in the backbone of *Internet service providers (ISPs)* and it has been deployed to provide point-to-point and point-to-multipoint video connections. It is also used in cellular telephony to carry multiple voice connections using the *ATM adaptation layer 2 (AAL 2)*. ATM is used for *circuit emulation*, which is a service that emulates a point-to-point T1/E1 circuit over an ATM network. ATM is also used in access networks such as ADSL-based residential access networks and *ATM passive optical networks (APON)*. ATM is not visible to the networking users, as is, for instance, the ubiquitous TCP/IP protocol. In view of this, it is often mistaken as a network that it is no longer in use – which is absolutely not the case!

ATM constituted a novel departure from previous networking architectures, and it has built-in mechanisms that permit it to transport different types of traffic with different QoS. Until the advent of *multi-protocol label switching (MPLS)* architecture in the late 1990s, ATM was the only networking technology that provided QoS on a per connection basis. The reader is encouraged to develop a good understanding of ATM and its congestion control schemes, before proceeding to Chapter 6 on MPLS.

This chapter is organized as follows. We first present the main features of the ATM architecture, such as the structure of the header of the *ATM cell*, the ATM protocol stack, and the physical layer. Then, we briefly describe the ATM shared memory switch architecture which is the dominant switch architecture, and various scheduling algorithms used to determine the order in which ATM cells are transmitted out. Subsequently, we describe the three *ATM adaptation layers (AAL): AAL 1*, *AAL 2*, and *AAL 5*. We conclude the chapter with a description of *classical IP and ARP over ATM*, a technique standardized by IETF designed for the transport of IP over ATM.

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

3.1 INTRODUCTION

The ATM architecture was designed with a view to transmitting voice, video, and data on the same network. These different types of traffic have different tolerance levels for packet loss and end-to-end delay, as shown in Table 3.1. For instance, packets containing voice have to be delivered on time so that the play-out process at the destination does not run out of data. On the other hand, the loss of some data might not necessarily deteriorate the quality of the voice delivered at the destination. At the other extreme, when transporting a data file, loss of data cannot be tolerated since this will compromise the file's integrity, but there is no stringent requirement that the file should be delivered as fast as possible.

The ATM architecture is based on the packet-switching principle and is connection-oriented. That is, in order for a sender to transmit data to a receiver, a connection has to be established first. The connection is established during the call setup phase, and when the transfer of data is completed, it is torn down.

There is neither error control nor flow control between two adjacent ATM nodes. Error control is not necessary, since the links in a network have a very low bit-error rate. In view of this, the payload of a packet is not protected against transmission errors. However, the header is protected in order to guard against forwarding a packet to the wrong destination! The recovery of a lost packet or a packet that is delivered to its destination with erroneous payload is left to the higher protocol layers. The lack of flow control requires congestion control schemes that permit the ATM network operator to carry as much traffic as possible without losing too many cells.

3.2 THE STRUCTURE OF THE HEADER OF THE ATM CELL

The ATM packet is known as *cell*, and it has a fixed size of 53 bytes. It consists of a payload of 48 bytes and a header of 5 bytes (see Figure 3.1). Several considerations – mostly related to the efficient transport of voice – led ITU-T to decide on such a small fixed-size packet.

Two different formats for the cell header were adopted, one for the *user network interface (UNI)* and a slightly different one for the *network-network interface (NNI)*. The

Table 3.1 Tolerance levels by traffic type.

Traffic Type	**Tolerance Levels**	
	Packet-loss sensitive	**Delay sensitive**
Voice	Low	High
Video	Moderate	High
Data	High	Low

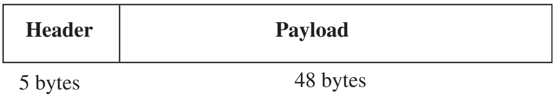


Figure 3.1 The ATM cell.

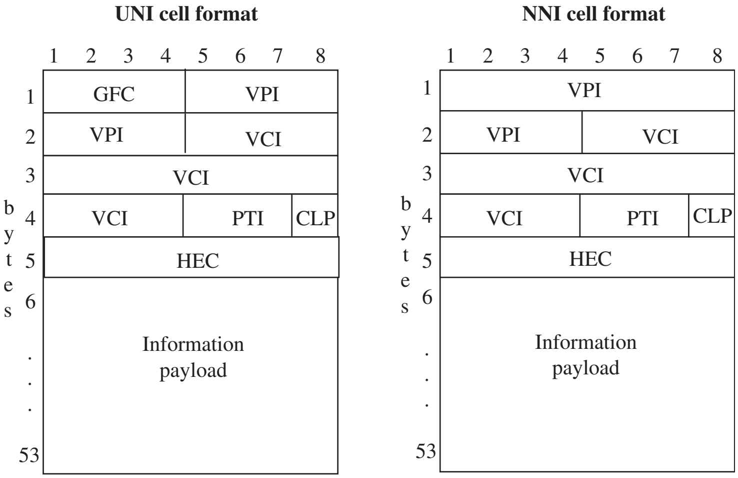


Figure 3.2 The structure of the cell header.

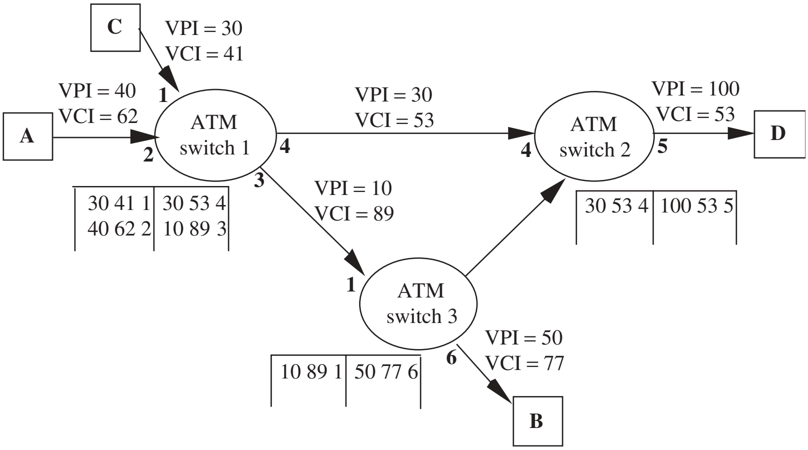
UNI is concerned with the interface between an ATM end device and the ATM switch to which it is attached. An ATM end device is any device that can be attached directly to an ATM network and that can transmit and receive ATM cells. The NNI is used between two ATM switches belonging to the same network or to two different networks.

The format of the cell header for these two interfaces is shown in Figure 3.2. As we can see, these two formats differ only in the first field. We now proceed to discuss in detail each field in the header. Understanding the meaning of these fields helps to better understand the ATM network architecture.

The *generic flow control (GFC)* field permits multiplexing of transmissions from several terminals on the same user interface. It is used to control the traffic flow from the end device to the network.

An ATM connection is identified by the combined *virtual path identifier (VPI)* and *virtual channel identifier (VCI)*. Such a connection is referred to as a *virtual channel connection (VCC)*. The VPI/VCI field is 24 bits in the UNI interface and 28 bits in the NNI interface. The VPI field is 8 bits in the UNI interface and 12 bits in the NNI interface. Therefore, in a UNI interface, there can be a maximum of 256 virtual paths, and in an NNI interface, there can be a maximum of 4096 virtual paths. In each interface, there can be a maximum of 65,536 VCIs. A VPI can be assigned to any value from 0 to 255. VCI values are assigned as follows: 0 to 15 are reserved by ITU-T, 16 to 31 are reserved by the ATM Forum, and 32 to 65,535 are used for user VCCs. The combined VPI and VCI allocated to a connection is known as the *connection identifier (CI)*. That is, CI = {VPI, VCI}.

A virtual channel connection between two users consists of a path through a number of different ATM switches. For each point-to-point link that lies on this path, the connection is identified by a different VPI/VCI. That is, each VPI/VCI has *local significance* and is translated to a different VPI/VCI at each switch that the cell traverses. This operation is referred to as *label swapping* since the connection identifier is also known as a *label*, a term adapted later on in MPLS. Label swapping involves a look-up in the switching



30 41 1	30 53 4
40 62 2	10 89 3

30 53 4	100 53 5

10 89 1	50 77 6

Figure 3.3 An example of label swapping.

table of the ATM switch. The VPI/VCI of the incoming cell is indexed into the switching table and the new VPI/VCI that the cell will carry in its header on the outgoing link is obtained. At the same time, the output port number is also obtained, so that the switch knows to which output port to forward the cell. As mentioned above, labels have only local significance; that is, they are only used on a single link. This simplifies the selection process of a new label for particular link.

An example of label swapping is given in Figure 3.3. Each switch is represented by a circle, and the switching table is given immediately below the circle. For presentation purposes, we assume that the switching table is centralized and it contains information for all input ports. (In practice, there is a switching table for each input port.) The first column in the switching table specifies the VPI/VCI of each incoming connection and its input port. The second column gives the new label and the destination output port of each connection. Let us follow the path from A to B, which traverses through ATM switches 1 and 3. Note that on the incoming link to ATM switch 1, the connection has the label VPI = 40, VCI = 62. From the switching table, we find that its new label is VPI = 10, VCI = 89 and it should be switched to output port 3 of ATM switch 1. At ATM switch 3, we see that the connection's new label on the outgoing link is VPI = 50, VCI = 77, and its destination output port is 6. Therefore, the path from A to B consists of the following three different labels: VPI/VCI = 40/62, VPI/VCI = 10/89, and VPI/VCI = 50/77. The input and output ports of each switch, through which the connection is established, can also be identified: the connection enters ATM switch 1 at input port 2, then exits from the same switch from output port 3, then enters ATM switch 3 at input port 1, and finally exits from output port 6. Similarly, the path from C to D can be traced.

ATM connections are point-to-point and point-to-multipoint. Point-to-point connections are bidirectional, and point-to-multipoint connections are unidirectional. An ATM connection, depending upon how it is set up, can be either a *permanent virtual connection (PVC)* or a *switched virtual connection (SVC)*. A PVC is established manually by a network administrator using network management procedures. Typically, it remains in place for a long time. An SVC is established in real-time by the network using signaling procedures, and it remains up for an arbitrary amount of time. The signaling protocol Q.2931 that is used to establish and release a point-to-point SVC is described in Chapter 5.

A point-to-point SVC is established when end device A sends a SETUP message to the switch to which it is attached – known as the *ingress* switch – requesting that a connection be established to a destination end device B. The ingress switch calculates a path through the network to B, and then it forwards the setup request to the next hop switch on the path, which forwards the request to its next hop switch. This forwarding process continues until the setup request reaches the switch to which B is attached, known as the *egress* switch. This last switch sends the setup request to B. If the request is accepted, then a confirmation message is sent back to A. At that time, A can begin to transmit data. Each switch on the path performs the following tasks: allocates some bandwidth to the new connection, selects a VPI/VCI label, and updates its switching table. When the connection is terminated, the switch tasks are done in reverse. That is, each switch: removes the entry in the switching table associated with the connection, returns the VPI/VCI label to the pool of free labels, and releases the bandwidth that was allocated to the connection.

Each switch on the path of a new connection has to decide – independently of the other switches – whether it has enough bandwidth to provide the QoS requested for this connection. This is done using a *call admission control (CAC)* algorithm (discussed in Chapter 4).

In addition to permanent and switched virtual connections, there is another type of connection known as a *soft PVC*. Part of this connection is permanent and part of it is switched. The connection is set up using both network management procedures and signaling procedures.

Let us now examine the remaining fields in the ATM header. The *payload type indicator (PTI)* field has three bits. It is used to indicate different types of payloads, such as user data and OAM. It also contains a bit used for *explicit congestion control notification (EFCN)* and a bit used in conjunction with the *ATM adaptation layer 5*. The explicit congestion control notification mechanism is used in the ABR scheme described in Section 4.8.1. Also, the ATM Adaptation Layer 5 is described below in Section 3.7.3. Table 3.2 summarizes the PTI values.

Bit 3, which is the left-most and most significant bit, is used to indicate if the cell is a user data cell (in which case, bit 3 is set to 0), or an OAM data cell (in which case, bit 3 is set to 1). For a user data cell, bit 2 carries the explicit forward congestion indicator.

Table 3.2 Payload type indicator (PTI) values.

PTI	Meaning
000	User data cell, congestion not experienced, SDU type = 0
001	User data cell, congestion not experienced, SDU type = 1
010	User data cell, congestion experienced, SDU type = 0
011	User data cell, congestion experienced, SDU type = 1
100	Segment OAM flow-related cell
101	End-to-end OAM flow-related cell
110	RM cell
111	Reserved

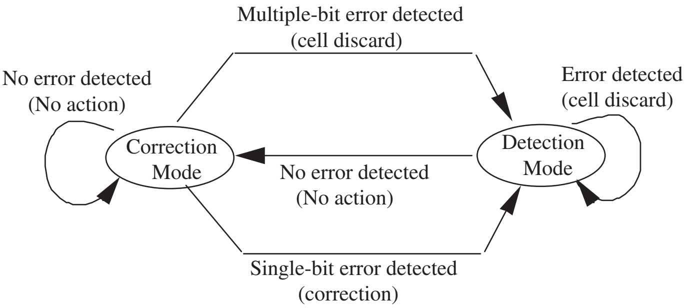


Figure 3.4 The header error control state machine.

Bit 2 is set to 0 if no congestion has been experienced, and to 1 if congestion has been experienced. Also, for a user data cell, bit 1 is used by the ATM Adaptation Layer 5. It is set to 0 if the *service data unit (SDU)* type is zero, and to 1 if the SDU type is one.

For OAM data cells, two types are defined. In addition, a *resource management (RM)* cell is defined, and is used in conjunction with the *available bit rate (ABR)* mechanism, which is a feedback-based congestion control mechanism (see Chapter 7).

The *cell loss priority (CLP)* bit is used to indicate whether a cell can be discarded when congestion arises inside the network. If a cell's CLP bit is set to one, then the cell can be discarded. On the other hand, if the cell's CLP bit is set to 0, then the cell cannot not be discarded. The use of the CLP bit is discussed in Chapter 4.

The *header error control (HEC)* field is used to correct single-bit and to detect multiple-bit transmission errors in the header. CRC is used with a 9-bit pattern given by the polynomial $x^8 + x^2 + x + 1$. The HEC field contains the 8-bit FCS obtained by using the formula of: (the first 32 bits of the header) x 2^8 divided by the above pattern. The state machine that controls the head error correction scheme (see Figure 3.4) is implemented in the physical layer (see Section 3.4).

At initialization, the receiver's state machine is set to the *correction mode*. Each time that a cell arrives, the CRC is carried out. If no errors are found, then the cell is allowed to proceed to the ATM layer and the state machine remains in the correction mode. If a single-bit error is detected, then the error is corrected and the cell is allowed to proceed to the ATM layer, but the state machine switches to the *detection mode*. If a multi-bit error is detected, then the cell is discarded and the state machine switches to the detection mode. In detection mode, then the CRC is carried out each time that a cell comes in. If a single-bit or a multi-bit error is detected, then the cell is discarded and the state machine remains in the detection mode. If no errors are detected, then the cell is allowed to proceed to the ATM layer and the state machine shifts back to the correction mode.

3.3 THE ATM PROTOCOL STACK

The ATM protocol stack is shown in Figure 3.5. It consists of the physical layer, the ATM layer, the ATM Adaptation Layer, and higher layers that permit various applications to run on top of ATM. It is important to note that the ATM layer and the ATM adaptation layer do not correspond to any specific layers of the OSI reference model; it is erroneous to refer to the ATM layer as the data link layer.

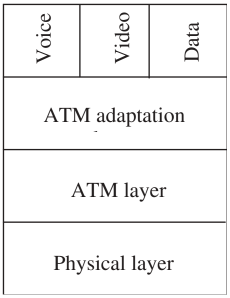


Figure 3.5 The ATM protocol stack.

The ATM stack shown in Figure 3.5 is for the transport of data. A similar stack for the ATM signaling protocols is discussed in Chapter 5. The physical layer, the ATM layer, and the ATM adaptation layer are described below in Sections 3.4, 3.5, and 3.7 respectively.

3.4 THE PHYSICAL LAYER

The physical layer transports ATM cells between two adjacent ATM layers. The ATM layer is independent of the physical layer, and it operates over a wide variety of physical link types. The physical layer is subdivided into the *transmission convergence (TC)* sublayer and the *physical medium dependent (PMD)* sublayer.

The PMD sublayer on the transmitter's side is concerned with the transmission and transport across a link of a stream of bits that it receives from the TC sublayer. At the receiver's side, it recovers the stream of bits and passes it on to the TC sublayer.

The TC sublayer interacts between the ATM layer and the PMD sublayer. On the transmitter's side, it receives ATM cells from the ATM layer and creates a bit stream that passes on to the PMD sublayer. On the receiver's side, it reconstructs the ATM cells from the bit stream that it receives from the PMD sublayer and passes them on to the ATM layer.

3.4.1 The Transmission Convergence (TC) Sublayer

The following are the main functions performed by this sublayer:

HEC cell generation and verification

The ATM layer passes to the physical layer ATM cells for transmission over the link. Each ATM cell is complete, except for the HEC byte. This byte is computed and inserted into the HEC field in the TC sublayer. At the receiving side of the link, the HEC state machine is implemented in the TC sublayer. TC will drop any cell whose header was found to be in error.

Decoupling of cell rate

The PMD sublayer expects to receive a continuous stream of bits. During the time that ATM cells are not passed down from the ATM layer, TC inserts idle cells in-between the cells received from the ATM layer so that to maintain the continuous bit stream expected from PMD. These idle cells are discarded at the receiver's TC sublayer. They are identified uniquely; their header is marked as: VPI = 0, VCI = 0, PTI = 0, and CLP = 0.

Cell delineation

Cell delineation is used in the extraction of cells from the bit stream received from the PMD sublayer. The following procedure for cell delineation is based on the HEC field. Let us consider a bit stream, and let us assume that we have guessed correctly the first bit of a new cell. This means that this bit and the following 39 bits make up the header of the cell. Consequently, if we carry out the CRC operation on these 40 bits, the resulting FCS should be 0. If it is not 0, then that means that the bit we identified as the beginning of the cell is not actually the first bit. So we repeat this process starting with the next bit, and so on, until we get a match. When we do get a match, then we know that we have correctly identified in the bit stream the beginning of a cell.

This simple idea is used in the state machine for cell delineation (see Figure 3.6). The state machine consists of the *hunt state*, the *presync state*, and the *sync state*. The state machine is in the hunt state when the link is initialized, or after a receiver failure is detected. In this state, the incoming bit stream is continuously monitored in order to detect the beginning of a cell using the above matching procedure. When a match occurs, then the state machine moves to the presync state. In this state, it checks that the FCS of δ consecutive cells is 0. If a mismatch is found (i.e., the FCS of one of these cells is not set at 0), then the state machine goes back to the hunt state. Otherwise, synchronization with the bit stream is achieved, and the state machine moves to the sync state. Note that synchronization is assumed to be lost if α consecutive mismatches occur. In this case, the state machine shifts to the hunt state. ITU-T recommends that $\delta = 6$ and $\alpha = 7$.

While in the sync state, the HEC state machine (described in Section 3.2) is used to detect errors in the header of the incoming cells. Recall that when the state machine is in the correction mode, a cell is dropped if more than one erroneous bits are detected in its header. If only one erroneous bit is detected, then it is corrected and the cell is delivered to the ATM layer. When the state machine is in the detection mode, a cell is dropped if one or more erroneous bits are detected in its header. If no errors are detected in a cell header – when the state machine is in either state – then the cell is delivered to the ATM layer.

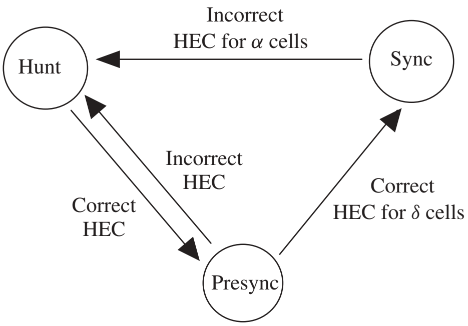


Figure 3.6 The cell delineation state machine.

Transmission frame generation and recovery

In frame-oriented transmission systems (such as SONET), TC generates frames at the sender's side by placing frame-related information and ATM cells into a well-defined frame structure. At the receiver's side, it recovers the frames and subsequently the ATM cells from the bit stream.

3.4.2 The Physical Medium-Dependent (PMD) Sublayer

The following are the main functions performed by this sublayer:

Timing function

This is used to synchronize the transmitting and receiving PMD sublayers. It generates timing for transmitted signals; it also derives correct timing for received signals.

Encoding/decoding

PMD can operate either on a bit-by-bit basis or with a group of bits (as in the 4B/5B and 8B/10B schemes). In the 4B/5B encoding scheme, each group of four bits is coded by a 5-bit code. In the 8B/10B scheme, each group of eight bits is coded by a 10-bit code. Coding groups of bits is known as *block coding*. Block coding requires more bandwidth than it effectively provides. For instance, FDDI uses 4B/5B block coding and runs at 125 Mbps, which gives an effective bandwidth of 100 Mbps. There are several benefits with block coding, such as bit boundary detection and exchange of control information. In addition, enhanced speed in the execution of the protocol is achieved, since it operates in chunks of bits.

3.5 THE ATM LAYER

The ATM layer is concerned with the end-to-end transfer of information. That is, it oversees functionality from the transmitting end device to the receiving end device. Below, we summarize its main features.

Connection-oriented packet switching

The ATM layer is a connection-oriented packet-switching network. Unlike the IP network, an ATM end device cannot transmit cells to a destination ATM end device over an ATM network unless a virtual channel connection is established first. Cells are delivered to the destination in the order in which they were transmitted.

A connection is identified by a series of VPI/VCI labels (as explained in Section 3.2). A connection can be either point-to-point or point-to-multipoint. *Point-to-point connections* are bidirectional, whereas *point-to-multipoint connections* are unidirectional. Connections can be either *permanent virtual circuits (PVC)* or *switched virtual circuits (SVC)*. PVCs are set up using network management procedures, whereas SVCs are set up on demand using ATM signaling protocol procedures.

Fixed size cells

In the ATM layer, packets are fixed-size cells of 53 bytes long, with a 48-byte payload and 5-byte header. The structure of the header was described in detail in Section 3.2.

Cell switching

Switching of cells in an ATM network is done at the ATM layer. For an example of the ATM stacks that are used when two end devices communicate with each other, see Figure 3.7. Both end devices run the complete ATM stack: the physical layer, the ATM layer, the AAL, and the application layer. The ATM switches only need the physical layer and the ATM layer in order to switch cells.

No error and flow control

In the OSI model, the data link layer provides error and flow control on each hop using the ARQ mechanism. In ATM networks, there is neither error control nor flow control between two adjacent ATM switches that are connected with a point-to-point link. If a cell arrives at an ATM switch when the switch is experiencing congestion, then the cell is simply lost, or perhaps is delivered to a destination end device with an erroneous payload.

Because of the high reliability of fiber-based transmission links, the probability that a cell is delivered to the destination end device with an erroneous payload is extremely small. Typically, the probability that a bit will be received wrongly is less than 10^{-8}. So, if we assume that bit errors occur independently of each other, then the probability that the payload of an ATM cell (which consists of 48 bytes or 384 bits) will not contain errors is $(1 - 10^{-8})^{384}$. Therefore, the probability that it will contain one or more erroneous bits is 1 - $(1 - 10^{-8})^{384}$, which is very low.

The cell loss rate is a QoS parameter that can be negotiated between the end device and the ATM network at setup time. Different applications tolerate different cell loss rates. For instance, video and voice are less sensitive to cell loss than is a file transfer. Cell loss rates typically vary from 10^{-3} to 10^{-6}. The ATM network guarantees the negotiated cell loss rate for each connection.

In the ATM standards, there is a mechanism for recovering lost cells or cells delivered with erroneous payload. However, it is only used to support the ATM signaling protocols (see SSCOP in Chapter 5). The recovery of the data carried by lost or corrupted cells is expected to be carried out by a higher-level protocol, such as TCP. Depending upon the application that created the data, there might not be enough time to recover lost cells. Also, it might be deemed unnecessary to recover lost cells, such as when transmitting video over ATM.

When TCP/IP runs over ATM, the loss or corruption of the payload of a single cell results in the retransmission of an entire TCP PDU. In order to clarify this point, let us

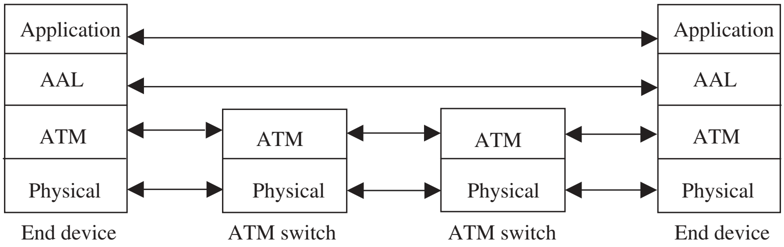


Figure 3.7 Cell switching in an ATM network.

assume that we want to send a single TCP PDU over an ATM network. This PDU will be encapsulated by IP and it will be passed on to the ATM network. (For simplicity, we assume no fragmentation of the IP PDU.) As will be seen in Section 3.7, the ATM adaptation layer will break the IP PDU into small segments, and each segment will be placed in the payload of an ATM cell. Let us assume that the IP PDU will be carried in n ATM cells. When these n cells arrive at the destination, their payloads will be extracted and the original IP PDU will be reconstructed, from which the TCP PDU will be extracted.

Assume that one of these n cells is either lost or its payload is corrupted. If this causes the IP header to get corrupted, then IP will drop the PDU. TCP will eventually detect that the PDU is missing and it will request its retransmission. On the other hand, if the cell in question causes the TCP PDU to get corrupted, then TCP will again detect it and it will request its retransmission. In either case, the loss of a cell or the corruption of the payload of a cell will cause the entire PDU to be retransmitted. Since this is not expected to happen very often, it should not affect the performance of the network.

Addressing

Each ATM end device and ATM switch has a unique ATM address. Private and public networks use different ATM addresses; public networks use E.164 addresses and private networks use the OSI NSAP format. Details on ATM addresses are given in Section 5.5.

ATM addresses are different from IP addresses. Therefore, when running IP over ATM, IP addresses must be translated into ATM addresses, and vice versa (see Section 3.8 below).

Quality of service (QoS)

Each ATM connection is associated with a QoS category. Six different categories are provided by the ATM layer: *constant bit rate (CBR), real-time variable bit rate (RT-VBR), non-real-time variable bit rate (NRT-VBR), available bit rate (ABR), unspecified bit rate (UBR)*, and *guaranteed frame rate (GFR)*. The CBR category is intended for real-time applications that transmit at a constant rate, such as circuit emulation. The RT-VBR category is intended for real-time applications that transmit at a variable rate, such as encoded video and voice. The NRT-VBR category is for delay-sensitive applications that transmit at a variable rate but that do not have real-time constraints. For example, when frame relay is carried over an ATM network, it can use this category. The UBR category is intended for delay-tolerant applications, such as those running on top of TCP/IP. The ABR category is intended for applications that can vary their transmission rate according to how much slack capacity there is in the network. Finally, the GFR category is intended to support non-real-time applications that might require a minimum guaranteed rate.

Each QoS category is associated with a set of traffic parameters and a set of QoS parameters. The traffic parameters are used to characterize the traffic transmitted over a connection, and the QoS parameters are used to specify the cell loss rate and the end-to-end delay required by a connection. The ATM network guarantees the negotiated QoS for each connection. QoS for ATM networks is discussed more in Chapter 4.

Congestion control

In ATM networks, congestion control permits the network operator to carry as much traffic as possible without affecting the QoS requested by the users. Congestion control can be either preventive or reactive. In *preventive congestion control*, network congestion can be prevented by using a *call admission control (CAC)* algorithm. CAC decides whether or not to accept a new connection; if accepted, then CAC polices the amount of data that is transmitted on that connection. In *reactive congestion control*, network congestion is managed by regulating how much the end devices transmit through feedback messages. These two schemes are described in detail in Chapter 4.

Higher-level layers

Various applications (such as voice, circuit emulation, and video) can run on top of AAL. Connection-oriented protocols (such as frame relay) and connectionless protocols (such as IP, signaling protocols, and network management protocols) also run on top of AAL.

3.6 THE ATM SWITCH ARCHITECTURE

The main function of an ATM switch is to transfer cells from its incoming links to its outgoing links. This is known as the *switching* function. A switch also performs several other functions, such as signaling and network management. A generic model of an ATM switch consisting of N input ports and N output ports is shown in Figure 3.8. Each input port can have a finite capacity buffer, where cells wait until they are transferred to their destination output ports. The input ports are connected to the output ports via the *switch fabric*. Each output port can also be associated with a finite capacity buffer, where cells can wait until they are transmitted out. Depending upon the structure of the switch fabric, there might be additional buffering inside the fabric. An ATM switch whose input ports are equipped with buffers is referred to as an *input buffering* switch (this is irrespective of whether or not its output ports have buffers). If an ATM switch only has buffers for its output ports, then it is referred to as an *output buffering* switch. Depending on the switch architecture, cell loss might occur at the input ports, within the switch fabric, or at the output ports.

An ATM switch is equipped with a CPU, which is used to carry out signaling and management functions.

Label swapping takes place before a cell is transferred to its destination output port. The value of the VPI/VCI fields of an arriving cell is looked up in a table, which provides the new VPI/VCI values and the destination output port number. This table is implemented

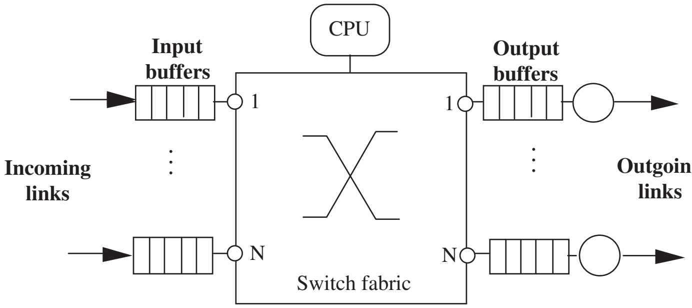


Figure 3.8 A generic model of an ATM switch.

on each input interface and is not very large, which makes the table search time to be minimal.

ATM switch architectures can be grouped into the following three classes: *space-division, memory sharing*, and *medium sharing*. Space-division switch architectures are based on *multistage interconnection networks (MIN)*. A MIN consists of a network of interconnected switching elements arranged in rows and columns. A shared memory switch architecture uses a single memory for storing all of the incoming cells from the input ports. Cells stored in the memory are organized into linked lists, one per output port. The cells in each linked list are transmitted out of the switch by its associated output port. Finally, in a medium sharing switch, all arriving cells at the switch are synchronously transmitted onto a bus. Each output port i sees all of the cells transmitted on the bus, and receives those cells whose destination is output port i. In front of each output port, there is a buffer, where the cells can wait until they are transmitted out. The shared memory switch is the dominant switch architecture and is described below.

3.6.1 The Shared Memory Switch

The main feature of this switch architecture is a shared memory that is used to store all of the cells coming in from the input ports. The cells in the shared memory are organized into linked lists – one per output port (see Figure 3.9). The shared memory is *dual ported;* that is, it can read and write at the same time. At the beginning of each slot, each input port that holds a cell, writes it into the shared memory. At the same time, each output port reads the cell from the top of its linked list (assuming that the linked list has a cell) and transmits it out. If N is the number of input/output ports, then one slot can write up to N cells into the shared memory and transmit up to N cells out of the shared memory. If the speed of transmission on each incoming and outgoing link is V, then the switch can keep up at maximum arrival rate, if the memory's bandwidth is at least $2NV$.

The total number of cells that can be stored in the memory is bounded by the memory's capacity B, expressed in cells. Modern shared memory switches have a large shared memory and can hold hundreds of thousands of cells. The total number of cells allowed to queue for each output port i is limited to B_i, where $B_i < B$. That is, the linked list associated with output port i cannot exceed B_i. This constraint is necessary for avoiding starvation of other output ports when output port i gets *hot*; that is, when a lot of the incoming traffic goes to that particular port. When this happens, the linked list associated

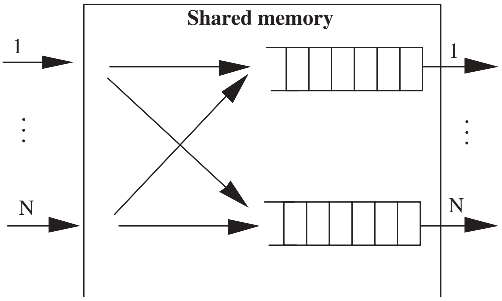


Figure 3.9 A shared memory switch.

with the hot output port might grow to the point that it takes over most of the shared memory. In this case, there will be little space left for cells destined to other output ports. Typically, the sum of the B_i capacities of all linked lists is greater than B. More complicated constraints can also be used. For instance, each linked list i, in addition to its maximum capacity B_i, can be associated with a minimum capacity LB_i, where $LB_i < B_i$. LB_i is a dedicated buffer for output port i and is never shared with the other output ports. The sum of the LB_i capacities of all linked lists is less than B.

Cell loss occurs when a cell arrives at a time when the shared memory is full; that is, it contains B cells. Cell loss also occurs when a cell with destination output port i arrives at a time when the total number of cells queued for this output port is B_i cells. In this case, the cell is lost – even if the total number of cells in the shared memory is less than B.

A large switch can be constructed by interconnecting several shared memory switches. That is, the shared memory switch described above is used as a switching element, and all of the switching elements are organized into a multistage interconnection network.

The shared memory switch architecture is an example of a *non-blocking output buffering* switch. In a *non-blocking* switch, the switching fabric does not cause blocking. That is, it is not possible for cells traversing the switch fabric to collide with each other cells and thus block each other from reaching their destination output ports. An *output buffering* switch is a switch that has buffers only in its output ports. Recall that any switch that has buffers in its input ports, irrespective of whether it has buffers at the output ports, is called an *input buffering* switch. It has been shown that input buffering switches have a lower throughput than output buffering switches, due to *head-of-line blocking*. Obviously, non-blocking output buffering switches are the preferable switch architectures.

3.6.2 Scheduling Algorithms

Let us consider a non-blocking switch with output buffering (see Figure 3.10). Each output buffer holds cells that belong to different connections passing through it. Each of these connections is associated with a QoS category signaled to the switch at call setup time. The cells belonging to these connections can be grouped into queues (one queue per QoS category), and these queues can be served using a scheduling algorithm.

As will be seen in the next chapter, several QoS categories have been defined in ATM. Let us consider the following four categories: *constant bit rate (CBR), real-time variable bit rate (RT-VBR), non-real-time variable bit rate (NRT-VBR)*, and *unspecified bit rate (UBR)*. The CBR service category is intended for real-time applications that transmit at

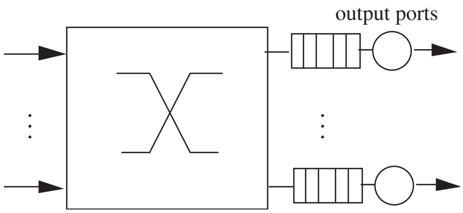


Figure 3.10 A non-blocking switch with output buffering.

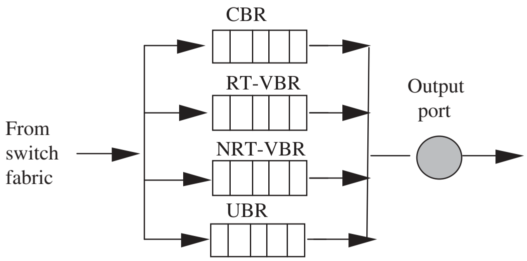


Figure 3.11 Logical queues for an output port.

a constant bit rate, such as unencoded video and circuit emulation. The RT-VBR service category is intended for real-time applications that transmit at a variable bit rate, such as encoded voice and video. The NRT-VBR service category is for applications that transmit at variable bit rate and that do not have real-time requirements. The UBR service category is intended for delay-tolerant applications that do not require any guarantees, such as data transfers.

Each output buffer can be organized into four different queues based on these four QoS categories (see Figure 3.11). When a cell arrives at an output buffer, it goes to the queue that corresponds to its connection's QoS category. The cells in these queues are transmitted out one at a time according to a scheduling algorithm. The scheduling algorithm gives different priorities to these queues so that the QoS for each queue is satisfied. For instance, it will give a higher priority to the CBR queue than to the UBR queue because the cells in the CBR queue carry delay-sensitive data. Below, we describe some of the scheduling algorithms that have been used.

Static priorities

The queues are assigned static priorities, which dictate the order in which they are served. These priorities are called *static* because they remain unchanged over time. For instance, the CBR queue has the highest priority, the RT-VBR the second highest priority, and so on, with the UBR queue having the lowest priority. The priorities are unaffected by the occupancy levels of the queues. These queues can be served as follows. Upon completion of the transmission of a cell, the next cell for transmission is selected from the CBR queue. If the CBR queue is empty, the next cell for transmission is selected from the RT-VBR queue. If this queue is empty, then the next cell is selected from the NRT-VBR queue, and so on. If all of the queues are empty, then no cell will be selected for transmission. Thus, the CBR queue is served until it becomes empty. Then, the next priority queue is served until it becomes empty, and so on. For example, if during the time that a cell from the UBR queue is being transmitted out, a cell arrives in one of the higher-priority queues (i.e. the CBR, RT-VBR, or NRT-VBR queue), then the higher- priority cell will be transmitted out next after the transmission of the cell from the UBR queue is completed. Additional scheduling rules can be introduced to account for the current status of the queues. A typical example is the *aging factor rule*. If a queue has not been served for a duration that exceeds its pre-specified threshold, then the queue's priority is momentarily raised so that some of its cells can be transmitted out. The aging factor rule typically occurs with a low priority queue.

Early deadline first (EDF) algorithm

In this algorithm, each cell is assigned a deadline upon arriving at the buffer. This deadline indicates the time by which the cell should depart from the buffer. The deadline is calculated by adding a fixed delay to the arrival time of the cell. This delay can vary according to the QoS category of the cell. The scheduler serves the cells according to their deadlines, so that the one with the earliest deadline gets served first. A cell that is assigned a deadline closer to its arrival time will experience a relatively lower delay in the buffer. Conversely, a cell that is assigned a deadline further out from its arrival time is likely to experience a longer delay before it gets transmitted out. Using this scheme, cells belonging to delay-sensitive applications (such as voice or video) can be served first by assigning them deadlines closer to their arrival times.

The weighted round-robin scheduler

Each output buffer is organized into any number of queues. For instance, there could be one queue for every connection that passes through the particular output port. Or there could be fewer queues, such as one queue per QoS category. The scheduler serves one cell from each queue in a round-robin fashion. The queues are numbered from 1 to M, and are served sequentially. That is, if a cell from queue 1 was just served, then the next queue to be served is queue 2. This sequential servicing of the queues continues until the Mth queue is served, whereupon the scheduler goes back to queue 1. If the next queue to be served (say queue i) is empty, then the scheduler skips it and goes on to queue $i + 1$.

Weighted round-robin scheduling can be used to serve a different number of cells from each queue. For instance, let us assume that there are five connections, each with the following weights: 0.1, 0.2, 0.4, 0.7, and 1, respectively, for each queue. Each weight is converted into an integer number by multiplying them all by a common number. For example, 0.1, 0.2, 0.4, 0.7, and 1 are each multiplied by 10 to get 1, 2, 4, 7, and 10, respectively. These integer numbers indicate how many cells should be served from each queue. Thus, the scheduler serves one cell from the first queue, two from the second queue, four from the third queue, seven from the fourth queue, and ten from the fifth queue.

Consider the case where one of the queues, say queue 5, becomes idle for some time. If that happens, then queues 1 to 4 will be served normally, and queue 5 will be skipped each time its turn comes up. The ten slots that would have been used for queue 5 are now allocated to queues 1 to 4, proportionally to their weights.

3.7 THE ATM ADAPTATION LAYER

The *ATM adaptation layer (AAL)* is sandwiched between the ATM layer and the higher-level layers (see Figure 3.12). AAL converts the traffic generated by a higher-level layer to ATM payloads and provides different types of services to the higher-level layer.

AAL consists of two sublayers: the *convergence sublayer (CS)* and the *segmentation and reassembly sublayer (SAR)*. (See Figure 3.13.) The convergence sublayer provides service-specific functions. It is further subdivided into the *service-specific convergence sublayer (SSCS)* and the *common part sublayer (CPS)*. SAR, on the other hand, has two

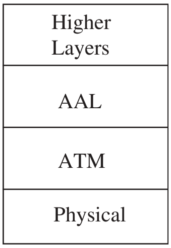


Figure 3.12 The ATM adaptation layer.

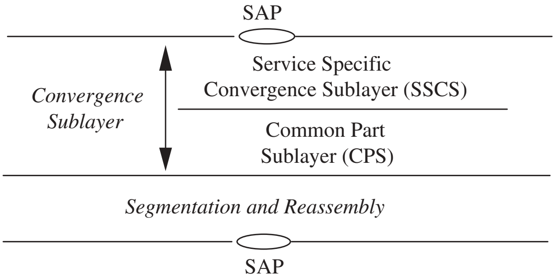

SAP
Convergence Sublayer
Service Specific Convergence Sublayer (SSCS)
Common Part Sublayer (CPS)
Segmentation and Reassembly
SAP

Figure 3.13 The ATM adaptation sublayers.

related functions, depending on where the ATM cell is currently located: at the transmitting side or at the receiving side. At the transmitting side, SAR segments higher-level layer PDUs into a suitable size for the information field of an ATM cell. At the receiving side, it reassembles the information fields of ATM cells into higher-level layer PDUs.

Four ATM Adaptation Layers have been standardized: *ATM adaptation layer 1 (AAL 1), ATM adaptation layer 2 (AAL 2), ATM adaptation layer 3/4 (AAL 3/4)*, and *ATM adaptation layer 5 (AAL 5)*. Of these, all are currently used except for AAL 3/4. An additional ATM adaptation layer, the *signaling ATM adaptation layer (SAAL)*, was defined to support the ATM signaling protocols. SAAL is described in Chapter 5.

3.7.1 ATM Adaptation Layer 1 (AAL 1)

AAL 1 can be used for applications such as circuit emulation services, constant-bit rate video, and high-quality constant-bit rate audio. It provides transfer of constant-bit rate data, delivery at the same bit rate, and transfer of timing information between the sending and receiving applications. Also, it can handle cell delay variation and detect lost or misrouted cells.

AAL 1 consists of a SAR sublayer and a CS. The SAR sublayer is responsible for the transport and bit error detection, and possibly correction, of blocks of data received from CS. The CS performs a variety of functions. These functions include handling cell delay variation, processing the sequence count, transferring structured and unstructured data, and transferring timing information.

The AAL 1 SAR sublayer

The SAR sublayer accepts blocks of 47 bytes from the CS and adds a 1-byte header to form the SAR-PDU. The SAR-PDU is then passed on to the ATM layer, where it gets encapsulated with a 5-byte ATM header. The ATM cell is then passed on to the physical layer, which transmits it out. At the receiving SAR sublayer, the 1-byte header is stripped and the payload of the SAR-PDU is delivered to the receiving CS.

The encapsulation of the SAR-PDU is shown in Figure 3.14. The header consists of two fields: the *sequence number (SN)* field and the *sequence number protection (SNP)* field. Both fields are 4 bits long. The SN field contains the subfields:

- *Convergence sublayer indication (CSI)*: It carries an indication that is provided by the CS. The default value of the CSI bit is 0.
- *Sequence count*: Provided by the transmitting CS, this field is associated with the block of data in the SAR-PDU. The count starts at 0 and is increased sequentially modulo 8. The receiving CS uses the sequence count to detect lost or misinserted cells.

The SNP field contains the following two subfields:

- *CRC-3*: It is computed over the CSI and sequence count fields.
- *Parity*: Even parity bit used calculated over the CSI, sequence count, and CRC-3 fields.

The transmitting SAR computes the FCS for the first four bits of the header and inserts it into the CRC-3 field. The pattern used to compute the FCS is given by the polynomial: $x^3 + x + 1$. After completing the CRC operation, the transmitting AAL calculates the even parity bit on the first seven bits of the header and inserts the result in the parity field.

The receiving SAR examines each SAR-PDU header by checking the FCS and the even parity bit. The state machine that controls the receiver's error detection and correction scheme is the same as the header error control scheme used for the ATM header (see Section 3.2; see also Figure 3.4). At initialization, the state machine is set to the correction mode. Each time an SAR-PDU comes in, the FCS and the parity bit are checked. If no errors are found, the SN field is declared as valid and the state machine remains in the correction mode. If a single-bit error is detected, then it is corrected and the SN field is declared as valid, but the state machine switches to detection mode. If a multi-bit error is detected, then SN field is declared as invalid and the state machine switches to detection mode. In detection mode, the FCS and the parity bit are checked each time an SAR-PDU comes in; if a single-bit or a multi-bit error is detected, then the SN field is declared as

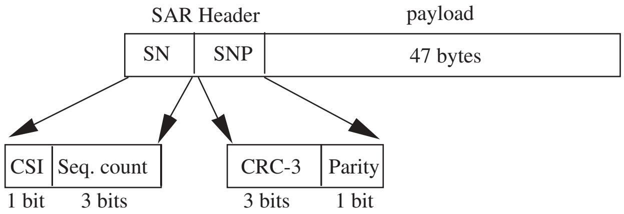


Figure 3.14 The SAR encapsulation for AAL 1.

invalid and the state machine remains in detection mode. If no errors are detected, then the SN field is declared as valid and the state machine returns to the correction mode.

The receiving SAR sublayer conveys the sequence count, the CS indication, and the status of the SN field (i.e., valid or invalid) to the receiving CS.

This error detection and correction scheme runs in addition to the error detection and correction scheme for the ATM header. However, these two mechanisms apply to two different fields of the ATM cell. The header error control mechanism applies to the first four bytes of cell's header, whereas the above scheme applies to the SN field.

The AAL 1 CS

The convergence sublayer performs various functions, such as handling cell delay variation, processing the sequence count, processing forward error correction, handling performance monitoring, transferring structured and unstructured data, and transferring timing information. Below, we describe each of these functions.

Handling cell delay variation

For AAL 1 to support constant bit rate applications, it has to deliver the data stream to the receiving application at the same bit rate at which it was transmitted. The ATM cells, which carry the data stream, might have to traverse multiple ATM switches before they reach their destination. In view of this, the arrival of these cells might sometimes be delayed because of network congestion. Also, the opposite might occur. That is, a group of cells might arrive closer to each other than they were originally transmitted.

To compensate for this variability in the arrival of the cells, CS writes the incoming SAR-PDUs into a buffer, from where it delivers the data stream to the receiving AAL application at a constant bit rate. (A similar method is used, for instance, when we listen to a radio station over the Internet.) In the event of buffer underflow, CS might need to insert dummy bits in order to maintain bit count integrity. Also, if there is a buffer overflow, then CS drops the appropriate number of bits.

Processing the sequence count

CS processes the sequence count values in order to detect lost or misinserted cells. Any detected misinserted cells are discarded. To maintain bit count integrity of the AAL user information, it might be necessary to compensate for lost cells by inserting dummy SAR-PDU payloads.

Forward error correction

For video, forward error correction might be performed in order to protect against bit errors. For enhanced error-protection, forward error correction can be combined with interleaving of AAL user bits.

Performance monitoring

CS can generate reports giving the status of end-to-end performance, as deduced by the AAL. The performance measures can be based on events of: lost or misinserted cells, buffer underflows and overflows, or bit errors.

Structured and unstructured data transfers

Two CS-PDU formats have been defined: the *CS-PDU non-P format* and the *CS-PDU P format*. The CS-PDU non-P format is constructed from 47 bytes of information supplied by the AAL user. The CS-PDU P format, on the other hand, is constructed from a 1-byte header and 46 bytes of information supplied by the AAL user. The 1-byte header contains an even parity bit and a 7-bit field that contains the *Structured Data Transfer (SDT)* pointer, which is used to point to the beginning of a block of data. The CS-PDU P format or non-P format becomes the payload of the SAR-PDU.

Structured and unstructured data transfers are used in *circuit emulation services (CES)*, which emulate a T1/E1 link over an ATM network using AAL 1. In order to implement CES, an *interworking function (IWF)* is required, as shown in Figure 3.15. The two IWFs are connected over the ATM network using AAL 1 via a bidirectional point-to-point virtual circuit connection. CES provides two different services: *DS1/E1 unstructured service* and *DS1/E1 $N \times 64$ Kbps structured service*. (For more details on CES, see Chapter 12).

In the unstructured service, CES simply carries an entire DS1/E1 signal over an ATM network. Specifically, IWF A receives the DS1 signal from user A and packs it bit-by-bit into the 47-byte CS-PDU non-P format, which in turn becomes the payload of a SAR-PDU. The SAR-PDUs then become the payload of ATM cells, which are transferred to IWF B. From there, the bits get delivered to user B as a DS1 signal.

In the structured service, CES provides a fractional DS1/E1 service, where the user only requires an $N \times 64$ Kbps connection. N can range from 1 to 24 for T1, and from 1 to 30 for E1. An $N \times 64$ Kbps service generates blocks of N bytes. These blocks are carried using the structured data transfer protocol. Such a block of data is referred to in the standards as a *structured block*. These blocks of N bytes are transported back-to-back over successive cells using the CS-PDU non-P format and P formats. The SDT pointer in the CS-PDU P format is used to help delineate the boundaries of these blocks. The actual rules as to how to use the SDT pointer in the P format are somewhat complex. Below, we describe these rules and then we give an example.

When the block is 1 byte long, the structured data transfer protocol generates only non-P format payloads. For block sizes greater than 1 byte, the structured data transfer protocol uses both the non-P and P formats. Recall that each SAR-PDU carries a CS-PDU P format or a non-P format in its payload, and a sequence count. CS provides both the CS-PDU (P or non-P format) and the sequence count. The sequence count ranges from 1 to 7, and eight successive SAR-PDUs with a sequence count from 1 to 7 form a *cycle*.

The CS-PDU P format is used only once within each cycle at the first available opportunity to point to the beginning of a structured block. The P format can only be used in an SAR-PDU with an even sequence count (i.e., 0, 2, 4, 6).

The SDT pointer gives the offset, expressed in bytes, between the end of the pointer field and the beginning of the first structured block within a 93-byte payload consisting

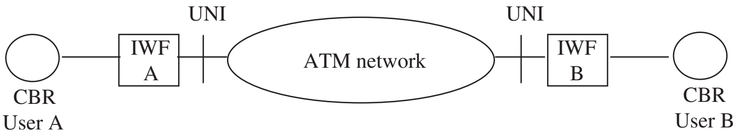


Figure 3.15 The interworking function in CES.

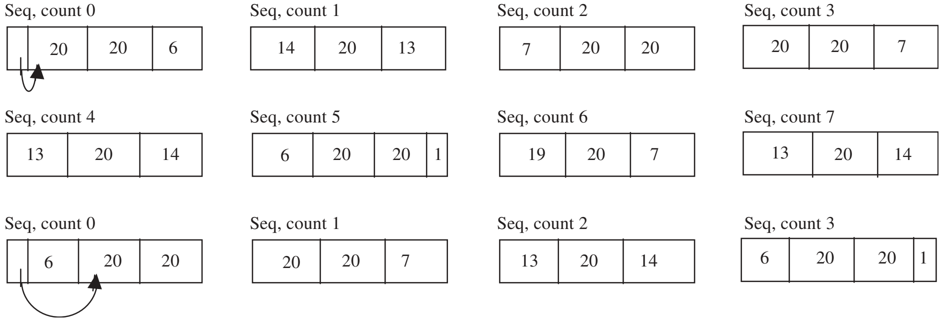


Figure 3.16 An example of the use of P format.

of the remaining 46 bytes of the CS-PDU P format and the 47 bytes of the next CS-PDU non-P format. This offset ranges between 0 and 93 inclusively.

After the AAL connection has been established, the first SAR-PDU to be transmitted has a sequence count of 0. It also contains a CS-PDU P format with the SDT value equal to 0. That is, the first byte of the structured block is placed in the second byte of the CS-PDU P format. In each subsequent cycle, the CS-PDU P format is used at the first available opportunity when the sequence count is an even number.

The offset value of 93 is used to indicate that the end of the 93-byte payload coincides with the end of a structured block.

If both the start and the end of a structured block are absent from a given cycle, then a P-format is used at the last opportunity, with the SDT pointer set to the dummy offset value of 127. That is, the P-format is used in the SAR-PDU with the sequence count 6.

Finally, let us consider what happens when the start of a structured block is absent from one cycle, but coincides with the beginning of the next cycle. In that case, the P format (with the SDT pointer set to 93) is used in the SAR-PDU with sequence count six, and the P format (with the SDT pointer set to 0) is used in the next cycle in the SAR-PDU with sequence count 0.

In the example shown in Figure 3.16, the structured block size is 20 bytes. Each box represents a CS-PDU; its sequence number is shown above the box. The first CS-PDU is a P format with the SDT pointer pointing to the second byte, which is the beginning of the first structured block. The first CS-PDU contains two complete blocks (i.e. 20 bytes each), and a third block with 6 bytes. The remaining CS-PDUs (i.e. with sequence count 1, 2, …, 7) are all non-P format CS-PDUs; these use the entire 47 bytes to pack the 20-byte blocks back-to-back. For instance, the CS-PDU with sequence number 1 contains the 14 remaining bytes of the third block (a full block) and 13 bytes of the fifth block. Note that the CS-PDU with sequence count 7 contains a partial block at the end with 14 bytes. The remaining bytes of this block are in the next CS-PDU, which starts a new cycle. This CS-PDU contains two complete blocks, and 1 byte of another block. In this case, the SDT pointer points to the beginning of the second block as shown in the figure.

Transfer of timing information

Some AAL users require that the clock frequency at the source be transferred to the destination. CS provides mechanisms for transferring such timing information.

If the sender's clock and the receiver's clock are phase-locked to a network's clock, then there is no need to transfer the source's clock frequency to the receiver, and AAL 1 is not required to transfer any timing information. However, if the two clocks are not phase-locked to a network's clock, then AAL 1 is required to transfer the source clock frequency to the receiver. This can be done using the *synchronous residual time stamp (SRTS)* method, where AAL 1 conveys to the receiver the difference between a common reference clock derived from the network and the sender's service clock. This information is transported over successive cells in the CSI bit of the SAR-PDU header. The common reference clock has to be available to both the sender and receiver. This is the case, for instance, when they are both attached to a synchronous network like SONET.

When a common reference clock is not available, then the *adaptive clock method* can be used. In this method, the receiver writes the received information into a buffer and then reads out from the buffer with a local clock. The fill level of the buffer is used to control the frequency of the local clock. To do this, the buffer fill is continuously measured; if it exceeds the median, then the local clock is assumed to be slow, and its speed is increased. If the fill level is lower than the median, then the clock is assumed to be fast and its speed is decreased.

3.7.2 ATM Adaptation Layer 2 (AAL 2)

This adaptation layer provides an efficient transport over ATM for multiple applications that are delay sensitive and have a low variable bit rate (such as voice, fax, and voiceband data traffic). AAL 2 is primarily used in cellular telephony. AAL 2 was designed to multiplex a number of such low variable bit rate data streams on to a single ATM connection. At the receiving side, it demultiplexes them back to the individual data streams. An example of AAL 2 is given in Figure 3.17. In this example, the transmitting AAL 2 multiplexes the data streams from users A, B, C, and D, onto the same ATM connection. The receiving AAL 2 demultiplexes the data stream into individual streams and delivers each stream from A, B, C, and D to its peer user A, B, C, and D, respectively.

CS, which provides the AAL 2 services, is further subdivided into the SSCS and the CPS. There is no SAR layer in AAL 2. The multiplexing of the different user data streams is achieved by associating each user with a different SSCS. Different SSCS protocols can be defined to support different types of service. Also, the SSCS might be null. Each SSCS receives data from its user and passes this data to the CPS in the form of short packets.

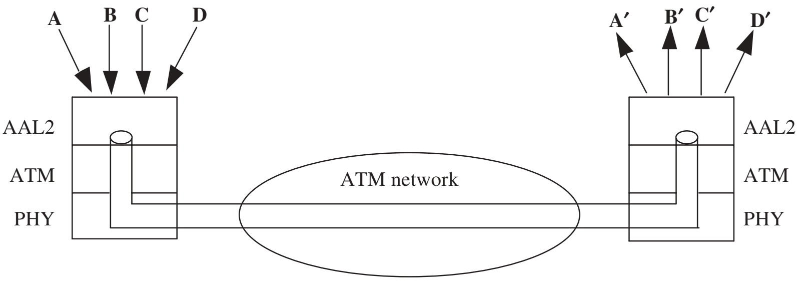


Figure 3.17 AAL 2 can multiplex several data streams.

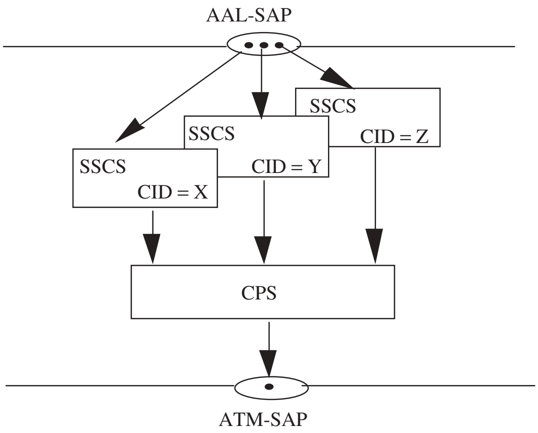


Figure 3.18 Functional model of AAL 2 (sender side).

The CPS provides a multiplexing function, whereby the packets received from different SSCS are all multiplexed onto a single ATM connection. At the receiving side of the ATM connection, these packets are retrieved from the incoming ATM cells by CPS and delivered to their corresponding SSCS receivers. Finally, each SSCS delivers its data to its user. The functional model of AAL 2 at the sender's side is shown in Figure 3.18.

A transmitting SSCS typically uses a timer to decide when to pass on data to CPS. When the timer expires, it passes the data that it has received from its higher-level layer application to CPS in the form of a packet, known as the *CPS-packet*. Since the applications that use AAL 2 are low variable bit rate, the CPS-packets are very short and can have a variable length. Each CPS-packet is encapsulated by CPS and then is packed into a CPS-PDU. As mentioned above, AAL 2 has been designed to multiplex several SSCS streams onto a single ATM connection. This is done by packing several CPS-packets into a single CPS-PDU, where each CPS-packet belongs to a different SSCS stream. A CPS-packet might potentially straddle two successive CPS-PDUs (see Figure 3.19). Note that CPS-packets 1 and 2 fit entirely in a CPS-PDU, whereas CPS-packet 3 has to be split between the first and second CPS-PDU. The point where a CPS-packet is split can occur anywhere in the CPS-packet, including the CPS-packet header. The unused payload in a CPS-PDU is padded with 0 bytes.

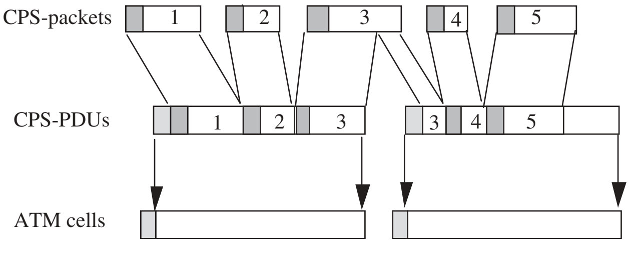
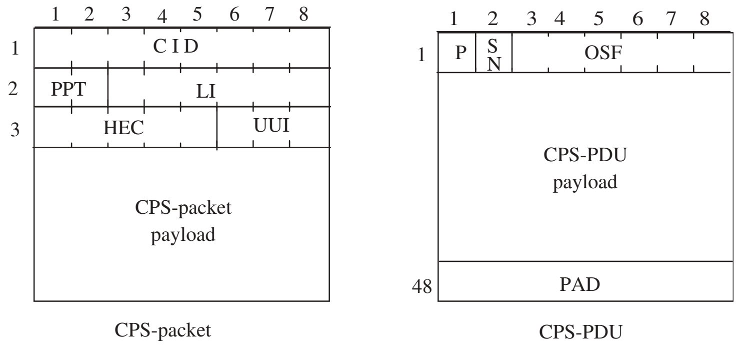


Figure 3.19 Packing CPS-packets into CPS-PDUs.



CPS-packet

CPS-PDU

Figure 3.20 The structure of the CPS-packet and CPS-PDU.

CPS-packets are encapsulated with a 3-byte header, and CPS-PDUs with a 1-byte header. The CPS-PDU, after encapsulation, is exactly 48 bytes long, and it becomes the payload of an ATM cell. The structure of the CPS-packet and the CPS-PDU is shown in Figure 3.20.

The header of the CPS-packet contains the following fields:

- *Channel identifier (CID)*: CPS can multiplex several streams, referred to as *channels*, onto a single ATM connection. The CID is an 8-bit field that identifies each channel. A channel is bidirectional, and the same CID value is used in both directions. CID values are allocated as follows: the 0 value is used as padding, and not as a channel identification; the 1 value is used by *AAL 2 negotiation procedure (ANP)* packets (described below); the 2 to 7 values are reserved; and the 8 to 255 values are valid CID values used to identify channels.
- *Packet payload type (PPT)*: The PPT is a 2-bit field that identifies the type of item being served by the CPS-packet. If the field variable is PPT $\neq 3$, then the CPS-packet is either serving a specific application (such as voice data) or carrying an ANP packet. If the field variable is PPT $= 3$, then the CPS-packet is serving an AAL network management function associated with managing the channel identified in the CID field.
- *Length indicator (LI)*: The total number of bytes in the payload of the CPS-packet is indicated in this 6-bit field. Its value is one less than the number of bytes in the CPS-packet payload. The default maximum length of the CPS-packet payload is 45 bytes. A maximum length of 64 bytes can be negotiated by ANP or by the network management protocol.
- *Header error control (HEC)*: This 5-bit field carries the FCS obtained from the CRC operation carried over the CID, PPT, LI, and UUI fields using the pattern $x^5 + x^2 + 1$. The receiver uses the contents of the HEC to detect errors in the header.
- *User-to-user-indication (UUI)*: This is a 3-bit field used for transferring information between the peer CPS users. The CPS transports this information transparently.

The header of the CPS-PDU is referred to as the *start field (STF)*, and it contains the following fields:

- *Parity (P)*: A 1-bit field used to detect errors in the STF. The transmitter sets this field so that the parity over the 8-bit STF is odd.

- *Sequence numbers (SN)*: A 1-bit field used to number modulo 2 the successive CPS-PDUs.
- *Offset field (OSF)*: The CPS-PDU payload can carry CPS packets in a variety of different arrangements. For instance, in the example given in Figure 3.19, the first CPS-PDU contains two complete CPS-packets (CPS-packets 1 and 2), followed by a partial CPS-packet (CPS-packet 3). The second CPS-PDU in Figure 3.19 consists of: the remainder of CPS-packet 3, two complete packets (CPS-packets 4 and 5), and padding. To aid the receiving CPS in extracting the CPS-packets from the CPS-PDU payload, a 6-bit *offset field (OSF)* is used to indicate the start of a new CPS-packet in the CPS-PDU payload. Specifically, OSF gives the number of bytes between the end of the STF (i.e., the header of the CPS-PDU) and the start of the first CPS-packet in the CPS-PDU payload. If the start of a CPS-packet is absent (which is signified by the value of "47"), then OSF gives the start of the PAD field.

Note that the AAL 2 CPS transfers the CPS-PDUs in a non-assured manner. That is, a CPS-PDU might be delivered, or it might be lost. Lost CPS-PDUs are not recoverable by retransmission.

The function that provides the dynamic allocation of AAL 2 channels on demand is called the *AAL negotiation procedures (ANP)*. This function is carried out by an AAL 2 layer management entity at each side of an AAL 2 link. This layer management entity uses the services provided by AAL 2 through a SAP for the purpose of transmitting and receiving ANP messages. These messages are carried on a dedicated AAL 2 channel with CID = 1. They control the assignment, removal, and status of an AAL 2 channel. The following types of messages have been defined: assignment request, assignment confirm, assignment denied, removal request, removal confirm, status poll, and status response.

3.7.3 ATM Adaptation Layer 5 (AAL 5)

AAL 5 is used for the transfer of data. Due to its simplicity, it is the most popular adaptation layer. AAL 5 services are provided by the CS and the SAR sublayer. CS is further subdivided into the SSCS and CPS (see Section 3.7). In the standards, CPS is referred to as the *common part convergence sublayer (CPCS)*.

Different SSCS protocols can be defined in order to support specific AAL users. The SSCS can be null, which is the assumption made in this section.

CPS provides a non-assured transfer of user-PDUs with any length that can vary from 1 byte to 65,535 bytes. At the receiving side, CPS can detect erroneous CPS-PDUs and it can indicate that to the higher-level application. However, since it provides a non-assured service, it does not recover erroneous CPS-PDUs by retransmission. This is left to a higher-level protocol, such as TCP. It also delivers user-PDUs in the order in which it received them.

CPS provides both message mode service and streaming mode service. In message mode, it is passed through a single user-PDU, which it transfers in a single CPS-PDU. In streaming mode, it is passed over time through several user-PDUs, which it blocks together and transports to the destination in a single CPS-PDU.

A user-PDU is encapsulated by CPS into a CPS-PDU by adding a trailer, as shown in Figure 3.21. The following fields in the trailer have been defined:

- *Padding (Pad)*: It can be between 0 and 47 bytes long, and is added so that the entire CPS-PDU including the padding and the remaining fields in the trailer becomes an

User-PDU	Pad	CPS-UU	CPI	Length	CRC-32

Figure 3.21 Encapsulation of a user-PDU.

integer multiple of 48 bytes. The padding is made up of unused bytes that do not convey any information.

- *CPS user-to-user indication (CPS-UU)*: A 1-byte field used to transfer transparently CPS user-to-user information.
- *Common part indicator (CPI)*: A 1-byte field to support future AAL 5 functions.
- *Length*: A 2-byte field used to indicate the length in bytes of the CPS-PDU payload, i.e. the user-PDU. The receiver can refer to this field to detect whether any information has been lost or gained.
- *CRC-32*: This 4-byte field contains the FCS calculated by the transmitting CPS over the entire contents of the CPS-PDU (i.e., the user-PDU, pad, CPS-UU, CPI, and length). The pattern used is: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.

The SAR sublayer fragments a CPS-PDU into a sequence of 48-byte segments, and each segment is carried in the payload of an ATM cell. The ATM cell that carries the last segment of a CPS-PDU is flagged by setting the SDU-type to 1 in its PTI field (see Table 3.2). Specifically, each ATM cell that carries a segment of a CPS-PDU has its SDU type indication set to 0, except the ATM cell that carries the last segment of the CPS-PDU whose PTI field contains the indication SDU type = 1.

The receiving SAR appends the payloads of the ATM cells into a buffer, associated with the VCC over which the ATM cells are being transmitted, until it either encounters an ATM cell with an indication SDU-type = 1 in its PTI field, or the CPS-PDU exceeds the buffer. Upon occurrence of either event, the buffer is passed on to a higher-level application with an indication as to whether the indication SDU-type = 1 was received or not, and in the case it was received, whether the CRC check was correct or not.

3.8 CLASSICAL IP AND ARP OVER ATM

Classical IP and ARP over ATM is a technique standardized by IETF designed to support IP over ATM in a single *logical IP subnet (LIS)*. A LIS is a group of IP hosts that have the same IP network address, say 192.43.0.0, and the same subnet mask (see Figure 3.22(a)). Let us assume that the LANs are replaced by three interconnected ATM switches (see Figure 3.22(b)). Each host can communicate directly with any other host in the subnetwork over an ATM connection. The traditional IP model remains unchanged and the IP router is still used to connect to the outside of the subnet.

The word *classical* in the *classical IP and ARP over ATM* scheme refers to the use of ATM in support of the IP protocol stack operating in a LAN environment.

IP packets are encapsulated using the IEEE 802.2 LLC/SNAP encapsulation. The protocol used in the payload, such as IP, ARP, Appletalk, and IPX, is indicated in the LLC/SNAP header. An encapsulated packet becomes the payload of an AAL 5 frame. The *maximum transfer unit (MTU)* is fixed to 9180 bytes. Adding an 8-byte LLC/SNAP header makes the total at 9188 bytes, which is the defaulted size for an AAL 5 frame.

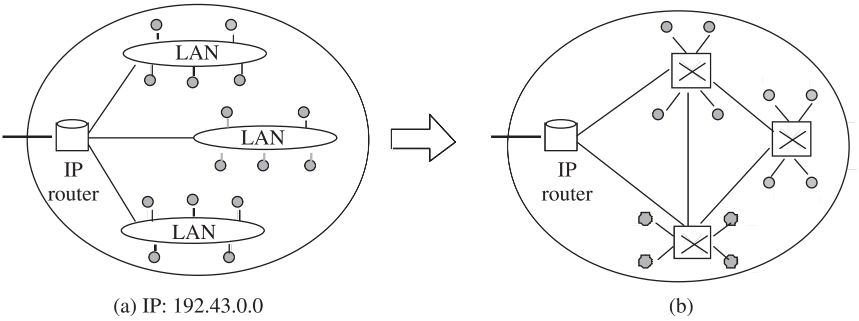


Figure 3.22 A logical IP subnet (LIS).

3.8.1 ATMARP

Each member of the LIS is configured with an IP address and an ATM address. When communicating with another member in the same LIS over ATM, it is necessary to resolve the IP address of the destination host with its ATM address. IP addresses are resolved to ATM addresses using the *ATMARP* protocol within the LIS. This protocol is based on ARP, see Section 6.1.3, and it has been extended to operate over a nonbroadcast unicast ATM network. The *inverse ATMARP (InATMARP)* protocol is used to resolve an ATM address to an IP address. It is based on RARP (see Section 6.1.3), but has been extended to support non-broadcast unicast ATM networks.

The ATMARP protocol uses an ATMARP server which can run on an IP host or an IP router and which must be located within the LIS. The LIS members are clients to the ATMARP server, and are referred to as *ATMARP clients*. The ATMARP server maintains a table or a cache of IP and ATM address mappings. It learns about the IP and ATM addresses of ATMARP clients through a registration process described below. At least one ATMARP server must be configured with each LIS. The following ATMARP messages have been defined.

- *ATMARP_request*: An ATMARP client sends an ATMARP request to the ATMARP server to obtain the ATM address of a destination ATMARP client. The message contains the client's IP and ATM addresses, and the IP address of the destination client.
- *ATMARP_reply*: This message is used by the ATMARP server to respond to an ATMARP_request with the requested ATM address. It contains the IP and ATM addresses of both the requesting and the destination clients.
- *ATMARP_NAK*: Negative response issued by the ATMARP server to an ATMARP_ request.
- *InATMARP_request*: Used to request the IP address of a destination. The message contains the sender's IP and ATM addresses and the destination's ATM address.
- *InATMARP_reply*: This is the response to an InATMARP_request with the destination's IP address. It contains the IP and ATM addresses of both the sender and the destination.

Registration

An ATMARP client must first register its IP and ATM addresses with the ATMARP server. To do this, the ATMARP protocol must be invoked as follows. Each ATMARP client is configured with the ATM address of the ATMARP server. After the client establishes a connection to the ATMARP server, it transmits an ATMARP_request on that connection. In the message, it provides its own IP and ATM addresses and it requests the ATM address of itself by providing its own IP address as the destination IP address. The ATMARP server checks against duplicate entries in its table, time stamps the entry, and adds it to its table. It confirms the registration of the ATMARP client by sending an ATMARP_reply. If a client has more than one IP address within the LIS, then it has to register each IP address with the ATMARP server.

Entries in the table of the ATMARP server are valid for a minimum of 20 minutes. If an entry ages beyond 20 minutes without being updated (refreshed), then the entry is removed from the table. Each ATMARP client is responsible for updating its entry in the ATMARP server's table at least every fifteen minutes. To do this, the same procedure for registering with the ATMARP server is used. That is, the ATMARP client sends an ATMARP_request to the ATMARP server with the destination IP address set to its own IP address. The ATMARP server updates the entry and confirms it by responding with an ATMARP_reply.

Address resolution

Let us assume that ATMARP client 1 wants to communicate with ATMARP client 2. Assume too that both clients are in the same LIS. If there is already an established connection between the two clients, traffic can flow immediately. Otherwise, a connection can be set up if client 1 knows the ATM address of the destination client 2. If its destination ATM address is not known, client 1 sends an ATMARP_request to the ATMARP server. If the server has the requested address in its table, it returns an ATMARP_reply. Otherwise, it returns an ATMARP_NAK. Upon receipt of the ATMARP_reply, a connection is established and traffic starts flowing.

An ATMARP client creates an entry in its ATMARP table for every connection (PVCs or SVCs) that it creates. An entry is valid for a maximum of fifteen minutes. When an entry has aged, the client must update it. If there is no open connection associated with the entry, then the entry is deleted. If the entry is associated with an open connection, then the client must update the entry prior to using the connection to transmit data. In the case of a PVC, the client transmits an InATMARP_request and updates the entry on receipt of the InATMARP reply. In the case of an SVC, it transmits an ATMARP_request to the ATMARP server, and updates the entry on receipt of the ATMARP_reply.

An ATMARP client is also permitted to initiate the above procedure for updating an entry in the table, before the entry has aged.

PROBLEMS

1. Why is there error control for the header and not for the payload of an ATM cell?
2. How long does it take to transmit an ATM cell over a link, when the link is:
   a) a T1 line?
   b) an OC-3?
   c) an OC-12?
   d) an OC-24?
   e) an OC-48?
3. Consider the HEC mechanism. Let p be the probability that a bit is received in error.

a) With what probability a cell is rejected when the HEC state machine is in the correction mode?
b) With what probability a cell is rejected when the HEC state machine is in the detection mode?
c) Assume that the HEC state machine is in the correction mode. What is the probability that n successive cells (where $n \geqslant 1$) will be rejected?
d) Assume that the HEC state machine is in the correction mode. What is the probability $p(n)$ that n successive cells will be accepted, where $n \geqslant 1$? (Hint: Write down the expression for $p(1)$ and $p(2)$, and express $p(3)$ as a function of $p(1)$ and $p(2)$. Then write down the general expression for $p(n)$ for any n as a function of $p(n-1)$ and $p(n-2)$.)

4. Consider the case of an application running over an ATM network. Assume that each packet generated by the application is carried by n ATM cells, which are transmitted back-to-back. The time to transmit a cell is T and the average time it takes for a cell to traverse the ATM network and reach the destination is D. When all of the ATM cells belonging to the same packet are received by the destination, their payloads are extracted from the cells and are assembled to the original packet. Subsequently, the CRC operation is carried out on the reassembled packet. If the CRC check is correct, the packet is released to the application. Otherwise, a negative acknowledgment is sent back to the source requesting the retransmission of the entire packet. The time it takes to carry out the CRC check is F, and the time it takes for the negative acknowledgment to reach the source is D. Let p be the probability that the ATM cell is received with erroneous payload.
a) What is the probability that all n cells are received correctly?
b) What is the probability that exactly m cells are received erroneously, where $m < n$.
c) Write down the expression for the average time, required to transfer correctly an application packet to the destination.
d) Plot the average time against the probability p, assuming that $n = 30$, $D = 20$ msec, $T = 3$ μsec, and $F = 0$. Vary p from 0.1 to 0.00001.
e) Discuss the results of the above plot in the light of the fact that there is no data link layer in ATM.

5. Consider the AAL 1 structured data transfer scheme. What is the value of the SDT pointer in the second and third cycle, when the structured block size is N bytes, $N = 5$, 10, 93, 156, 75, 1000.

6. Let us assume that the SDT pointer is not used in the AAL 1 structured data transfer scheme, i.e. no CS-PDU P format is used within each cycle. Give an example where the receiver might not be able to find the beginning of a structured block.

7. In AAL 2, the receiving CPS retrieves the CPS-packets carried in the payload of a CPS-PDU using the offset field (OSF) in the header of the CPS-PDU and the LI field in the header of each CPS-packet carried in the CPS-PDU. The use of the OSF might appear redundant! Construct an example where the receiving CPS cannot retrieve the CPS-packets carried in a CPS-PDU by using only the LI field in each CPS-packet. Assume no cell loss or corrupted ATM payloads.

8. The following CPS-packets have to be transmitted over the same AAL2 connection: CPS-packet 1 (20 bytes), CPS-packet 2 (48 bytes), CPS-packet 3 (35 bytes), and CPS-packet 4 (20 bytes). For simplicity, assume that the length of each of these CPS-packets includes the 3-byte CPS-packet header.
a) How many CPS-PDUs are required to carry these four CPS-packets?
b) What is the value of the OSF field in each CPS-PDU?

9. A voice source is active (talkspurt) for 400 msec and silent for 600 msec. Let us assume that a voice call is transported over an ATM network via AAL 2. The voice is coded to 32 Kbps and silent periods are suppressed. Assume that the SSCS has a timer set to 5 msec. That is, each time the timer expires, it sends the data it has gathered to CPS as a CPS-packet. Assume that the timer begins at the beginning of the busy period.
a) How long (in bytes) is each CPS-packet?
b) How many CPS-packets are produced in each active period?

10. A 1500-byte user-PDU is transported over AAL5.
    a) How many bytes of padding will be added?
    b) How many cells are required to carry the resulting CPS-PDU?
    c) What is the total overhead (i.e. additional bytes) associated with this user-PDU?

APPENDIX: SIMULATION PROJECT: AAL 2

The objective of this simulation project is to establish the value of the timer used in the CPS of AAL 2, that maximizes the average percentage fill of the ATM cells that carry CPS-PDUs, and minimizes the packetization delay in AAL 2.

Project Description

You will assume that there are two end devices interconnected with a VCC. The end devices support a number of phone calls multiplexed onto the same connection using AAL 2. We will only model the AAL 2 functions in one of the two end devices. The transfer of ATM cells across the ATM connection, the AAL 2 at the receiving end device, and the flow of information in the opposite direction will not be modeled. The number of voice calls multiplexed on the same connection will be provided as input to the simulation.

Each voice call is associated with an SSCS, which provides a number of different functions. Of interest to this simulation project is the voice encoding algorithm and the size of the audio packet. We will assume that *pulse code modulation (PCM)* is used which produces 8 bits per 125μsec. PCM is defined in ITU-T's standard G.711. The output of the G.711 coder is accumulated over an interval of 1 msec to yield a sequence of 8 bytes, referred to as the *encoded data unit (EDU)*. Further, five EDUs are blocked together into an audio packet of 40 bytes every 5 msec. SSCS also detects the beginning of a silence period and it transmits as special *silence insertion description (SID)* code.

CPS serves multiple SSCSs – each associated with a different phone call. The SSCSs operate asynchronously from each other. When CPS receives an audio packet, it encapsulates it into a CPS-packet and places it into the payload of a CPS-PDU (which equals 47 bytes). CPS keeps a timer T1. When the timer expires, it encapsulates whatever CPS-packets it has received into a CPS-PDU, adds padding if necessary, and sends the resulting CPS-PDU to the ATM layer. The amount of time it takes to carry out these tasks is assumed to be zero. If the CPS has not received any CPS-packets when T1 expires, then it does not create a CPS-PDU. T1 is reset immediately each time it expires. If the CPS payload is filled up before the timer expires, CPS prepares a CPS-PDU, sends it to the ATM layer, and resets the timer.

Each voice will be assumed to go through a talkspurt followed by a silence period. Only the bytes generated during each talkspurt are transported by AAL2, since the silence periods are suppressed. The rate of transmission of a voice during a talkspurt is 64 Kbps. The length of the talkspurt is assumed to be exponentially distributed, with an average of 400 msec. The length of the silence period is also assumed to be exponentially distributed, with an average of 600 msec.

The Simulation Structure

Simulation of a single SSCS

Since a voice alternates between a talkspurt and a silence period, the first thing to do is to generate the length of a talkspurt. When the talkspurt expires, generate a silence period; when it expires, generate a talkspurt; and so on, until the simulation is ended. Keep a status variable that indicates whether the voice is in the talkspurt state or in the silence state.

Use the following procedure to generate exponentially distributed talkspurts or silence periods:

1. Draw a random number r, $0 < r < 1$, using a random number generator.
2. $X = -(\text{mean})\log_e r$

For a talkspurt, set mean = 400 msec, and for a silence period, set mean = 600 msec. For simplicity, make sure that the generated talkspurts and silence periods are integers. This is easily done by taking the "floor" of each generated number. Remember to draw a new random r, $0 < r < 1$ each time you need one!

An SSCS can be simulated as follows:

1. Generate a talkspurt.
2. Every 5 msec, the SSCS produces an audio packet of 40 bytes, which is passed on to the CPS. Remember that the last audio packet might contain less than 40 bytes. Do not model the transmission of the SID code.
3. When the talkspurt expires, generate a silence period, during which no audio packets are created.
4. Go back to Step 1.

In this simulation, disregard the content of the audio packets or the content of the CPS-packet and CPS-PDU headers. Only consider the length (expressed in bytes) of each audio packet, CPS-packet, and CPS-PDU.

Simulation of the CPS layer

The CPS layer should be simulated as follows:

1. When CPS receives an audio packet, it creates a CPS-packet, and it checks to see whether the total number of bytes of the CPS packets received so far is equal to or more than 47. If yes, then it creates a CPS-PDU and passes it down to the ATM layer. Keep track of the leftover bytes of a CPS-packet that did not fit entirely into the CPS-PDU. Reset T1.
2. If T1 expires, CPS creates a CPS-PDU with the CPS-packets that it has received so far, adds padding, and passes the CPS-PDU down to the ATM layer. T1 is reset.
3. If no CPS-packets have been received when T1 expires, simply reset T1.

Event-based simulation

Two different simulation designs (*event-based* and *unit-time based*) can be used to build a simulation model. Below, we describe an event-based simulation design of the simulation model of the AAL 2 layer.

First identify the events that need to be monitored throughout the simulation:

- Events associated with each SSCS:
  1. Completion of a talkspurt.

  2. Completion of a silence period.
  3. During a talkspurt, completion of a 5-msec period.
- Events associated with CPS:
  1. T1 expires.
  2. CPS receives an audio packet, in which case it has to check for the event that the CPS-PDU payload becomes full.

All of the pending events in the simulation are kept in an event list. For each pending event, keep track of the type of event so that you will know what action to take when it occurs. Also keep an event clock that indicates the time that it will occur in the future. The event list can be implemented as a linked list, sorted in ascending order of the event clocks. The next event to occur is the one at the top of the event list. Alternatively, you could use a fixed array, where each row is dedicated to a particular event. To locate the next event, scan the array to find the row that corresponds to the smallest event clock. Some of the events in the fixed array might be void. To ensure that you do not select a void event, set its event clock to a very large number – one that is not likely to occur during the execution of the simulation program.

In addition to the event list, you should keep a *master clock (MCL)* that you advance each time an event occurs. The master clock always gives the current time of the simulation when an event occurs. Since all of the generated numbers are integers, all of the event clocks and the master clock are integers too.

Main structure of the event-based simulation

1. Locate the next event to occur in the future.
2. Set the master clock to the time the event occurs, and take appropriate action, depending on the type of event as indicated below.
3. If you have implemented the event list as a linked list, you have to insert any new events that you generate so that all of the event clocks are sorted in an ascending order. If you use a fixed table, then it is a matter of locating the appropriate row and update the event clock.
4. If total simulation time is not over, go back to Step 1. Otherwise, calculate final statistics, print, and stop.

Next event: completion of a silence period

1. Generate a talkspurt, set the corresponding event clock to the generated talkspurt plus MCL, and update the event list.
2. Generate the time the first 5-msec period will expire. That is, set the corresponding event clock to MCL + 5 and update the event list.

Next event: 5-msec period expires

1. Calculate the number of bytes generated.
2. Generate the time the next 5-msec period will expire That is, set the corresponding event clock to MCL + 5 and update the event list.
3. Check to see whether the CPS-PDU is full. That is, execute the logic associated with event: *CPS receives an audio packet.*

Next event: completion of a talkspurt

1. Generate the time the silence period will be completed, set the corresponding event clock to the generated silence period plus MCL, and update the event list.
2. Generate the number of bytes created and check to see whether the CPU-PDU payload is full. That is, execute the logic associated with event: *CPS receives an audio packet.*

Next event: T1 expires

1. The CPS-PDU is passed on to the ATM layer
2. Generate the next time that T1 will expire. That is, set the appropriate event clock to T1 + MCL, and update the event list.

Next event: CPS receives an audio packet

1. Create a CPS-packet and add it to the payload.
2. Check to see if the payload is equal to or greater than 47 bytes. If yes, the CPS-PDU is passed on to the ATM layer, and T1 is reset. That is, the corresponding event clock is set to T1 + MCL, and the event list is updated.

Remember that there might be left over bytes from the last CPS-packet. These should be added to the next CPS-PDU.

Because all of the event clocks are integers, multiple events are likely to be scheduled to occur simultaneously. The simulation will execute them all correctly, one at a time. To simplify matters, do not impose any particular order in which they should be executed.

Initialization

Each voice call starts its first talkspurt at a time t_b randomly chosen between (0,1000) msec. This time is calculated as follows: $t_b = 1000r$, where r is a random number, $0 < r < 1$.

What Information to Collect

Percentage of fill

Calculate the percentage fill of an ATM cell including the AAL 2 overheads as follows:

%fill_1 = (48 – total number of bytes of padding in the CPS-PDU)/48.

Calculate the percentage fill excluding the AAL 2 overheads as follows:

%fill_2 = (48 – 1 byte for the CPS-PDU header – 3 bytes per CPS-packet present in the payload of the CPS-PDU – total number of bytes of padding in the CPS-PDU)/48.

Calculate %fill_1 and %fill_2 each time you create a CPS-PDU, add cumulatively %fill_1 to Sum1 and %fill_2 to Sum2, and increase counter $N1$ by 1. At the end of the simulation, calculate the average value of %fill_1 by dividing Sum1 by $N1$, and the average value of %fill_2 by dividing Sum2 by $N1$.

Delay

Let D be the elapsed time from a moment an audio packet arrives to CPS to the moment that the CPS-PDU containing this CPE-packet is passed down to the ATM layer. (For a CPS-packet that straddles over two CPS-PDUs, consider only the part that fits into the second CPS-PDU). Calculate D for each audio packet, add it cumulatively to Sum3, and increase counter $N2$ by 1. At the end of the simulation, calculate the average delay by dividing Sum3 by $N2$. (You could also calculate confidence intervals of the above performance measures, but for this you need to consult a book on simulation techniques.)

Results

For each set of input values of T1 and number of voice calls, run your simulation for 2100 sec. At the end, calculate the average values of %fill_1, %fill_2, and the average delay by ignoring the values for %fill_1, %fill_2, and the delay that you collected in the first 100 msec. This is done because the first observations might be affected by the initial conditions of the simulation and the initial seed used in the random number generator. If you do not get smooth curves, increase the simulation run.

Run you simulation for different values of T1 and number of voice calls, and plot the average values of %fill_1, %fill_2, and the average delay as a function of T1 and number of voices. Determine from your curves the optimum value for T1. (The default value for T1 is T1 = 2 msec.)

4

Congestion Control in ATM Networks

Congestion control, otherwise known in the ATM Forum standards as *traffic management*, is a very important component of ATM networks. It permits an ATM network operator to carry as much traffic as possible so that revenues can be maximized without affecting the QoS offered to the users.

As we will see in this chapter, an ATM network can support several QoS categories. A new connection at call setup signals to the network the type of QoS category that it requires. If the new connection is accepted, the ATM network will provide the requested QoS to this connection without affecting the QoS of all of the other existing connections. This is achieved using congestion control in conjunction with a scheduling algorithm that is used to decide in what order cells are transmitted out of an ATM switch (see Section 3.6.2).

Two different classes of congestion control schemes exist: *preventive congestion control* and *reactive congestion control.* In preventive congestion control, as its name implies, we attempt to prevent congestion from occurring. This is done using the following two procedures: *call (or connection) admission control (CAC),* and *bandwidth enforcement.* Call admission control is exercised at the connection level; it determines whether or not to accept a new connection. Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the source transmitting on this connection is within its negotiated traffic parameters.

Reactive congestion control is based on a totally different philosophy than preventive congestion control. In reactive congestion control, the network uses feedback messages to control the amount of traffic that an end device transmits so that congestion does not arise.

In this chapter, we first present the parameters used to characterize ATM traffic, the QoS parameters, and the ATM QoS categories. Then, we describe in detail the preventive and the reactive congestion control schemes.

4.1 TRAFFIC CHARACTERIZATION

The traffic submitted by a source to an ATM network can be described by the following traffic parameters: *peak cell rate (PCR), sustained cell rate (SCR), maximum burst size (MBS), burstiness*, and *correlation of inter-arrival times.* Also, various probabilistic and empirical models have been used to describe the arrival process of cells. Below, we

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

examine these traffic parameters in detail and we briefly introduce some empirical and probabilistic models. Two additional parameters – *cell delay variation tolerance (CDVT)* and *burst tolerance (BT)* – will be introduced later on in this chapter.

4.1.1 Types of Parameters

The types of traffic parameters are described below.

Peak cell rate (PCR)

This is the maximum amount of traffic that can be submitted by a source to an ATM network, and is expressed as ATM cells per second. Due to the fact that transmission speeds are expressed in bits per second, it is more convenient to talk about the peak bit rate of a source, i.e., the maximum number of bits per second submitted to an ATM connection, rather than its peak cell rate. The peak bit rate can be translated to the peak cell rate, and vice versa, if we know which ATM adaptation layer is used. The peak cell rate was standardized by both the ITU-T and the ATM Forum.

Sustained cell rate (SCR)

Let us assume that an ATM connection is up for a duration equal to D. During that time, the source associated with this connection transmits at a rate that varies over time. Let S be the total number of cells transmitted by the source during the period D. Then, the average cell rate of the source is S/D. (One would be inclined to use the acronym *ACR* for *average cell rate,* but *ACR* was already taken, to indicate *allowed cell rate* in the ABR mechanism, described in Section 4.8.1.)

The average cell rate was not standardized by ITU-T and by the ATM Forum. Instead, an upper bound of the average cell rate, known as the *sustained cell rate (SCR),* was standardized by the ATM Forum. This is obtained as follows.

Let us first calculate the average number of cells submitted by the source over successive short periods T. For instance, if the source transmits for a period D equal to 30 minutes and T is equal to one second, then there are 1800 T periods and we will obtain 1800 averages (one per period). The largest of all of these averages is called the *sustained cell rate (SCR).* Note that the SCR of a source cannot be larger than the source's PCR, nor can it be less than the source's average cell rate.

The SCR is not to be confused with the average rate of cells submitted by a source. However, if we set T equal to D, then the SCR simply becomes the average cell rate at which the source submits cells to the ATM network. For instance, in the above example, the SCR will be equal to the average cell rate, if T is equal to 30 minutes. The value of T is not defined in the standards, but is often assumed to be one second in industry practice.

Maximum burst size (MBS)

Depending upon the type of the source, cells might be submitted to the ATM network in bursts. These bursts are either fixed or variable in size. For instance, in a file transfer, if the records retrieved from the disk are of fixed size, then each record results to a fixed number of ATM cells submitted to the network back-to-back. In an encoded video transfer, however, each coded image has a different size, which results to a variable number of cells submitted back-to-back. The *maximum burst size (MBS)* is defined as the maximum number of cells that can be submitted by a source back-to-back at peak cell rate. The MBS was standardized by the ATM Forum.

Burstiness

This is a notion related as to how the cells transmitted by a source are clumped together. Typically, a source is *bursty* if it transmits for a time and then becomes idle for a time, as shown in Figure 4.1. The longer the idle period, and the higher the arrival rate during the active period, the more bursty the source is.

The burstiness of a source can significantly affect the cell loss in an ATM switch. Let us consider an output buffer of the output buffering non-blocking ATM switch. (The switch is shown in Figure 3.10; the buffer is shown in Figure 4.2.) The buffer has a finite capacity queue and is served by a link (see the circle in Figure 4.2). The arrival stream of ATM cells to the queue can be seen as the superposition of several different arrival streams coming from the input ports of the switch. A cell that arrives at a time when the queue is full is lost.

From queueing theory, we know that as the arrival rate increases, the cell loss increases as well. What is interesting to observe is that a similar behavior can be also seen for the burstiness of a source. The curve in Figure 4.3 shows qualitatively how the cell loss rate increases as the burstiness increases while the arrival rate remains constant.

Correlation

Let us consider successive inter-arrival times of cells generated by a source (see Figure 4.4). In an ATM environment, it is highly likely that the inter-arrival times are correlated either positively or negatively. Positive correlation means that, if an inter-arrival time is large (or small), then it is highly likely that the next inter-arrival time will also be large (or small). Negative correlation implies the opposite. That is, if an inter-arrival time is large (or small), then it is highly likely that the next inter-arrival time will be small (or large). As in the case

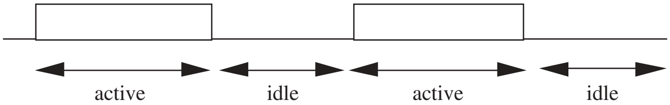


Figure 4.1 A bursty source.

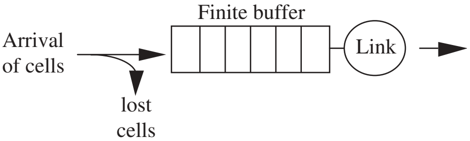


Figure 4.2 A finite capacity buffer.

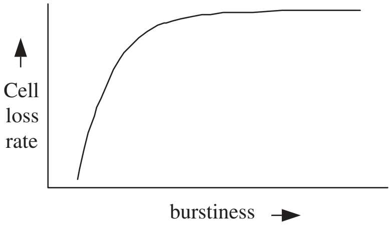


Figure 4.3 Cell loss rate vs burstiness.

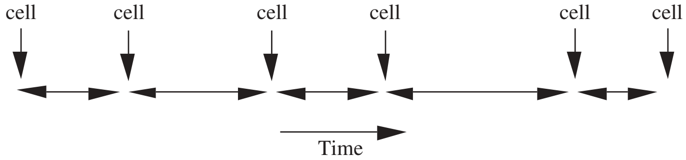


Figure 4.4 Successive inter-arrival times of cells.

of burstiness, the correlation of the inter-arrival time of cells can significantly affect the cell loss probability in an ATM switch.

4.1.2 Standardized Traffic Descriptors

The ATM Forum has standardized the following traffic descriptors: peak cell rate, cell delay variation tolerance, sustained cell rate, and maximum burst size. The ITU-T has only standardized the peak cell rate. The source's characteristics determine the peak cell rate, sustained cell rate, and maximum burst size.

The cell delay variation tolerance is used in the *generic cell rate algorithm (GCRA).* (See Section 4.7.1.) It is independent of the characteristics of the source, and is specified by the administrator of the network to which the source is directly attached.

4.1.3 Empirical Models

Several empirical models have been developed to predict the amount of traffic generated by a variable bit rate MPEG video-coding algorithm. These empirical models are statistical models and are based on regression techniques.

MPEG is a standards group in ISO that is concerned with the issue of compression and synchronization of video signals. In MPEG, successive video frames are compressed following a format like: I B B B P B B B P B B B I, where *I* stands for I-frame, *B* for B-frame, and *P* for P-frame. An *intra-coded frame (I-frame)* is an encoding of a picture based entirely on the information in that frame. A *predictive-coded frame (P-frame)* is based on motion compensated prediction between that frame and the previous I-frame or P-frame. A *bidirectional-coded frame (B-frame)* is based on motion compensated prediction between that frame and the previous I- or P-frame or the next I-frame or P-frame.

The encoder also can select the sequence of I, P, and B frames, which form a group of frames known as a *group of pictures (GOP).* The group of frames repeats for the entire duration of the video transmission.

The size of the resulting frame varies significantly between frame types. I-frames are the largest while B-frames are the smallest. The size of an I-frame varies based on picture content. P-frames and B-frames vary depending on the motion present in the scene as well as picture content.

The number of bits produced by each frame in such a sequence is correlated and it can be predicted using an *autoregressive integrated moving average (ARIMA)* model. Such a model can be used in a performance evaluation study to generate video traffic. For the group of pictures I B B P B B P B B P B B, the following ARIMA model can be used to predict the number of bits $S(i)$ of the *ith* frame: $S(i) = S(i-12) + e(i) - 0.69748\ e(i-3)$, where $e(i)$ is white noise and it follows the distribution $N(0, \sigma^2)$, with $\sigma^2 = 4849.5$ bits.

4.1.4 Probabilistic Models

Probabilistic models of arrival processes are abstractions of real-life arrival processes. They do not represent real-life arrival processes exactly, but they capture some of the traffic parameters described above, and in view of this, they are extremely useful in performance evaluation studies.

When we talk about a probabilistic model of an ATM arrival process, we assume that the arrival process is generated by a source that transmits cells over an ATM link. The link is assumed to be used exclusively by this source, and is slotted with a slot being equal to the time it takes for the link to transmit a cell. If we were to place ourselves in front of the link and observe the slots go by, then we would see that some of the slots carry a cell while others are empty. A model of an ATM arrival process describes which slots carry a cell and which slots are idle.

ATM sources are classified into *constant bit rate (CBR)* and *variable bit rate (VBR).* A CBR source generates the same number of bits every unit time whereas a VBR source generates traffic at a rate that varies over time. Examples of CBR sources are circuit emulation services such as T1 and E1, unencoded voice, and unencoded video. Examples of VBR sources are encoded video, encoded voice with suppressed silence periods, IP over ATM, and frame relay over ATM. The arrival process of a CBR source is easy to characterize. The arrival process of a VBR source is more difficult to characterize and it has been the object of many studies.

CBR sources

As mentioned above, a CBR source generates the same number of bits every unit time. For instance, a 64-Kbps unencoded voice produces 8 bits every 125 msec. Since the generated traffic stream is constant, the PCR, SCR, and average cell rate of a CBR source are all the same, and a CBR source can be completely characterized by its PCR.

Let us assume that a CBR source has a PCR equal to 150 cells per second, and the ATM link over which it transmits has a speed, expressed in cells per second, of 300. Then, if we observe the ATM link, we will see that every other slot carries a cell. If the speed of the link is 450 cells per second, then every third slot carries a cell, and so on.

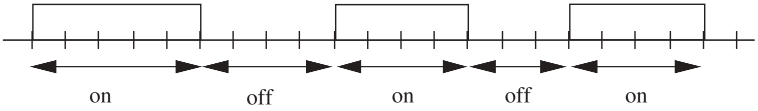


Figure 4.5 The on/off process.

VBR sources

A commonly used traffic model for data transfers is the *on/off process* (see Figure 4.5). In this model, a source is assumed to transmit only during an active period, known as the *on period*. This period is followed by a silent period, known as the *off period*, during which the source does not transmit. This cycle of an on period followed by an off period repeats continuously until the source terminates its connection. During the on period, a cell might be transmitted with every slot or with every fixed number of slots, depending upon the source's PCR and the link speed.

The PCR of an on/off source is the rate at which it transmits cells during the on period. For example, if it transmits every other slot, then its PCR is equal to half the speed of the link, where the link's speed is expressed in cells per second. Alternatively, we can say that the source's peak bit rate is half the link's capacity, expressed in bits per second. The average cell rate is:

$$\frac{\text{PCR} \times \text{mean length of on period}}{\text{mean length of on and off period}}$$

The on/off model captures the notion of burstiness, which is an important traffic characteristic in ATM networks. The burstiness of a source is indicative of how cells are clumped together. There are several formulas for measuring burstiness. The simplest formula is the ratio of the mean length of the on period, divided by the sum of the mean on and off periods:

$$\text{r} = \frac{\text{mean on period}}{\text{sum of mean on and off periods}}$$

This quantity can be also seen as the fraction of time that the source is active transmitting. When r is close to 0 or 1, the source is not bursty. The burstiness of the source increases as r approaches 0.5. Another commonly used measure of burstiness, but more complicated to calculate, is the squared coefficient of variation of the inter-arrival times defined by Var $(X)/(E\ (X))^2$, where X is a random variable indicating the inter-arrival times.

The length of the on and off periods of the on/off process follow an arbitrary distribution. A special case of the on/off process is the well-known *interrupted Bernoulli process (IBP),* which has been used extensively in performance studies of ATM networks. In an IBP the on and off periods are geometrically distributed and cells arrive during the on period in a Bernoulli fashion. That is, during the on period, each slot contains a cell with probability α, or it is empty with probability $1 - \alpha$.

The IBP process can be generalized to the two-state *Markov modulated Bernoulli process (MMBP).* A two-state MMBP consists of two alternating periods, Periods 1 and 2. Each period is geometrically distributed. During Period i, we have Bernoulli arrivals with rate $\alpha_i, i = 1, 2$. That is, each slot during Period i has α_i probability of containing

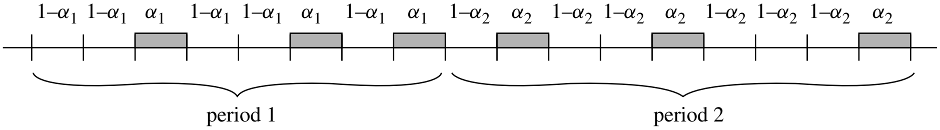

1–α1 1–α1 α1 1–α1 1–α1 α1 1–α1 α1 1–α2 α2 1–α2 1–α2 α2 1–α2 1–α2 1–α2 α2
period 1
period 2

Figure 4.6 The two-state MMBP.

a cell (see Figure 4.6). Transitions between the two periods are as follows:

$$\begin{array}{c} \\ \text{Period 1} \\ \text{Period 2} \end{array} \begin{array}{c} \begin{array}{cc} \text{Period 1} & \text{Period 2} \end{array} \\ \left[\begin{array}{cc} \text{p} & 1-\text{p} \\ 1-\text{q} & \text{q} \end{array}\right] \end{array}$$

That is, if the process is in Period 1 (Period 2), then in the next slot it will be in the same period with probability $p(q)$ or it will change to Period 2 (Period 1) with probability $1 - p(1 - q)$. A two-state MMBP model captures both the notion of burstiness and the correlation of inter-arrival times. More complicated MMBPs can be obtained using n different periods.

The above arrival processes were defined in discrete time. That is, we assumed that the link is slotted, and the length of the slot is equal to the time it takes to transmit a cell. Similar arrival processes have been defined in continuous time. In this case, the underlying assumption is that the link is not slotted and the arrival of an ATM cell can occur at any time. The continuous-time equivalent of the IBP is the *interrupted Poisson process (IPP)* which is a well known process used in teletraffic studies. In an IPP the on and off periods are exponentially distributed and cells arrive in a Poisson fashion during the on period. An alternative model can be obtained using the fluid approach. In this case, the on and off periods are exponentially distributed as in the IPP model, but the arrivals occur during the on period at a continuous rate, like fluid flowing in. This model, referred to as the *interrupted fluid process (IFP)*, has been used extensively in performance studies.

The IPP can be generalized to a two-state *Markov modulated Poisson process (MMPP),* which consists of two alternating periods, period 1 and period 2. Each period $i, i = 1, 2$, is exponentially distributed with a mean $1/\mu_i$ and during the *ith* period arrivals occur in a Poisson fashion at the rate of γ_i. More complicated MMPPs can be obtained using n different periods.

4.2 QUALITY OF SERVICE (QOS) PARAMETERS

A number of different parameters can be used to express the QoS of a connection, such as *cell loss rate (CLR), jitter, cell transfer delay (CTD), peak-to-peak cell delay variation*, and *maximum cell transfer delay (max CTD).*

The *cell loss rate (CLR)* is a very popular QoS parameter. It was the first one to be used in ATM networks. This is not surprising, since there is no flow control between two adjacent ATM switches or between an end device and the switch to which it is attached. Also, cell loss is easy to quantify, as opposed to other QoS parameters such as jitter and cell transfer delay. Minimizing the cell loss rate in an ATM switch has been used

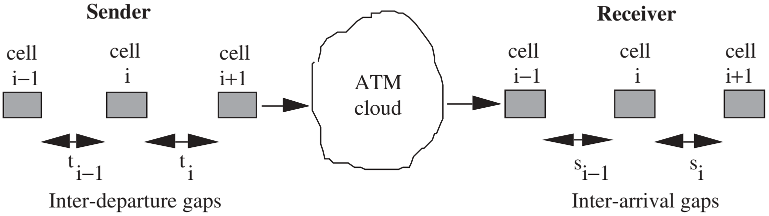


Figure 4.7 Inter-departure and inter-arrival gaps.

as a guidance to dimensioning ATM switches, and also a large number of call admission control algorithms were developed based on the cell loss rate.

The jitter is an important QoS parameter for real-time applications, such as voice and video. In these applications, the inter-arrival gap between successive cells at the destination end device cannot be greater than a certain value, as this can cause the receiving play-out process to pause. In general, the inter-departure gaps between successive cells transmitted by the sender are not the same as the inter-arrival gaps at the receiver. Let us consider Figure 4.7. The gap between the end of the transmission of the ith cell and the beginning of the transmission of the $(i + 1)$st cells is t_i. The gap between the end of the arrival of the ith cell and the beginning of the arrival of the $(i + 1)$st cell is s_i. The inter-departure gap t_i can be less, equal, or greater than s_i. This is due to buffering and congestion delays in the ATM network. This variability of the inter-arrival times of cells at the destination is known as jitter.

It is important that the service provided by an ATM network for a voice or a video connection is such that the jitter is bounded. If the inter-arrival gaps s_i are less than the inter-departure gaps t_i, then the play-out process will not run out of cells. (If this persists for a long time, however, it might cause overflow problems). If the inter-arrival gaps are consistently greater than the inter-departure gaps, then the play-out process will run out of cells and will pause. This is not desirable, because the quality of the voice or video delivered to the user will be affected. Bounding jitter is not easy to accomplish.

The *cell transfer delay (CTD)* is the time it takes to transfer a cell end-to-end; in other words, from the UNI of the transmitting end device to the UNI of the receiving end device. CTD is made up of a fixed component and a variable component. The fixed cell transfer delay is the sum of all fixed delays that a cell encounters from the transmitting end device to the receiving end device, such as propagation delay, fixed delays induced by transmission systems, and fixed switch processing times. The variable cell transfer delay, known as the *peak-to-peak cell delay variation*, is the sum of all variable delays that a cell encounters from the transmitting end device to the receiving end device. These delays are primarily due to queueing delays in the switches along the cell's path. The peak-to-peak cell delay variation should not to be confused with the *cell delay variation tolerance (CDVT),* which is used in the *generic cell rate algorithm (GCRA)* described in Section 4.7.1.

The *maximum cell transfer delay (max CTD)* is another QoS parameter that defines an upper bound on the end-to-end cell transfer delay. This upper bound is not an absolute bound. Rather, it is a statistical upper bound, which means that the actual end-to-end cell transfer delay might occasionally exceed the max CTD. That is, the sum of the fixed

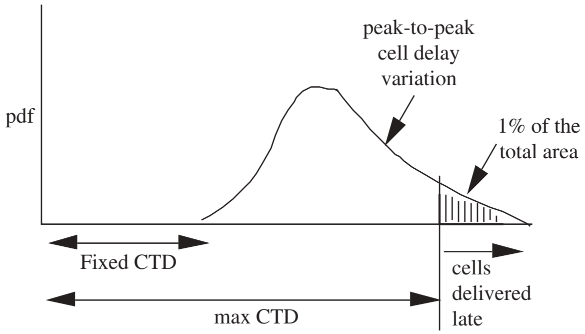


Figure 4.8 Cell transfer delay.

cell transfer delay and the peak-to-peak cell delay variation might exceed max CTD (see Figure 4.8). For example, let us assume that the max CTD is set to 20 msec and the fixed CTD is equal to 12 msec. Then, there is no guarantee that the peak-to-peak cell delay variation will always be less than 8 msec. The max CTD can be obtained as a percentile of the end-to-end cell transfer delay, so that the end-to-end cell transfer delay exceeds it only a small percent of the time. For instance, if it is set to the 99th percentile, then 99% of the time the end-to-end cell transfer delay will be less than the max CTD and 1% of the time it will be greater.

Of the QoS parameters described above, the CLR, the peak-to-peak cell delay variation, and the max CTD were standardized by the ATM Forum and can be signaled at call setup time. That is, at call setup time, the calling party can specify values for these parameters. These values are the upper bounds, and represent the highest acceptable (and consequently the least desired) values. The values for the peak-to-peak cell delay variation and for the max CTD are expressed in msec. As an example, the calling party can request that the CLR is less or equal than 10^{-6}, the peak-to-peak cell delay variation is less or equal than 3 msec, and the max CTD is less or equal than 20 msec.

The network will accept the connection, if it can guarantee the requested QoS values. If it cannot guarantee these values then it will reject the connection. Also, the network and the calling party might negotiate new values for the QoS parameters. As will be seen in the following section, the number of QoS parameters signaled at call setup time depends on the type of ATM service requested by the calling party.

Three additional QoS parameters are used: the *cell error rate (CER),* the *severely errored cell block ratio (SECBR),* and the *cell misinsertion rate (CMR).* These three parameters are not used by the calling party at call set-up. They are only monitored by the network.

The CER of a connection is the ratio of the number of *errored cells* (i.e., the cells delivered to the destination with erroneous payload) to the total number of cells transmitted by the source.

The *severely errored cell block ratio (SECBR)* is the ratio of the total number of severely errored *cell blocks* divided by the total number of transmitted cell blocks. A cell block is a sequence of cells transmitted consecutively on a given connection. A severely errored cell block occurs when more than a predefined number of errored cells, lost cells, or misinserted cells are observed in a received cell block.

The *cell misinsertion rate (CMR)* is the number of cells delivered to a wrong destination divided by a fixed time interval. A misinserted cell is a cell transmitted on a different connection due to an undetected error in its header.

4.3 ATM SERVICE CATEGORIES

An ATM service category is, in simple terms, a QoS class. Each service category is associated with a set of traffic parameters and a set of QoS parameters. Functions such as call admission control and bandwidth allocation (see Section 4.6) are applied differently for each service category. Also, the scheduling algorithm that determines in what order the cells in an output buffer of an ATM switch are transmitted out, provides different priorities to cells belonging to different service categories (see Section 3.6.2). In addition, a service category might be associated with a specific mechanism that is in place inside the network. The service category of a connection is signaled to the network at call setup time, along with its traffic and QoS parameters.

The ATM Forum has defined the following six 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), available bit rate (ABR),* and *guaranteed frame rate (GFR).* The first two service categories (CBR and RT-VBR) are for real-time applications; the remaining service categories are for non-real-time applications.

Note that there is no restriction as to which ATM adaptation layer should be used for each service category.

4.3.1 The CBR Service

This service is intended for real-time applications which transmit at constant bit rate, such as circuit emulation services and constant-bit rate video.

Since the rate of transmission of a constant-bit rate application does not change over time, the peak cell rate is sufficient to describe the amount of traffic that the application transmits over the connection. The *cell delay variation tolerance (CDVT)* is also specified (see Section 4.7.1). A CBR service is for real-time applications, and therefore, the end-to-end delay is an important QoS parameter. In view of this, in addition to the CLR, the two delay-related parameters (peak-to-peak cell delay variation and the max CTD) are specified.

In summary, the PCR and CDVT traffic parameters are specified. Also, the following QoS parameters are specified: CLR, peak-to-peak cell delay variation, and max CTD.

4.3.2 The RT-VBR Service

This service is intended for real-time applications that transmit at a variable bit rate, such as encoded video and encoded voice.

Since the rate of transmission of a variable-bit rate application varies over time, the peak cell rate is not sufficient to describe the amount of traffic that the application will transmit over the connection. In addition to the PCR and the cell delay variation tolerance, the *sustained cell rate (SCR)* and the *maximum burst size (MBS)* are specified. As in the CBR service, the RT-VBR service is also intended for real-time applications. Therefore, in addition to the CLR, the two delay-related parameters (peak-to-peak cell delay variation and the max CTD) are specified.

In summary, the following traffic parameters are specified: PCR, CDVT, SCR, and MBS. Also, the following QoS parameters are specified: CLR, peak-to-peak cell delay variation, and max CTD.

4.3.3 The NRT-VBR Service

This service is intended for non-real-time applications that transmit at a variable bit rate. As in the RT-VBR service, the traffic parameters PCR, the *cell delay variation tolerance (CDVT),* the *sustained cell rate (SCR)* and the *maximum burst size (MBS)* are specified. Since this service is not intended for real-time applications, only the CLR is specified.

In summary, the following traffic parameters are specified: PCR, CDVT, SCR, MBS. Also, the CLR QoS parameter is specified.

4.3.4 The UBR Service

This is a best-effort type of service for non-real-time applications with variable bit rate. It is intended for applications that involve data transfer, such as file transfer, Web browsing, and email. No traffic or QoS parameters are specified.

The PCR and the CDVT can be specified, but a network can ignore them. Similarly, a UBR user can indicate a *desirable minimum cell rate (DMCR),* but a network is not required to guarantee such as a minimum bandwidth.

4.3.5 The ABR Service

This service is intended for non-real-time applications that can vary their transmission rate according to the congestion level in the network.

A user requesting the ABR service specifies a *minimum cell rate (MCR)* and a maximum cell rate, which is its PCR. The minimum cell rate could be 0. The user varies its transmission rate between its MCR and its PCR in response to feedback messages that it receives from the network. These feedback messages are conveyed to the user through a mechanism implemented in the network. During the time that the network has a slack capacity, the user is permitted to increase its transmission rate by an increment. When congestion begins to build up in the network, the user is requested to decrease its transmission rate by a decrement. A detailed description of the ABR service is given in Section 4.8.1.

The following traffic parameters are specified: PCR, CDVT, and MCR. The CLR for ABR sources is expected to be low. Depending upon the network, a value for the CLR can be specified.

4.3.6 The GFR Service

This service is for non-real-time applications that require a *minimum cell rate (MCR)* guarantee, but they can transmit in excess of their requested MCR. The application transmits data organized into frames, and the frames are carried in AAL 5 CPS-PDUs. The network does not guarantee delivery of the excess traffic. When congestion occurs, the network attempts to discard complete AAL 5 CPS-PDUs rather than individual cells. The GFR service does not provide explicit feedback to the user regarding the current level of congestion in the network. Rather, the user is supposed to determine network congestion through a mechanism such as TCP, and adapt its transmission rate.

The following traffic parameters are specified: PCR, MCR, *maximum burst size (MBS)* and *maximum frame size (MFS)*. The CLR for the frames that are eligible for the service guarantee is expected to be low. Depending upon the network, a value for the CLR can be specified.

4.3.7 ATM Transfer Capabilities

In the ITU-T standard, the ATM service categories are referred to as *ATM transfer capabilities*. Some of the ATM transfer capabilities are equivalent to ATM Forum's service categories, but they have a different name. The CBR service is called the *deterministic bit rate (DBR)* service, the RT-VBR service is called the *real-time statistical bit rate (RT-SBR)* service, and the NRT-VBR service is called the *non-real-time statistical bit rate (NRT-SBR)* service. The UBR service category has no equivalent ATM transfer capability. Both the ABR and GFR services were standardized by ITU-T. Finally, the ITU-T ATM transfer capability *ATM block transfer (ABT),* described in Section 4.6.3, has no equivalent service category in the ATM Forum standards.

4.4 CONGESTION CONTROL

Congestion control procedures can be grouped into the following two categories: *preventive control* and *reactive control.*

In preventive congestion control, as its name implies, we attempt to prevent congestion from occurring. This is achieved using the following two procedures: *call admission control* (alternately, *connection admission control*) *(CAC)* and *bandwidth enforcement.* CAC is exercised at the connection level; it determines whether or not to accept a new connection. Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the source transmitting on this connection is within its negotiated traffic parameters.

Reactive congestion control is based on a different philosophy than preventive congestion control. In reactive congestion control, the network uses feedback messages to control the amount of traffic that an end device transmits so that congestion does not arise.

In the remaining sections of the chapter, we examine in detail various preventive and reactive congestion control schemes.

4.5 PREVENTIVE CONGESTION CONTROL

As mentioned above, preventive congestion control involves the following two procedures: *call admission control (CAC)* and *bandwidth enforcement.* CAC is used by the network to decide whether to accept a new connection or not.

As we have seen so far, ATM connections can be either *permanent virtual connections (PVC)* or *switched virtual connections (SVC).* A PVC is established manually by a network administrator using network management procedures, whereas an SVC is established in real-time by the network using the signaling procedures described in Chapter 5.

In our discussion below, we will consider a point-to-point SVC. Recall that point-to-point connections are bidirectional. The traffic and QoS parameters can be different for each direction of the connection.

Let us assume that an end device, referred to as end device 1, wants to set up a connection to a destination end device, referred to as end device 2. A point-to-point SVC is established between the two end devices, when end device 1 sends a SETUP message to its ingress switch (let us call it switch A), requesting that a connection is established to end device 2. Using a routing algorithm, the ingress switch calculates a path through the network to the switch that the destination end device is attached to (let us call it switch B). Then, it forwards the setup request to its next hop switch, which in turn forwards it to its next hop switch, and so on until it reaches switch B. Switch B sends the setup request to end device 2, and if it is accepted, a confirmation message is sent back to end device 1.

The setup message, as will be seen in chapter 5, contains a variety of different types of information, including values for the traffic and QoS parameters. This information is used by each switch in the path to decide whether it should accept or reject the new connection. This decision is based on the following two questions:

- Will the new connection affect the QoS of the existing connections already carried by the switch?
- Can the switch provide the QoS requested by the new connection?

As an example, let us consider a non-blocking ATM switch with output buffering (see Figure 3.10), and let us assume that the QoS is measured by the cell loss rate. Typically, the traffic that a specific output port sees is a mixture of different connections that enter the switch from different input ports. Let us assume that, so far, the switch provides a cell loss probability of 10^{-6} for each existing connection routed through this output port. Now, let us assume that the new connection requests a cell loss rate of 10^{-6}. What the switch has to decide is whether the new cell loss rate for both the existing connections and the new connection will be 10^{-6}. If the answer is yes, then the switch can accept the new connection. Otherwise, it will reject it.

Each switch on the path of a new connection has to decide independently of the other switches whether it has enough bandwidth to provide the QoS requested for this connection. This is done using a CAC algorithm. Various CAC algorithms are discussed in the following section.

If a switch along the path is unable to accept the new connection, then the switch refuses the setup request and it sends it back to a switch in the path that can calculate an alternative path.

Once a new connection has been accepted, bandwidth enforcement is exercised at the cell level to assure that the transmitting source is within its negotiated traffic parameters. Bandwidth enforcement procedures are discussed in Section 4.7.

4.6 CALL ADMISSION CONTROL (CAC)

CAC algorithms can be classified as *nonstatistical bandwidth allocation (* or *peak bit rate allocation)* or as *statistical bandwidth allocation*. Below, we examine these two classes.

4.6.1 Classes of CAC Algorithms

Nonstatistical bandwidth allocation

Nonstatistical bandwidth allocation, otherwise known as peak bit rate allocation, is used for connections requesting a CBR service. In this case, the CAC algorithm is very simple, as the decision to accept or reject a new connection is based purely on whether its peak bit rate is less than the available bandwidth on the link. Let us consider, for example, a non-blocking switch with output buffering (see Figure 3.10), and suppose that a new connection with a peak bit rate of 1 Mbps has to be established through output link 1. Then, the new connection is accepted if the link's available capacity is more or equal to 1 Mbps.

In the case where nonstatistical allocation is used for all of the connections routed through a link, the sum of the peak bit rates of all of the existing connections is less than the link's capacity. Peak bit rate allocation can lead to a grossly underutilized link, unless the connections transmit continuously at peak bit rate.

Statistical Bandwidth Allocation

In statistical bandwidth allocation, the allocated bandwidth on the output link is less than the source peak bit rate. In the case where statistical allocation is used for all of the connections on the link, the sum of the peak bit rates of all of the connections can exceed the link's capacity. Statistical allocation makes economic sense when dealing with bursty sources, but it is difficult to implement effectively. This is due to the fact that it is not always possible to characterize accurately the traffic generated by a source and how it is modified deep in an ATM network. For instance, let us assume that a source has a maximum burst size of 100 cells. As the cells that belong to the same burst travel through the network, they get buffered in each switch. Due to multiplexing with cells from other connections and scheduling priorities, the maximum burst of 100 cells might become much larger deep in the network. Other traffic descriptors, such as the PCR and the SCR, can be similarly modified deep in the network. For instance, let us consider a source with a peak bit rate of 128 Kbps. Due to multiplexing and scheduling priorities, it is possible that several cells from this source can get batched together in the buffer of an output port of a switch. Let us assume that this output port has a speed of, say 1.544 Mbps. Then, these cells will be transmitted out back-to-back at 1.544 Mbps, which will cause the peak bit rate of the source to increase temporarily!

Another difficulty in designing a CAC algorithm for statistical allocation is due to the fact that an SVC has to be set up in real-time. Therefore, the CAC algorithm cannot be CPU intensive. This problem might not be as important when setting up PVCs. The problem of whether to accept or reject a new connection can be formulated as a queueing problem. For instance, let us consider again our non-blocking switch with output buffering. The CAC algorithm has to be applied to each output port. If we isolate an output port and its buffer from the switch, we will obtain the queueing model shown in Figure 4.9.

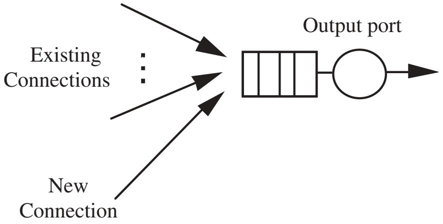


Figure 4.9 An ATM multiplexer.

This type of queueing structure is known as the *ATM multiplexer*. It represents a number of ATM sources feeding a finite-capacity queue, which is served by a server, i.e., the output port. The service time is constant and is equal to the time it takes to transmit an ATM cell.

Let us assume that the QoS, expressed in cell loss rate, of the existing connections is satisfied. The question that arises is whether the cell loss rate will still be maintained if the new connection is accepted. This can be answered by solving the ATM multiplexer queueing model with the existing connections and the new connection. However, the solution to this problem is CPU intensive and it cannot be done in real-time. In view of this, a variety of different CAC algorithms have been proposed which do not require the solution of such a queueing model.

Most of the CAC algorithms that have been proposed are based solely on the cell loss rate QoS parameter. That is, the decision to accept or reject a new connection is based on whether the switch can provide the new connection with the requested cell loss rate without affecting the cell loss rate of the existing connections. No other QoS parameters, such as peak-to-peak cell delay variation and the max CTD, are considered by these algorithms. A very popular example of this type of algorithm is the *equivalent bandwidth,* described below.

CAC algorithms based on the cell transfer delay have also been proposed. In these algorithms, the decision to accept or reject a new connection is based on a calculated absolute upper bound of the end-to-end delay of a cell. These algorithms are closely associated with specific scheduling mechanisms, such as static priorities, early deadline first, and weighted fair queueing. Given that the same scheduling algorithm runs on all of the switches in the path of a connection, it is possible to construct an upper bound of the end-to-end delay. If this is less than the requested end-to-end delay, then the new connection is accepted.

Below, we examine the equivalent bandwidth scheme and then we present the *ATM block transfer (ABT)* scheme used for bursty sources. In this scheme, bandwidth is allocated on demand and only for the duration of a burst. Finally, we present a scheme for controlling the amount of traffic in an ATM network based on *virtual path connections (VPC).*

4.6.2 Equivalent Bandwidth

Let us consider a finite capacity queue served by a server at the rate of μ. This queue can be seen as representing an output port and its buffer in a non-blocking switch with output buffering. Assume that this queue is fed by a single source, and let us calculate its equivalent bandwidth. If we set μ equal to the source's peak bit rate, then we will observe no accumulation of cells in the buffer. This is because the cells arrive as fast as they are transmitted out. If we slightly reduce the service rate μ, then we will see that cells are beginning to accumulate in the buffer. If we reduce the service rate still a little bit more, then the buffer occupancy will increase. If we keep repeating this experiment (each time slightly lowering the service rate), then we will see that the cell loss rate begins to increase. The equivalent bandwidth of the source is defined as the service rate e at which the queue is served that corresponds to a cell loss rate of ε. The equivalent bandwidth of a source falls somewhere between its average bit rate and its peak bit rate. If the source is very bursty, it is closer to its peak bit rate; otherwise, it is closer to its average bit rate. Note that the equivalent bandwidth of a source is not related the source's SCR.

There are various approximations that can be used to compute quickly the equivalent bandwidth of a source. A commonly used approximation is based on the assumption that the source is an *interrupted fluid process (IFP).* IFP is characterized by the triplet *(R, r, b),* where R is its peak bit rate; r the fraction of time the source is active, defined as the ratio of the mean length of the on period divided by the sum of the mean on and off periods; and b the mean duration of the on period. Assume that the source feeds a finite-capacity queue with a constant service time, and let K be the size of the queue expressed in bits. The service time is equal to the time it takes to transmit out a cell. Then, the equivalent bandwidth e is given by the expression:

$$e = \frac{a - K + \sqrt{(a - K)^2 + 4Kar}}{2a}R, \tag{4.1}$$

where $a = b(1 - r)R\ln(1/\varepsilon)$.

The equivalent bandwidth of a source is used in statistical bandwidth allocation in the same way that the peak bit rate is used in nonstatistical bandwidth allocation. For instance, let us consider an output link of a non-blocking switch with output buffering, and let us assume that it has a transmission speed of 25 Mbps and its associated buffer has a capacity of 200 cells. Assume that no connections are currently routed through the link. The first setup request that arrives is for a connection that requires an equivalent bandwidth of 5 Mbps. The connection is accepted and the link has now 20 Mbps available. The second setup request arrives during the time that the first connection is still up and is for a connection that requires 10 Mbps. The connection is accepted and 10 Mbps are reserved, leaving 10 Mbps free. If the next setup request is for a connection that requires more than 10 Mbps and arrives while the first two connections are still active, then the new connection is rejected.

This method of simply adding up the equivalent bandwidth requested by each connection can lead to underutilization of the link. That is, more bandwidth might be allocated for all of the connections than it is necessary. The following approximation for the equivalent bandwidth of N sources corrects the over-allocation problem:

$$c = \min\left\{\rho + \sigma\sqrt{-2\ln(\varepsilon) - \ln 2\pi}, \sum_{i=1}^{N} e_i\right\}, \tag{4.2}$$

where ρ is the average bit rate of all of the sources, e_i is the equivalent bandwidth of the ith source, calculated using the expression (4.1), and σ is the sum of the standard deviation of the bit rate of all of the sources and is equal to:

$$\sigma = \sum_{i=1}^{N} \sqrt{r_i(R_i - r_i)}.$$

When a new setup request arrives, the equivalent bandwidth for all of the existing connections, and the new one is calculated using the expression (4.2). The new connection is accepted if the resulting bandwidth c is less than the link's capacity.

Below is a numerical example that demonstrates how the maximum number of connections admitted using the above expressions for the equivalent bandwidth varies with the buffer size K, the cell loss rate ε, and the fraction of time the source is active r. Consider a link that has a transmission speed of C equal to 150 Mbps and a buffer capacity of K cells. The parameters for each connection are as follows: the peak bit rate is R; the average bit rate is ρ; and the mean duration of the on period is b. Note that the quantity r defined above is related to ρ through the expression $rR = \rho$. In the numerical examples presented below, assume that all connections are identical with traffic parameters $(R, \rho, b) = (10\,\text{Mbps}, 1\,\text{Mbps}, 310\text{ cells})$.

Note that if connections are admitted, using their peak bit rate, then a maximum of 150 Mbps/10 Mbps = 15 connections will be admitted. On the other hand, if connections are admitted, using the average bit rate, a maximum of 150 Mbps/1 Mbps = 150 connections will be admitted. These two values can be seen as the upper and lower range points on the number of connections that can be admitted using the equivalent bandwidth method.

In Figure 4.10, the maximum number of connections that can be admitted is plotted as a function of the buffer size K. The buffer size was increased from 31 cells to 31,000 cells. For each value of K, the maximum number of admitted connections was obtained using expressions (4.1) and (4.2) with the cell loss rate fixed to 10^{-6}. For small values of K, the maximum number of connections admitted by the equivalent bandwidth algorithm is constant. As K increases, the maximum number of admitted connections increases as well, and eventually flattens out.

In Figure 4.11, the maximum number of admitted connections is plotted against the cell loss rate ε. The buffer size is fixed to 1236 cells (i.e., 64 Kbytes). The maximum number of admitted connections is not very sensitive to the cell loss rate ε. In this particular example, the buffer size is large enough so that the equivalent algorithm admits a large number of connections. In general, the equivalent bandwidth algorithm becomes more sensitive to ε when the buffer size is smaller.

Finally, in Figure 4.12, the maximum number of admitted connections is plotted against r, the fraction of time that a source is active, where $r = \rho/R$. Recall that r can be used to express the burstiness of a source (see Section 4.1.1). The buffer size was fixed to 1236 cells and the cell loss rate ε to 10^{-6}. The maximum number of admitted connections depends on r. As r increases, the source becomes more bursty and requires more buffer

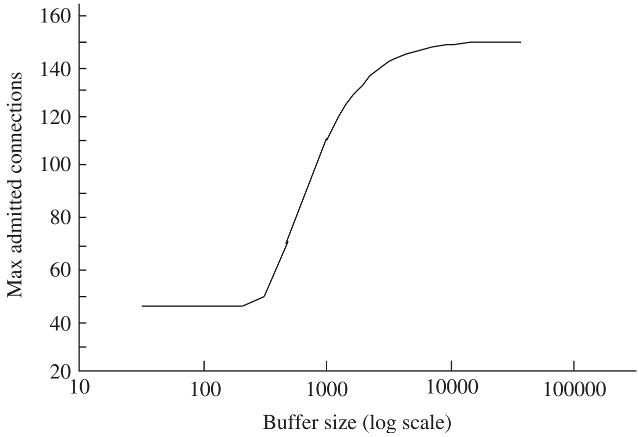


Figure 4.10 Varying the buffer size K.

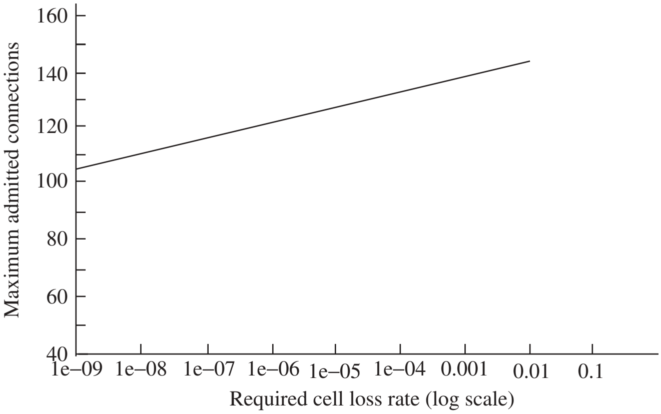


Figure 4.11 Varying the required cell loss rate ε.

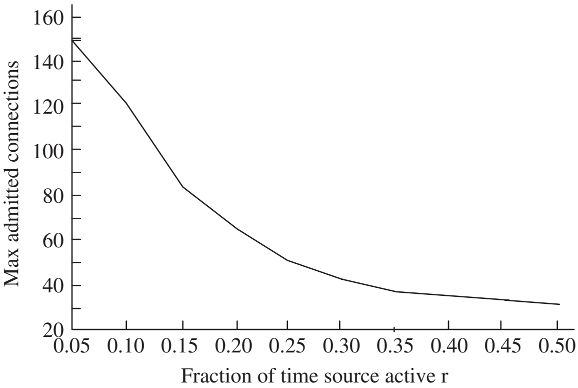


Figure 4.12 Varying r.

space in order to maintain the same cell loss rate. As a result the maximum number of admitted connections falls sharply as r tends to 0.5.

4.6.3 The ATM Block Transfer (ABT) Scheme

A number of congestion control schemes were devised for bursty sources whereby each switch allocates bandwidth on demand and only for the duration of a burst. The main idea behind these schemes is the following. At connection setup time, the path through the ATM network is selected, and each switch in the path allocates the necessary VPI/VCI labels and updates the switching table used for label swapping in the usual way. However, it does not allocate any bandwidth to this connection. When the source is ready to transmit a burst, it notifies the switches along the path. Once notified, each switch allocates the necessary bandwidth for the duration of the burst.

These congestion control schemes are known as *fast bandwidth allocation schemes*. The *ATM block transfer (ABT)* scheme is a fast bandwidth allocation scheme; it is a standardized ATM transfer capability. ABT only uses the peak bit rate and is intended for VBR sources whose peak bit rate is less than 2% of the link's capacity.

In ABT, a source requests bandwidth in incremental and decremental steps. The total requested bandwidth for each connection might vary between 0 and its peak bit rate. For a step increase, a source uses a special reservation request cell. If the requested increase is accepted by all of the switches in the path, then the source can transmit at the higher bit rate. If the step increase is denied by a switch in the path, then the step increase request is rejected. Step decreases are announced through a management cell. A step decrease is always accepted. At the cell level, the incoming cell stream of a source is shaped, so that the enforced peak bit rate corresponds to the currently accepted peak bit rate.

A *fast reservation protocol (FRP)* unit was implemented to handle the relevant management cells. This unit is located at the UNI. The protocol uses different timers to ensure its reliable operation. The end device uses a timer to ensure that its management cells, such as step increase requests, sent to its local FRP unit are not lost. When the FRP unit receives a step increase request, it forwards the request to the first switch in the path, which in turn forwards it to the next hop switch, and so on. If the request can be satisfied by all of the switches on the path, then the last switch will send an ACK to the FRP unit. The FRP unit then informs the end device that the request has been accepted, updates the policing function, and sends a validation cell to the switches in the path to confirm the reservation. If the request cannot be satisfied by a switch, the switch simply discards the request. The upstream switches, which have already reserved bandwidth, will discard the reservation if they do not receive the validation cell by the time a timer expires. This timer is set equal to the maximum round trip delay between the FRP unit and the furthermost switch. If the request is blocked, the FRP unit will retry to request the step increase after a period set by another timer. The number of attempts is limited.

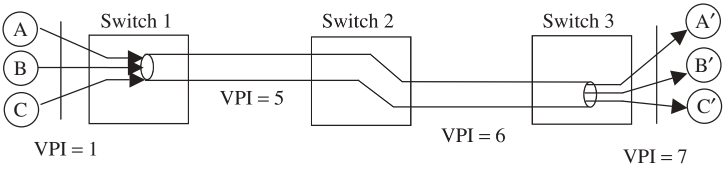
This mechanism can be used by an end device to transmit bursts. When the end device is ready to transmit a burst, it issues a step increase request with a requested bandwidth equal to its peak bit rate. If the request is granted, the end device transmits its burst, and at the end it announces a step decrease with bandwidth equal to its peak bit rate.

In a slightly different version of the ABT protocol, the end device starts transmitting its burst immediately after it issues a reservation request. The advantage of this scheme is that the end device does not have to wait until the request is granted. The burst will get lost if a switch in the path is unable to accommodate the request.

4.6.4 Virtual Path Connections

A virtual path connection can be used in an ATM network to create a dedicated connection between two switches. Within this connection, individual virtual circuit connections can be set up without the knowledge of the network.

Let us assume, for instance, that a permanent virtual path connection is established between two switches (switch 1 and switch 2). These two switches might not be adjacent, in which case, they can communicate through several other switches. A fixed amount of bandwidth is allocated to the virtual path connection. This bandwidth is reserved for this particular connection and it cannot be shared with other connections, even when it is not used entirely. An end device attached to switch 1 and wishing to communicate to an end device attached to switch 2, is allocated part of the bandwidth of the virtual path connection using nonstatistical or statistical bandwidth allocation. The connection



1	47	1	5	47	3
1	39	3	5	39	3
1	41	5	5	41	3

5	47	2	6	47	8
5	39	2	6	39	8
5	41	2	6	41	8

6	47	3	7	47	2
6	39	3	7	39	3
6	41	3	7	41	7

Figure 4.13 Label swapping in a virtual path connection.

is rejected if there is not enough bandwidth available within the virtual path connection, since the total amount of traffic carried by this virtual path connection cannot exceed its allocated bandwidth.

A virtual channel connection maintains the same VCI value through out an entire virtual path connection. That is, its VCI value has global significance. The virtual path, however, is identified by a series of VPI values, each having local significance.

An example of label swapping in a virtual path connection is given in Figure 4.13. A virtual path connection has been established between switches 1 and 3. Users A, B, and C are attached to switch 1, and via the virtual path connection, they are connected to their respective destinations A′, B′, and C′, which are attached to switch 3. Each switch is represented by a square, and its switching table is given immediately below the square. (For simplicity, we assume that the switching table in switch 1 is centralized and it contains information for all input ports.) The first three columns in the switching table give the VPI/VCI of each incoming connection and its input port. The second three columns give the new label and the destination output port of the connection. Also, the first row of each switching table is associated with the connection from A to A′, the second row is associated with the connection from B to B′, and the third row is associated with the connection from C to C′. The virtual path connection has a VPI = 1 on the UNI between users A, B, C and switch 1; a VPI = 5 on the hop from switch 1 to switch 2; a VPI = 6 on the hop from switch 2 to switch 3; and a VPI = 7 on the UNI between switch 3 and users A′, B′, and C′. The virtual channel connections from A to A′, B to B′, and C to C′ are identified by the VCIs 47, 39, and 41, respectively.

Virtual path connections provide a network operator with a useful mechanism. For instance, it can be used to provide a customer with a dedicated connection between two locations. Within this connection, the customer can set up any number of virtual circuit connections, as long as the total bandwidth allocated to the virtual path connection is not exceeded.

Virtual path connections can be combined to form a virtual network overlaid on an ATM network. Such a virtual network can be set up by a network operator in order to control the amount of traffic in the network. In addition, the network operator can set up different virtual networks for different ATM service categories.

4.7 BANDWIDTH ENFORCEMENT

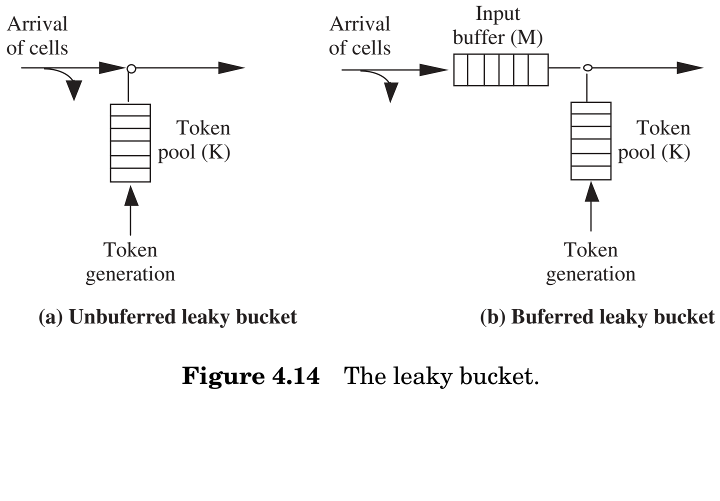
The function of bandwidth enforcement is to ensure that the traffic generated by a source conforms with the *traffic contract* that was agreed upon between the user and the network at call setup time. According to the ITU-T and the ATM Forum, the traffic contract consists of: the traffic parameters, the requested QoS parameters, and a definition of conformance. The traffic and the QoS parameters, as we have seen, depend upon the requested service category.

Testing the conformance of a source, otherwise known as policing the source, is carried out at the *user-network interface (UNI)*. It involves policing the peak cell rate and the sustained cell rate using the *generic cell rate algorithm (GCRA)*. ITU-T first standardized this algorithm for the peak cell rate. The ATM Forum adapted the same algorithm, and it also extended it for testing the conformance of the sustained cell rate. It is possible that multiple GCRAs can be used in series, such as one for the peak cell rate and another one for the sustained cell rate.

Policing each source is an important function from the point of view of a network operator, since a source exceeding its contract might affect the QoS of other existing connections. Also, depending upon the pricing scheme used by the network operator, revenue might be lost. A source might exceed its contract due to various reasons; the user equipment might malfunction, or the user might underestimate (either intentionally or unintentionally) the bandwidth requirements.

The generic cell rate algorithm is based on a popular policing mechanism known as the *leaky bucket*. The leaky bucket can be *unbuffered* or *buffered*. The unbuffered leaky bucket consists of a token pool of size K (see Figure 4.14(a)). Tokens are generated at a fixed rate. A token is lost if it is generated at a time when the token pool is full. An arriving cell takes a token from the token pool, and then enters the network. The number of tokens in the token pool is then reduced by one. A cell is considered to be a *violating cell* (or, a *noncompliant cell*), if it arrives at a time when the token pool is empty. The buffered leaky bucket is shown in Figure 4.14(b). It is the same as the unbuffered leaky bucket with the addition of an input buffer of size M, where a cell can wait if it arrives at a time when the token pool is empty. A cell is considered to be a violating cell, if it arrives at a time when the input buffer is full. Violating cells are either dropped or tagged (see Section 4.7.2).

A leaky bucket is completely defined by its parameters: K, token generation rate and M, if it is a buffered leaky bucket. The difficulty with the leaky bucket is in fixing its parameters, so that it is transparent when the source adheres to its contract, and it catches all of the violating cells when the source exceeds its contract. Given a probabilistic model of an



(a) Unbuferred leaky bucket

(b) Buferred leaky bucket

Figure 4.14 The leaky bucket.

arrival process of cells to the UNI, it is possible to fix the parameters of the leaky bucket using queueing-based models. However, it has been shown that the leaky bucket can be very ineffective in catching violating cells. Dual leaky buckets have been suggested for more efficient policing, where the first leaky bucket polices violations of the peak cell rate, and the second one polices violations of the source's burstiness. As will be seen below, GCRA does catch all violating cells, but to do that it needs an additional traffic parameter.

In addition to GCRA, a source can shape its traffic using a *traffic shaper* in order to attain desired characteristics for the stream of cells it transmits to the network. Traffic shaping involves peak cell rate reduction, burst size reduction, and reduction of cell clumping by suitably spacing out the cells in time.

4.7.1 The Generic Cell Rate Algorithm (GCRA)

Unlike the leaky bucket mechanism, GCRA is a deterministic algorithm and it does catch all of the violating cells. However, for this it requires an additional new traffic parameter known as the *cell delay variation tolerance (CDVT)*. Note that this parameter is different from the peak-to-peak cell delay variation parameter described in Section 4.2.

Let us assume that a source is transmitting at peak cell rate and it produces a cell every T units of time, where $T = 1/PCR$. Due to multiplexing with cells from other sources and with signaling and network management cells, the inter-arrival time of successive cells belonging to the same UNI source could potentially vary around T (see Figure 4.15). That is, for some cells it might be greater than T, and for others it might be less than T. In the former case, there is no penalty in arriving late. However, in the latter case, the cells will appear to the UNI that they were transmitted at a higher rate, even though they were transmitted conformally to the peak cell rate. In this case, these cells should not be penalized by the network. The cell delay variation tolerance is a parameter that permits the network to tolerate a number of cells arriving at a rate which is faster than the agreed upon peak cell rate. This parameter is not source dependent. Rather, it depends on the number of sources that use the same UNI and the access to the UNI. It is specified by a network administrator.

GCRA can be used to monitor the peak cell rate and the sustained cell rate. There are two implementations of GCRA: the *virtual scheduling algorithm* and the *continuous-state leaky bucket algorithm*. These two algorithms are equivalent to each other.

Policing the peak cell rate

In order to monitor the peak cell rate, the following two parameters are required: peak emission interval T and cell delay variation tolerance τ. T = 1/*PCR*. This is obtained from the user's declared peak cell rate and the network administrator's τ.

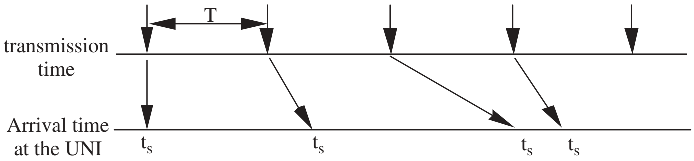


Figure 4.15 Arrival times at the UNI.

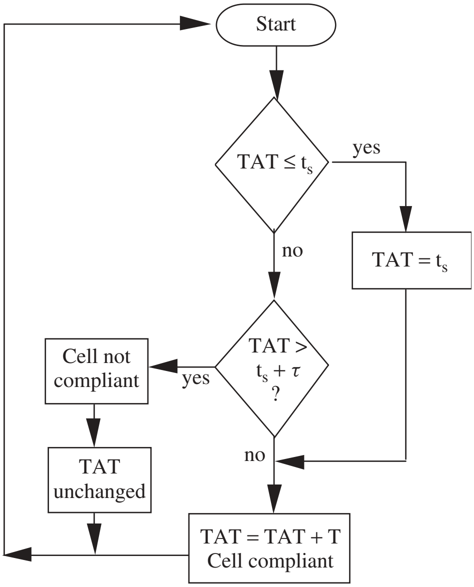


Figure 4.16 The virtual scheduling algorithm.

A flowchart of the virtual scheduling algorithm is shown in Figure 4.16. Variable *TAT* is the theoretical arrival time of a cell and t_s is the actual arrival time of a cell. At the time of arrival of the first cell, $TAT = t_s$. Each time a cell arrives, the algorithm calculates the theoretical time *TAT* of the next arrival. If the next cell arrives late (i.e., $TAT < t_s$), then the next theoretical arrival time is set to $TAT = t_s + T$. If the next arrival is early (i.e., $TAT > t_s$), then the cell is either accepted or classified as noncompliant. The decision is based on the cell delay variation tolerance τ and also on previously arrived cells that were late but they were accepted as compliant. Specifically, if $TAT < t_s + \tau$, then the cell is considered as compliant. Notice, however, that the next theoretical arrival time *TAT* is set equal to the theoretical arrival time of the current cell plus T (i.e., $TAT = TAT + T$). If the next arrival occurs before the theoretical arrival time *TAT,* then it can still be accepted if $TAT < t_s + \tau$ However, if cells continue to arrive early, then the cell delay variation will be used up and eventually a cell will be classified as noncompliant.

As an example, let us consider the case where $T = 10$, $\tau = 15$, and the actual arrival times of the first five cells are: 0, 12, 18, 20 and 25. For cell 1, we have that $t_s = TAT = 0$. The cell is accepted, and TAT is set to $TAT + 10 = 10$. For cell 2, $t_s = 12$ and since $TAT \leq t_s$, the cell is accepted and TAT is set equal to $t_s + T = 22$. Cell 3 arrives at time $t_s = 18$, and so $TAT > t_s$. Since $TAT \leq t_s + \tau$, the cell is accepted and TAT is set equal to $TAT + T = 32$. Cell 4 arrives at time $t_s = 20$, and $TAT > t_s$. Since $TAT \leq t_s + \tau$, the cell is accepted, and *TAT* is set equal to $TAT + T = 42$. Cell 5 is not as lucky as cells 3 and 4. Its arrival time is $t_s = 25$, which makes $TAT > t_s$. Since $TAT > t_s + \tau$, the cell is considered to be noncompliant.

A flowchart of the continuous state leaky bucket algorithm is shown in Figure 4.17. In this algorithm, a finite-capacity leaky bucket is implemented whose real-value content is drained out at a continuous rate of one unit of content per unit-time. Its content is

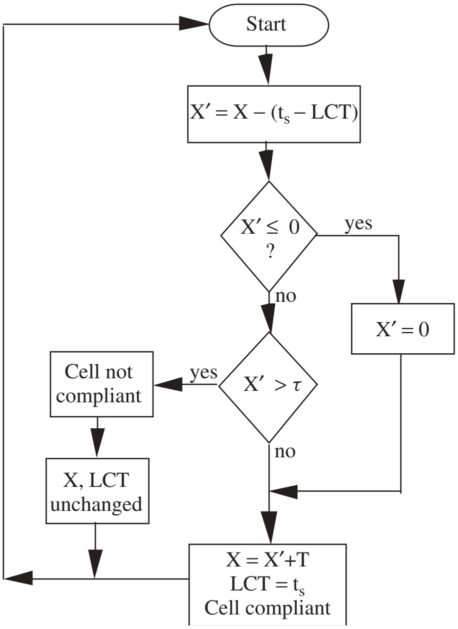


Figure 4.17 Continuous state leaky bucket algorithm.

increased by a fixed increment T each time a conforming cell arrives. The algorithm makes use of the variables X, X', and LCT. X indicates the current value of the leaky bucket; X' is an auxiliary variable; and LCT is the last compliance time (i.e., the last time a compliant cell arrived). At the arrival time of the first cell, $X = 0$ and $LCT = t_s$.

When a cell arrives, the quantity $X - (t_s - LCT)$ is calculated and saved in the auxiliary variable X'. If $X' \leq 0$, then the cell has arrived late and the cell is accepted, X is increased by T, and $LCT = t_s$. If $X' > 0$, then depending upon whether X' is less or greater than τ, the cell is accepted or it is considered as noncompliant. If $X > \tau$, the cell is classified as noncompliant and the values of X and LCT remain unchanged. If $X \leq \tau$, then the cell is accepted, X is set to $X' + T$, and $LCT = t_s$.

Now consider the same example as in the virtual scheduling algorithm: $T = 10$, $\tau = 15$, and the actual arrival times of the first five cells are: 0, 12, 18, 20 and 25. For cell 1, we have that $X = 0$ and $LCT = 0$. The cell is accepted and X is set to 10 and LCT to 0. For cell 2, we have $X' = -2$. The cell is accepted and X is set to 10 and LCT to 12. Cell 3 arrives at time 18, which gives a value for X' equal to 4. Since $X' < \tau$, the cell is accepted and X is set to $X' + T = 14$ and LCT to 18. Cell 4 arrives at time 20, and we have $X' = 12$. Since $X' < \tau$, the cell is accepted and X is set to $X' + T = 22$ and LCT to 20. Cell 5 arrives at time 25, and so that $X' = 17$. Since $X' > \tau$, the cell is classified as noncompliant, and X and LCT remain unchanged.

Policing the sustained cell rate

The sustained cell rate of a source is policed by either GCRA algorithm. As we saw above, GCRA uses the parameters T and τ in order to police the peak cell rate of a source. For policing the sustained cell rate of a source, it uses the parameters T_s and τ_s. T_s is the emission interval when the source transmits at its sustained cell rate, and is equal to $1/SCR$. τ_s is known as the *burst tolerance (BT)* and is calculated from the *maximum burst size (MBS)* provided by the source using the expression:

$$\tau_s = (MBS - 1)(T_s - T).$$

If the inter-arrival time of cells is equal to or greater than T_s, then the cells are compliant. However, if cells are transmitted at peak cell rate, then some might arrive every T units of time, where $T < T_s$. Since these cells arrive every T units of time, they are in essence noncompliant as far as GCRA is concerned. How many such cells should GCRA tolerate, before it starts classifying them as noncompliant? Obviously, the maximum number of cells that can arrive every T units of time is equal to the source's MBS minus the first cell that initiates the burst. That is, we expect a maximum of $(MBS - 1)$ cells to arrive $(T_s - T)$ units of time faster. This gives a total time of $(MBS - 1)$ $(T_s - T)$, which is the burst tolerance τ_s. In conformance testing, τ_s is set equal to:

$$\tau_s = (MBS - 1)(T_s - T) + CDVT.$$

4.7.2 Packet Discard Schemes

As we saw in the previous section, GCRA will either accept a cell or classify it as noncompliant. The question, therefore, that arises is what to do with noncompliant cells. The simplest scheme is to just drop them. A more popular mechanism, known as *violation tagging*, attempts to carry the noncompliant cells if there is slack capacity in the network. The violating cells are tagged at the UNI and then they are allowed to enter the network. If congestion arises inside the network the tagged cells are dropped. Tagging of a cell is done using the *cell loss priority (CLP)* bit in the cell's header. If the cell is untagged, then its $CLP = 0$. When a cell is tagged, its $CLP = 1$.

Violation tagging introduces two types of cells: the untagged cell and the tagged cell. A simple way to handle tagged cells is through a priority mechanism, such as the *push-out scheme* and the *threshold scheme*. In the push-out scheme, both untagged and tagged cells are freely admitted into a buffer as long as the buffer is not full. If a tagged cell arrives during the time that the buffer is full, the cell is lost. If an untagged cell arrives during the time that the buffer is full, the cell will take the space of the last arrived tagged cell. The untagged cell will get lost if all of the cells in the buffer are untagged. In the threshold scheme, both untagged and tagged cells are admitted as long as the total number of cells is below a threshold. Over the threshold, only untagged cells are admitted, and the tagged cells are rejected. The push-out priority scheme is more efficient than the threshold priority scheme, but the latter is preferable because it is simpler to implement. Other priority mechanisms have also been proposed, including dropping from the front. This mechanism is similar to the threshold mechanism, only cells are dropped from the front. That is, when a tagged cell is ready to begin its service, the total number of cells in the buffer is compared against the threshold. If it is below, service begins, else the cell is dropped.

A discarded cell might be part of a user packet, such as a TCP packet. In this case, the receiving TCP will detect that the packet is corrupted and it will request the sending TCP to retransmit it. In view of this, when discarding a cell we can save bandwidth by discarding the subsequent cells that belong to the same user packet since the entire packet will have to be retransmitted anyway. For applications using AAL 5, it is possible to identify the beginning and the end of each user packet, and consequently drop the subsequent cells that belong to the same packet. There are two such discard mechanisms: *partial packet discard (PPD)* and *early packet discard (EPD).* Partial packet discard can be applied when the discarded cell is not the first cell of an AAL 5 frame. In this case, all subsequent cells belonging to the same AAL 5 frame are discarded except the last cell. This cell has to be kept so that the destination can determine the end of the AAL 5 frame. Early packet discard can be applied when the discarded cell is the first cell of an AAL 5 frame. In this case, all cells belonging to the same frame, including the last one, are discarded.

4.8 REACTIVE CONGESTION CONTROL

Reactive congestion control is based on a different philosophy to the one used in preventive congestion control. In preventive congestion control we attempt to prevent congestion from occurring. This is done by first reserving bandwidth for a connection on each switch along the connection's path, and subsequently policing the amount of traffic transmitted on the connection. In reactive congestion control, at least in its ideal form, we let sources transmit without bandwidth reservation and policing, and we take action only when congestion occurs. The network is continuously monitored for congestion. If congestion begins to build up, a feedback message is sent back to each source requesting them to slow down or even stop. Subsequent feedback messages permit the sources to increase their transmission rates. Typically, congestion is measured by the occupancy level of critical buffers within an ATM switch, such as the output port buffers in a non-blocking switch with output buffering.

The *available bit rate (ABR)* service, described below, is the only standardized ATM service category that uses a reactive congestion control scheme.

4.8.1 The Available Bit Rate (ABR) Service

This is a feedback-based mechanism whereby the sending end device is allowed to transmit more during the time that there is a slack in the network. At connection setup time, the sending end device requests a *minimum cell rate (MCR).* It also specifies a *maximum cell rate*, which is its PCR. The network accepts the new connection if it can satisfy the requested MCR (note that the MCR might be 0). If the network has a slack capacity, then the source transmission rate might exceed the requested MCR. When congestion begins to build up in the network, the sending end device is requested to decrease its transmission rate. However, its transmission rate will never drop below its MCR. The ABR service is not intended to support real-time applications.

It is expected that the sending end device is capable of increasing or decreasing its transmission rate according to the feedback messages it receives from the network. Also, it is expected that the sending end device that conforms to the feedback messages received by the network, will experience a low cell loss rate and it will obtain a fair share of the available bandwidth within the network.

The control mechanism through which the network can inform the source to change its transmission rate is implemented using *resource management (RM)* cells. These are ATM

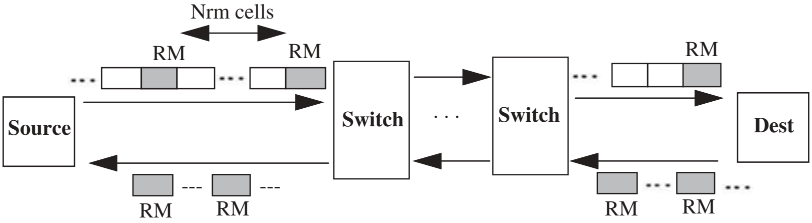


Figure 4.18 The ABR mechanism.

cells whose *payload type indicator (PTI)* is set to 110 (see Table 3.2). The transmitting end device generates *forward RM cells,* which travel through the network to the receiver following the same path as its data cells (see Figure 4.18). The receiver turns around these RM cells and transmits them back to the sending end device as *backward RM cells.* These backward RM cells follow the opposite path to the sending end device. (Recall that point-to-point connections are bidirectional.) ATM switches along the path of the connection might insert feedback control information in the RM cells; the sending end device uses this information for managing (i.e. increasing or decreasing) its transmission rate. Thus, a closed control loop is formed between the sending end device and its destination end device. This closed loop is used to regulate the transmission rate of the sending end device. A similar loop can be set up to regulate the transmission rate of the destination end device.

Because a link might be used as temporary storage for a large number of cells, feedback messages might be ineffective when dealing with a link that has a long propagation delay. For example, consider a link that is 125 miles long connecting an end device to a switch, and let us assume that the end device transmits at 622 Mbps. This transmission speed translates to about 1462 cells per msec. Since light propagates through a fiber link at approximately 125 miles per msec, a maximum of 1462 cells can be propagated along the link at any given time. Now assume that at time t, the switch sends a message to the end device requesting it to stop transmitting. By the time the end device receives the message, the switch is likely to receive a maximum of 2×1462 cells. Of these cells, a maximum of 1462 cells can be already in flight at time t, and another maximum of 1462 cells can be transmitted by the time the end device receives the message. In order to account for large propagation delays, manufacturers have introduced large buffers in their switch architectures. In addition, several feedback loops can be set up (see Figure 4.19), to try reducing the length of the control loop.

The ABR service does not include a formal conformance definition. However, verification that the source complies can be done using a dynamic GCRA, where the monitored transmission rate is modified based on the receipt of backwards RM cells.

RM cell structure

The RM cell's payload contains a number of different fields. Below, we describe some of these fields.

- *Message type field*: This is a 1-byte field and it contains the following 1-bit subfields:
  1. *DIR*: This bit indicates the direction of the RM cell (i.e., whether it is a forward or a backward RM-cell).

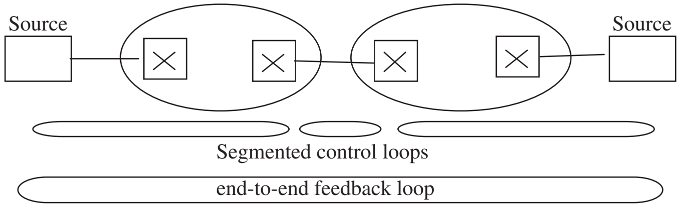


Figure 4.19 Feedback loops.

2. *BN*: This bit indicates whether the RM cell is a *backward explicit congestion notification (BECN)* cell. As will be seen later on, an ATM switch or the destination end device might independently generate a backward RM cell in order to notify the sending end device, instead of having to wait for an RM cell generated by the sending end device to come by. This RM cell has its BN bit set to 1. RM cells generated by the source have their BN field set to 0.
3. *CI*: This congestion indication bit is used to by an ATM switch or the destination end device, to indicate to the sending end device that congestion has occurred in the network.
4. *NI*: A no increase indicator, used to prevent the sending end device from increasing its *allowed cell rate (ACR),* which is its current transmission rate.

- *Explicit rate (ER)*: This is a 2-byte field. It is used to carry the explicit rate that is calculated by an ATM switch along the path. It is also used to limit the sending end device's transmission rate. If another ATM switch calculates an ER that is lower than the one indicated in the ER field of the RM cell, then it can reduce the ER variable.
- *Current cell rate (CCR)*: A 2-byte field used by the sending end device to indicate its ACR (i.e. its current transmission rate).
- *Minimum cell rate (MCR)*: This the minimum cell rate that the connection has requested and the network has agreed to guarantee.

The ABR mechanism

The source sends an RM cell every Nrm-1 data cells. The defaulted value for Nrm is 32. The RM cells and data cells might traverse a number of switches before they reach their destination end device. The destination turns around the RM cells, making them into backward RM cells, and transmits them back to the sending end device. Each switch writes information about its congestion status onto the backward RM cells, which eventually reach the sending end device. The feedback information send to the source depends on the mode of the ABR scheme. There are two modes: the *binary mode* and the *explicit rate mode*.

In the binary mode, the switch marks the EFCN bit in the header of the data cells to indicate pending congestion. (Recall that the EFCN bit is one of the three bits defined in the payload type indicator of the cell header.) The destination translates the EFCN information into bits (such as the CI or NI), which are marked in the corresponding backward RM cell. The source receives the backward RM cell and then determines what to do with the transmission rate: increase it, decrease it, or leave it as-is. This mode is used to provide backward compatibility with ATM switches that conformed to earlier standards.

In the explicit rate mode, a switch computes a local fair share for the connection and marks the rate at which the source is allowed to transmit in the ER field of the backward RM cell. The switch does that only if the bandwidth it can offer to the connection is lower than what it is already marked in the backwards RM cell. The source, upon receipt of the backward RM cell, extracts the ER field and sets its transmission rate to the ER value. When detecting congestion, a switch can generate a backwards RM cell in order to convey the congestion status, without having to wait for a backwards RM cell to arrive.

Source behavior

The source is responsible for inserting an RM cell every *Nrm*-1 data cells. These RM cells are part of the source's *allowed cell rate (ACR).* If the source does not have enough data cells to send, an RM cell is generated after a timer has expired and *Mrm* data cells have been transmitted. *Mrm* is fixed to 2. The data cells are sent with EFCN = 0.

The source adjusts its ACR according to the information received in an RM cell. ACR is greater than or equal to MCR and less than or equal to PCR. The ACR is adjusted as follows:

a. If CI = 1, then the ACR is reduced by at least $ACR \times RDF$, where RDF is a prespecified *rate decrease factor*. If the reduction results to a value below the MCR, then the ACR is set equal to the MCR.
b. If the backward RM cell has both CI = 0 and NI = 0, then the ACR can be increased by no more than RIF × PCR, where RIF is a prespecified *rate increase facto*r. The resulting ACR should not exceed the source's PCR.
c. If the backward RM cell has NI = 1, then the ACR is not increased.

After ACR has been adjusted as above, it is set to at most the minimum of ACR as computed above and to the ER field, but no lower than MCR.

Destination behavior

When a data cell is received, its EFCN is saved in the EFCN status of the connection. Upon receiving a forward RM cell, the destination turns around the cell and transmits it back to the source. The DIR bit is changed from forward to backward; the BN = 0; and the fields CCR, MCR, ER, CI, and NI in the RM cell remain unchanged, except in the following cases:

a. If the saved EFCN status of the connection is set, then the destination sets CI = 1 in the RM cell, and resets the EFCN state.
b. If the destination is experiencing internal congestion, then it can reduce the ER to whatever rate it can support and set it to either CI = 1 or NI = 1.

The destination can also generate a new backward RM cell, with CI = 1 or NI = 1, DIR = 1, and BN = 1. This permits the destination to send feedback information to the source without having to wait for a source-generated RM cell to come by. The rate of these backwards RM cells is limited to 10 cells/sec.

Switch behavior

At least one of the following methods is implemented in a switch:

a. *EFCN marking*: The switch can set the EFCN bit in the header of the data cells.
b. *Relative rate marking*: The switch can set CI = 1 or NI = 1 in forward and/or backward RM cells.
c. *Explicit rate marking*: The switch can reduce the ER field of forward and/or backward RM-cells.

The first two marking methods are part of the binary mode, whereas the third one is for the explicit rate mode. The term *binary* is used because the switch provides information of the type: congestion/no congestion.

To send feedback information to the source without having to wait for a source-generated RM cell, a switch can generate backwards RM cells. The rate of these backwards RM cells is limited to 10 cells/sec. Its fields are marked as follows: CI = 1 or NI = 1, BN = 1, DIR = 1.

A switch can also segment the ABR closed loop using a virtual source and destination. This can be useful in cases where the loop between the source and destination involves many hops, or long haul links with a large propagation delay. In such cases, the time it takes for the RM cells to return to the source can be significant. This might impact the time required for the source to react to an RM cell.

The calculation of the ER has to be done in such a way so that the available bandwidth in the switch has to be shared fairly among all of the competing ABR connections. A number of different algorithms for calculating the ER have been proposed in the ATM Forum standards.

In the binary mode operation, the switch has to decide when to raise the alarm that congestion is pending. If we consider a non-blocking switch with output buffering, then if congestion occurs at an output port, the number of cells in its associated output buffer will increase dramatically. Typically, there are two thresholds associated with this buffer: a low threshold (T_{low}) and a high threshold (T_{high}). When the number of cells goes over T_{high}, the switch can start marking the EFCN bit of the data cells or turn on the CI or NI bit in a forward or backward RM cell. As the sources begin to react to the feedback information, the number of cells in the buffer will go down below T_{high}. However, the switch continues marking until the number of cells in the buffer goes below T_{low}. At that moment, the switch stops the binary marking.

Simulation studies have shown that in a binary feedback scheme as the one presented above, some connections might receive more than their fair share of the bandwidth. Let us consider the case where source A is very close to a switch, and source B very far away. Then A will react to the feedback information from the switch much faster than B. For instance, if switch congestion occurs, then A will decrease its transmission rate more quickly than B. Similarly, when the congestion is lifted, A will increase its transmission faster than B. As a result, source A can put through more traffic than B.

PROBLEMS

1. Consider a 64-Kbps voice connection transmitted at constant bit rate (silence periods are also transmitted).
   a. What is its PCR?

b. What is its SCR?
c. What is its average cell rate?

2. This is the continuation of Problem 9, Chapter 3. On the average, a voice source is active (talkspurt) for 400 msec and silent for 600 msec. Let us assume that a voice call is transported over an ATM network via AAL 2. The voice is coded to 32 Kbps and silent periods are suppressed. Assume that the SSCS has a timer set to 5 msec. That is, each time the timer expires, it sends whatever data it has gathered to CPS as a CPS-packet. In Chapter 3, Problem 9, you were asked to calculate the length of each CPS-packet, and the number of CPS-packets produced in each active period. Using this information, answer the following two questions:
   a. What is the peak and average transmission bit rate of the voice source including the CPS-packet overhead?
   b. What is the peak and average transmission rate of the voice source including the overheads due to the CPS-packet, the CPS-PDU and the ATM cell, assuming one CPS-packet per CPS-PDU?

3. Consider an on/off source where the off period is constant and equal to 0.5 msec. The MBS of the source is 24 cells. During the on period, the source transmits at the rate of 20 Mbps.
   a. What is its PCR?
   b. What is the maximum length of the on period in msec?
   c. Assuming a 1-msec period, calculate its SCR.

4. Explain why the end-to-end cell transfer delay consists of a fixed part and a variable part. What is the fixed part equal to?

5. Explain why jitter is important to a delay-sensitive applications.

6. Consider an on/off source with a peak bit rate = 500 Kbps, and an average on period = 100 msec. The source will be transmitted over the output port of a non-blocking switch, which has a buffer $K = 500$ cells. Plot the average bit rate and the equivalent bandwidth of the source as a function of r (i.e., the fraction of time that the source is active). You should observe that the equivalent bandwidth tends to its peak bit rate when the source is bursty and to its average bit rate when the source is regular (i.e. not bursty). (Remember to express K in bits!)

7. Consider the virtual scheduling algorithm for policing the PCR. Assume that T (1/PCR) = 40 units of time and the CDVT = 60 units of time. The arrival times are: 0, 45, 90, 120, 125, 132, 140, and 220. Which of these arrivals will get tagged?

8. Repeat Problem 7 using the continuous state leaky bucket algorithm.

APPENDIX: SIMULATION PROJECT: ATM TRAFFIC CHARACTERIZATION OF AN MPEG VIDEO SOURCE

An MPEG video encoder generates frames which are transmitted over an ATM link. Write a program to simulate the traffic generated by the MPEG video encoder with a view to characterizing the resulting ATM traffic.

Problem Description

An MPEG video encoder generates frames whose size is correlated and can be predicted using an *autoregressive integrated moving average (ARIMA)* model. As described in Section 4.1.3, for the group of pictures I B B P B B P B B P B B, the following ARIMA model can be used to predict the size $S(i)$ of the *ith* frame in bits: $S(i) = S(i-12) + e(i) - 0.69748e(i-3)$, where $e(i)$ is white noise and it follows the distribution $N(0, \sigma^2)$, with $\sigma^2 = 4849.5$ bits.

An MPEG (I, B, or P) frame is generated every 30 msec. The information generated for each frame is transmitted over an ATM link using AAL1 unstructured PDUs. Assume that it takes zero time to pack the bits of a frame into AAL1 PDUs and subsequently into ATM cells. Also, assume that the ATM cells generated by a single frame are transmitted out back-to-back over a slotted link, with a slot equal to 3 μsec.

Assume that you are observing the ATM cells transmitted out on this slotted link. Due to the nature of the application, you will see that the ATM traffic behaves like an on/off model. Measure the following parameters of this ATM traffic: average cell rate, sustained cell rate with $T = 900$ msec, MBS, average off period, and the squared coefficient of variation of the inter-arrival c^2.

Simulation Structure

The simulation program can be organized into three parts. In the first part, you generate the size of the next frame, and in the second part you collect statistics on the ATM cells generated by this frame. Repeat these two parts until you have generated 5000 frames. Then go to Part 3 to calculate and print the final statistics.

Part 1

Use the above auto-regressive model to generate the size in bits of the next frame. For this, you will need to keep the size of the previous twelve frames. Start generating from frame 13 using the following initial values, expressed in bits: $S(1) = 999700$, $S(2) = 97600$, $S(3) = 395500$, $S(4) = 516460$, $S(5) = 89840$, $S(6) = 295500$, $S(7) = 696820$, $S(8) = 77900$, $S(9) = 89840$, $S(10) = 619220$, $S(11) = 97300$, $S(12) = 95360$.

Use the following procedure to generate Normally distributed variates for $e(i)$:

1. Draw two random numbers r_1 and r_2, $0 < r_1, r_2 < 1$.
   Calculate $v = 2r_1 - 1$, $u = 2r_2 - 1$, and $w = v^2 + u^2$.
2. If $w > 1$, then repeat Step 1; otherwise, $x = v\ [(-2\log_e w)/w]^{1/2}$.
3. Set $e\ (i) = 69.638x$.

Part 2

A new frame is generated every 30 msec. Having generated the frame size, calculate how many ATM cells are required to carry this frame. Let X be the number of required ATM cells. These ATM cells are generated instantaneously, and are transmitted out back-to-back, with a transmission time equal to 3 μsec per cell. Calculate how many slots will be idle before the next frame arrives. Let the number of idle slots be Y. Update the following variables:

frame_counter = frame_counter + 1
total_simulation_time = total_simulation_time + 30
total_cells_arrived = total_cells_arrived + X
MBS = max$\{MBS, X\}$
on_period = on_period + X
off_period = off_period + Y

For the sustained rate, set up a loop to calculate the total number of cells S arrived in thirty successive frames (i.e. in 900 msec). When thirty frames have been generated, compare this value against S, and save the largest of the two back in S. (Initially, set $S = 0$).

To calculate c^2, you will need to keep all of the inter-arrival times of the ATM cells. Between two cells that are transmitted back-to-back, the inter-arrival time is 1; between the last cell of a frame and the first cell of the next frame, it is $Y + 1$. Maintain two variables: *Sum* and *SqSum*. For each inter-arrival time t, do the following:

$Sum = Sum + t$
$SumSq = SumSq + t^2$

If frame_counter < 5000, repeat Parts 1 and 2 to continue to generate frames. Otherwise, go to Part 3.

Part 3

Calculate and print out the required ATM traffic parameters:

Average cell rate = total_cells_arrived/total_simulation_time
$SCR = S/900$
MBS
average on period = on_period/frame_counter
average off period = off_period/frame_counter
$c^2 = Var/MeanSq$, where:
$Var = [SumSq - (Sum^2/\text{total\_cells\_arrived})]/(\text{total\_cells\_arrived} - 1)$
$MeanSq = (Sum/\text{total\_cells\_arived})^2$.

5

Signaling in ATM Networks

Recall that in ATM networks there are two types of connections: *permanent virtual connections (PVC)* and *switched virtual connections (SVC)*. PVCs are established off-line using network management procedures, whereas SVCs are established dynamically in real-time using signaling procedures. To establish an SVC, two separate signaling protocols are used; one is used exclusively over the UNI and another is used exclusively within the ATM network. As an example, let us consider the case where user A wants to establish a point-to-point connection to a destination user B over a private ATM network. user A sends a request for the establishment of the connection to its ingress ATM switch using the signaling protocol Q.2931. The ATM switch then establishes a connection to the egress switch that serves B using the *private network node interface (PNNI)* protocol, and finally, the egress switch uses Q.2931 to interact with B over their UNI.

This chapter deals with the signaling protocol Q.2931. This signaling protocol runs on top of a specialized AAL, known as the *signaling AAL (SAAL)*. A special sublayer of this AAL is the *service-specific connection oriented protocol (SSCOP)*. The main features of SAAL and SSCOP are first discussed, and then the various ATM addressing schemes are presented. Then, the signaling messages and procedures used by Q.2931 are described.

5.1 INTRODUCTION

An ATM virtual circuit connection can be either a *point-to-point connection* or *point-to-multipoint connection. A* point-to-point connection is *bidirectional*: it is composed of two unidirectional connections, one in each direction. Both connections are established simultaneously over the same physical route. Bandwidth requirements and QoS can be specified separately for each direction. A point-to-multipoint connection is a unidirectional connection, and it consists of an ATM end device, known as the *root*, which transmits information to a number of other ATM end devices, known as the *leaves*.

Point-to-point SVC connections are established over the private UNI using the ITU-T signaling protocol Q.2931. Point-to-multipoint SVC connections are established over the UNI using the ITU-T signaling protocol Q.2971 in conjunction with Q.2931.

The establishment of an ATM SVC connection across one or more private ATM networks is done using PNNI. PNNI provides the interface between two ATM switches that either belong to the same private ATM network or to two different private ATM networks (see Figure 5.1). The abbreviation PNNI can be interpreted as either the *private network node interface* or the *private network-network interface*, reflecting these two possible uses.

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

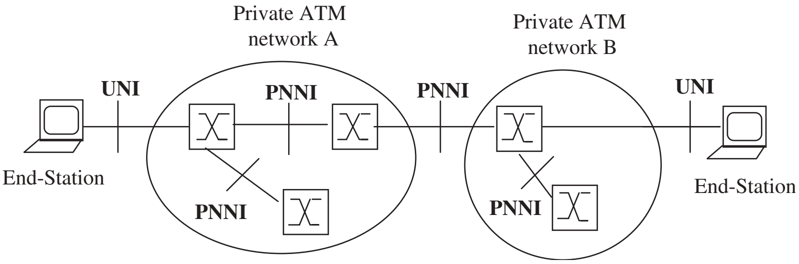


Figure 5.1 The private network-network interface (PNNI).

The PNNI protocol consists of two components: the *PNNI signaling protocol* and the *PNNI routing protocol*. The PNNI signaling protocol is used to dynamically establish, maintain, and clear ATM connections at the private network-network interface and at the private network node interface. The PNNI routing protocol is used to distribute network topology and reachability information between switches and clusters of switches. This information is used to compute a path from the ingress switch of the source end device to the egress switch of the destination end device over which signaling messages are transferred. The same path is used to set up a connection along which the data will flow. PNNI was designed to scale across all sizes of ATM networks, from a small campus network with a handful of switches to large world-wide ATM networks. Scalability is achieved by constructing a multi-level routing hierarchy based on the 20-byte ATM NSAP addresses.

The Q.2931 signaling protocol, described in this chapter, is used exclusively over the UNI between a user and its ingress switch. It was standardized by ITU-T and it was subsequently modified by the ATM Forum. Q.2931 is based on Q.931, a protocol that is part of the *Digital Subscriber Signaling System No.1 (DSS1)*, which was defined by ITU-T for signaling between a *Narrowband ISDN (N-ISDN)* user and its local exchange (see Section 12.1.5).

5.2 THE SIGNALING PROTOCOL STACK

The signaling protocol stack is shown in Figure 5.2. It is analogous to the ATM protocol stack shown in Figure 3.5 (see also Section 3.3). The ATM protocol stack shows the protocol layers used for the transfer of data, whereas the stack in Figure 5.2 shows the

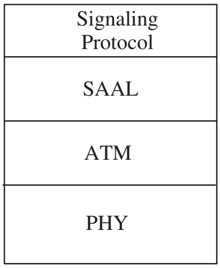


Figure 5.2 The signaling protocol stack.

protocol layers used for setting-up an SVC. As we can see, a signaling protocol (such as Q.2931, Q.2971, and PNNI) is an application that runs on top of SAAL. Below SAAL, we have the familiar ATM layer and the physical layer. The signaling protocol stack is often referred to as the *control plane*, as opposed to the *data plane* that refers to the ATM protocol stack.

5.3 THE SIGNALING ATM ADAPTATION LAYER (SAAL)

SAAL consists of an SSCS, which is composed of the *service-specific coordination function (SSCF)* and the *service-specific connection oriented protocol (SSCOP)*. (See Figure 5.3.) The common part of the convergence sublayer is AAL 5 (see Section 3.7.3). The SSCF maps the services required by the signaling protocol to the services provided by SSCOP. The SSCOP is a protocol designed to provide a reliable connection over the UNI to its peer SSCOP. This connection is used by a signaling protocol to exchange signaling messages with its peer protocol over the UNI in a reliable manner.

Recall that in the data plane there is neither error control nor flow control between an end device and its ingress switch, or between any two adjacent ATM switches. In the signaling plane, however, it is not desirable to lose or to deliver erroneously signaling messages, as this might significantly impact network performance. In view of this, the SSCOP was developed to assure a reliable transfer of messages between peer signaling protocols.

5.3.1 The SSCOP

In the OSI model, the data link layer provides error/loss recovery and flow control on each hop using the HDLC go-back-n and selective−reject ARQ scheme. In the case where the transmission link is reliable, the selective-reject ARQ scheme turns out to be more effective than the go-back-n scheme since only the lost/erroneous PDU is retransmitted. These ARQ schemes were devised for networks where the transmission speeds and propagation

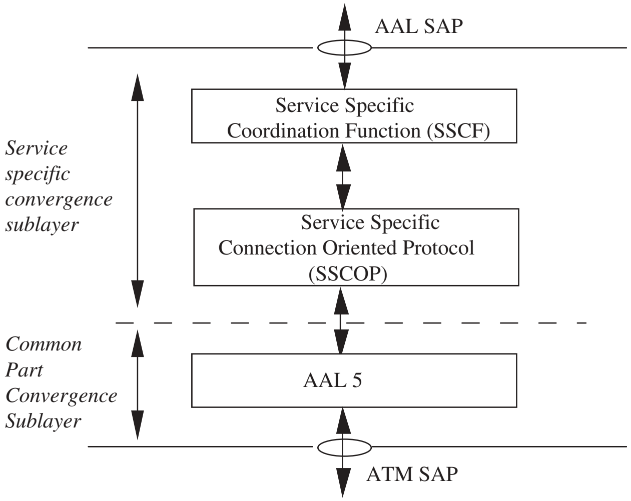

AAL SAP
Service Specific Coordination Function (SSCF)
Service specific convergence sublayer
Service Specific Connection Oriented Protocol (SSCOP)
Common Part Convergence Sublayer
AAL 5
ATM SAP

Figure 5.3 The signaling AAL (SAAL).

delays were much slower in comparison with modern networks. High speed networks, such as ATM networks, are characterized by high bandwidth-delay product. That is, the product of the transmission speed of an ATM link times the round trip propagation delay is high. Network designers became aware that the traditional selective-reject scheme was inefficient for networks with high bandwidth-delay product, since it only retransmitted one lost/erroneous PDU at a time. Several new lightweight protocols with a novel selection transmission scheme were proposed and were shown to have an improved throughput over traditional protocols. In these protocols, there is a periodic exchange of the full state of the communicating entities, decisions about received packets are communicated to the transmitter for entire blocks of SD PDUs, and the processing of the protocol is parallelized.

SSCOP was standardized in 1994; it incorporates many design principles of these lightweight protocols. It provides most of the services provided by LAP-D in ISDN and *message transfer part (MTP)* level 2 in *signaling system no. 7 (SS7).*

The SSCOP PDUs

SSCOP's main function is to establish and release a connection to a peer SSCOP, and to maintain an assured transfer of data over the connection. This is done using the *protocol data units (PDU)* described in Table 5.1. The PDUs are grouped together according to their function. The first column gives the various functions, the second column gives the name of each SSCOP PDU and its abbreviation, and the last column provides a brief description of each PDU.

The BEGIN (BGN), BEGIN ACKNOWLEDGMENT (BGAK), and BEGIN REJECT (BGREJ) PDUs are used to establish an SSCOP connection between two peer entities. The BGN PDU is used to request the establishment of an SSCOP connection between two peer entities. BGN PDU requests the clearing of the peer SSCOP's transmitter and receiver buffers, and the initialization of the peer SSCOP's transmitter and receiver state variables. The BGAK PDU is used to acknowledge the acceptance of a connection request from the peer, and the BGREJ PDU is used to reject the connection request from the peer SSCOP.

The END PDU is used to release an SSCOP connection between two peers, and the ENDAK PDUs is used to confirm the release of an SSCOP connection.

The RESYNCHRONIZATION (RS) PDU is used to request the resynchronization of buffers and state variables, and the RESYNCHRONIZATION ACKNOWLEDGMENT (RSAK) is used to acknowledge the resynchronization request. The ERROR RECOVERY (ER) PDU is used to recover from errors and the ERROR RECOVERY ACKNOWLEDGMENT (ERAK) to acknowledge an error recovery request.

The assured transfer of data is implemented using the PDUs: SEQUENCED DATA (SD), STATUS REQUEST (POLL), SOLICITED STATUS RESPONSE (STAT), and UNSOLICITED STATUS RESPONSE (USTAT). The assured data transfer scheme is described below.

The UNNUMBERED USER DATA (UD) PDU is used for non-assured data transfer between two SSCOP users. The UD PDU does not carry a sequence number as the SD PDU, and consequently it cannot be recovered by retransmission in the case where it gets lost or it arrives erroneously. UD PDUs are used by SSCOP to send information to its peer without affecting the SSCOP states or variables.

Table 5.1 The SSCOP SD PDUs.

Function	SSCOP PDU name	Description
Establishment	BEGIN (BGN)	Used to request the establishment an SSCOP connection.
	BEGIN ACKNOWLEDGEMENT (BGAK)	Used to acknowledge acceptance of an SSCOP connection request by the peer SSCOP.
	BEGIN REJECT (BGREJ)	Used to reject the establishment of a connection requested by the peer SSCOP.
Release	END	Used to release an SSCOP connection between two peer SSCOP entities.
	END ACKNOWLEDGEMENT (ENDAK)	Used to confirm the release of an SSCOP connection requested by the peer SSCOP.
Resynchronize	RESYNCHRONIZATION (RS)	Used to resynchronize buffers and the data transfer state variables.
	RESYNCHRONIZE ACKNOWLEDGEMENT	Used to acknowledge the acceptance of a resynchronization request.
Recovery	ERROR RECOVERY (ER)	Recovery command.
	ERROR RECOVERY ACK (ERAK)	Recovery acknowledgment.
Assured transfer	SEQUENCED DATA (SD)	Used to transfer user data.
	STATUS REQUEST (POLL)	Used by transmitting SSCOP to request status information from the receiving SSCOP.
	SOLICITED STATUS RESPONSE (STAT)	Used to respond to a POLL. It contains the sequence numbers of outstanding SD PDUs and credit information for the sliding window.
	UNSOLICITED STATUS RESPONSE (USTAT)	Similar to STAT message, but issued by the transmitter when a missing or erroneous SD PDU is identified.
Unacknowledged data transfer	UNNUMBERED USER DATA (UD)	Used to transfer data in an non-assured manner.
Management data transfer	UNNUMBERED MANAGEMENT DATA (MD)	Used to transfer management data in a non-assured manner.

Finally the UNNUMBERED MANAGEMENT DATA (MD) PDU is used to transmit management information. It does not include a sequence number and it can be lost without notification.

The assured data transfer scheme

This scheme provides recovery of lost or erroneously received data by retransmission. It also provides flow control using an adjustable sliding window. The data is carried in SEQUENCED DATA (SD) PDUs. An SD PDU is sequentially numbered and it can carry a variable length payload of up to 65,535 bytes. Sequence numbers range between 0 and $n - 1$ (where $n = 2^{24} - 1$).

The PDUs STATUS REQUEST (POLL), SOLICITED STATUS RESPONSE (STAT), and UNSOLICITED STATUS RESPONSE (USTAT) are used to implement a retransmission scheme for erroneously received or lost SD PDUs. Specifically, the transmitter periodically sends a POLL PDU to request the status of the receiver. This POLL is either triggered when the timer, Timer_POLL, expires or after a certain number of SD PDUs has been sent. The POLL contains the sequence number of the next SD PDU to be sent by the transmitter. It is sequentially numbered with a poll sequence number which essentially functions as a time-stamp. Poll sequence numbers are between 0 and $n - 1$ (where $n = 2^{24} - 1$), and are independent of the SD PDU sequence numbers. The receiver, upon receipt of a POLL, responds with a STAT PDU. The STAT PDU contains the following information: an SD sequence number (which sets the maximum window at which the transmitter can transmit), the number of the next expected SD PDU, the echoed poll sequence number, and a list of all SD PDUs that are currently missing or have been received erroneously. The receiver can determine the missing SD PDUs by checking for gaps in the sequence numbers of the received SD PDUs and by examining the SD sequence number contained in the POLL. Based on the received STAT message, the transmitter retransmits the outstanding SD PDUs and advances the transmit window.

The window granted by the SSCOP receiver allows the peer SSCOP transmitter to transmit new SD PDUs. The process by which the receiver determines how many new SD PDUs the transmitter can transmit is not part of the standard. Typically, it should be a function of buffer availability at the receiver and of the bandwidth-delay product. The SSCOP receiver allocates a buffer to support each connection. In principle, this buffer should match or exceed the window granted by the receiver, in order to avoid discarding of successfully transmitted data.

If the receiver detects a missing or erroneous SD PDU, it sends a USTAT immediately, instead of having to wait for a POLL. A USTAT PDU is identical to a STAT PDU except that it is not associated with a POLL. A USTAT PDU is not sent by the receiver, if the same SD PDU is found missing for the second time.

SSCOP does not have the means to check for erroneously received SD PDUs. SSCOP runs on top of AAL 5, and the payload of each AAL 5 CPS-PDU contains an SD PDU. Upon receipt of an AAL 5 CPS-PDU, the CPS checks for errors using the CRC check. If it finds that the received CPS-PDU is in error, it still passes its payload to SSCOP with a notification that it is in error. SSCOP checks for invalid PDUs. A PDU is invalid if its PDU type is unknown, or it is not 32-bit aligned, or it does not have the proper length for a PDU of the stated type. Invalid PDUs are discarded.

Lost SD PDUs are detected by SSCOP by checking whether there is a missing SD PDU sequence number. An SD PDU might be lost for a variety of reasons; it might have arrived when the SSCOP receiver buffer was full, or it might have been invalid, in which case, the SSCOP discarded it. It will also be lost in the unlikely case where all of the ATM cells carrying the SD PDU are lost.

5.3.2 Primitives

In the OSI model, the functions of a layer n provide a set of services which can be used by the next higher layer $n + 1$. The function of a layer n are built on services it requires from the next lower layer $n - 1$. The services in each layer are provided through a set of *primitives*. Each primitive might have one or more parameters that convey information required to provide the service. There are four basic types of primitives: *request, indication, response*, and *confirm*. A *request type primitive* is passed from a layer n to a lower layer $n - 1$ to request a service to be initiated. An *indication type primitive* is passed from a layer $n - 1$ to a higher layer n to indicate an event or condition significant to layer n. A *response type primitive* is passed from a layer n to a lower layer $n - 1$ to complete a procedure previously invoked by an indication primitive. Finally, a *confirm type primitive* is used by a layer $n - 1$ to pass results from one or more previously invoked request primitives to an upper layer n. These primitive types are also used between a signaling protocol and SAAL (see Figure 5.4).

SAAL functions are accessed by a signaling protocol through the AAL-SAP, using the primitives: AAL-ESTABLISH, AAL-RELEASE, AAL-DATA, and AAL-UNIT-DATA.

The AAL-ESTABLISH is issued by a signaling protocol to SAAL in order to request the establishment of a connection over the UNI to its peer protocol. This is necessary, in order for the two peer signaling protocol to exchange signaling messages. This is a reliable connection that is managed by the SSCOP as described above.

The AAL-RELEASE primitive is a request by a signaling protocol to SAAL to terminate a connection established earlier on using the AAL-ESTABLISH primitive.

The AAL-DATA primitive is used by a signaling protocol to request the transfer of a signaling message to its peer signaling protocol. Signaling messages have a specific structure, and will be discussed below in detail. Finally the AAL-UNIT-DATA is used to request a data transfer over an unreliable connection.

An example of how these primitives are used to establish a new connection over the UNI between two peer signaling protocols is shown in Figure 5.5. The primitive AAL-ESTABLISH.request is used to request SAAL to establish a connection. (In order to simplify the presentation, we do not present the signals exchanged between the SSCF and the SSCOP). In response to this request, SSCOP sends a BEGIN frame to its peer SSCOP.

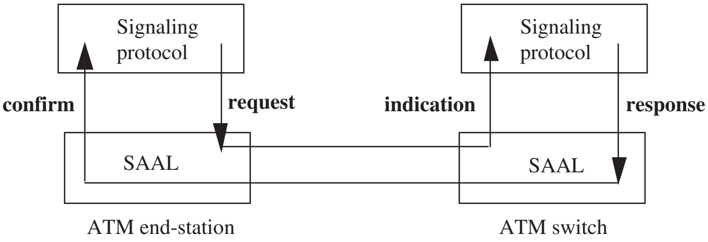


Figure 5.4 The four primitive types.

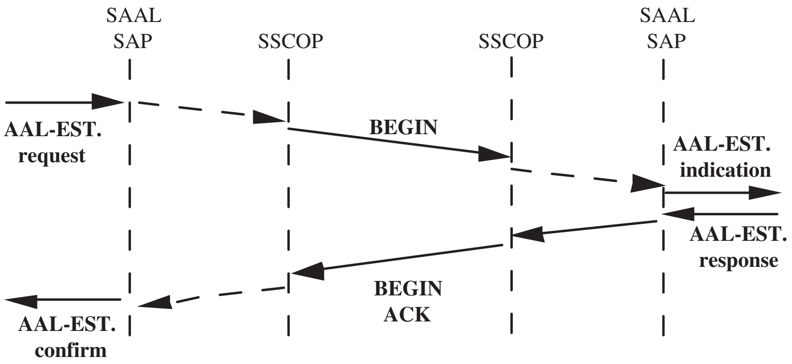


Figure 5.5 Establishment of a connection between two peer signaling protocols.

The peer SAAL generates an AAL-ESTABLISH.indication to the peer signaling protocol, and its SSCOP returns a BEGIN ACKNOWLEDGE frame, upon receipt of which, the SAAL issues a AAL-ESTABLISH.confirm to the signaling protocol.

An example of how a connection over the UNI between two peer signaling protocols is terminated is shown in Figure 5.6. The signaling protocol issues an AAL-RELEASE.request to SAAL, in response of which the SSCOP sends an END frame to its peer SSCOP. The peer SAAL sends an AAL-RELEASE.indication to the peer signaling protocol, and its SSCOP returns an END ACKNOWLEDGE frame, upon receipt of which the SAAL issues a AAL-RELEASE.confirm to the signaling protocol.

An example of how a signaling protocol transfers messages to its peer protocol is shown in Figure 5.7. The signaling protocol transfers a message to SAAL in an AAL-DATA.request, which is then transferred by SSCOP in an SD frame. The SD frame is passed onto AAL 5, which encapsulates it and then breaks it up to 48 byte segments, each of which is transferred by an ATM cell. Figure 5.7 also shows the POLL/STAT frames exchanged between the two peer SSCOPs. The SD frame at the destination side is delivered to the peer signaling protocol using the AAL-DATA.indication primitive.

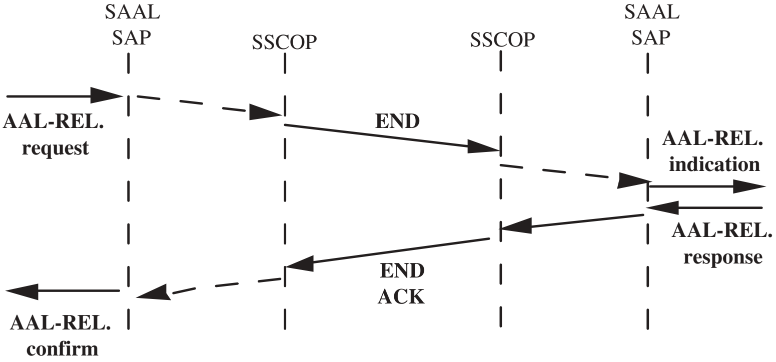


Figure 5.6 Termination of a connection between two peer signaling protocols.

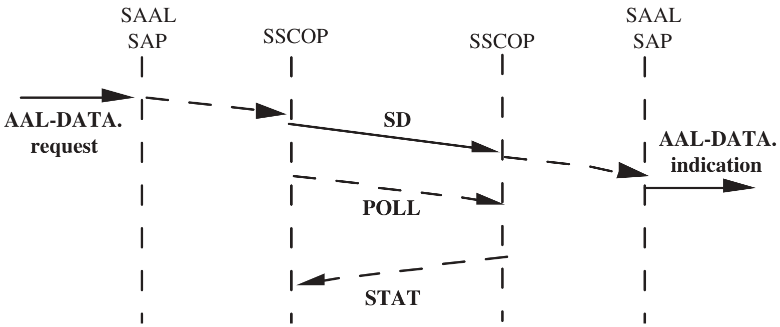


Figure 5.7 Transfer of a signaling message.

5.4 THE SIGNALING CHANNEL

This is a VC connection that is used exclusively to carry the ATM traffic that results from the exchange of signaling messages between two peer signaling protocols. It is a default connection identified by VPI = 0 and VCI = 5. This signaling channel is used to control VC connections within all of the virtual paths. It is also possible to set up a signaling channel with a VCI = 0 within a virtual path connection with a VPI other than 0, say with a VPI = x. In this case, this signaling channel can only be used to control VC connections within the virtual path x.

The signaling channel VPI/VCI = $0/5$ is used in conjunction with the signaling mode known as *non-associated signaling*. In this mode, all of the VC connections are created, controlled, and released via the signaling channel VPI/VCI = $0/5$. A signaling channel within a VPI = x, where $x > 0$, is used in conjunction with the signaling mode known as *associated signaling*. In this mode, only the VC connections within the virtual path x are created, controlled, and released via the signaling channel VPI/VCI = $x/5$.

5.5 ATM ADDRESSING

Each ATM end device and each ATM switch has a unique ATM address. Private and public networks use different ATM addressing formats. Public ATM networks use E.164 addresses, whereas ATM private network addresses use the OSI *network service access point (NSAP)* format.

The E.164 addressing scheme is based on the global ISDN numbering plan (see Figure 5.8). It consists of sixteen digits, each of which are coded in *binary coded decimal (BCD)* using 4 bits. Thus, the total length of the E.164 address is 64 bits (or 8 bytes). The first digit indicates whether the address is a unicast or multicast. The next three digits indicate the country code, and the remaining digits are used to indicate an area or city

digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U, M	Country code			Area, city, exchange, end-system											

Figure 5.8 The E.164 addressing scheme.

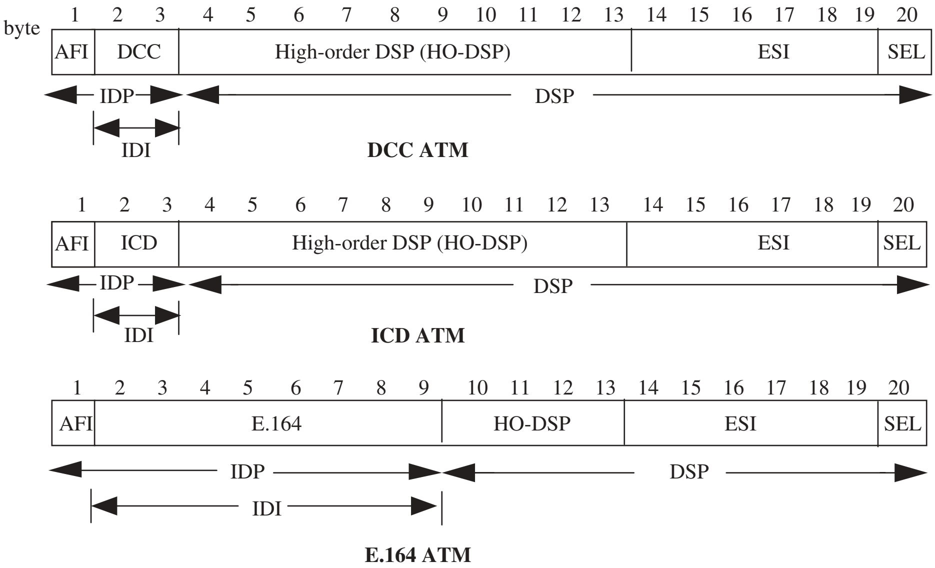


Figure 5.9 The NSAP ATM formats.

code, an exchange code, and an end device identifier. When connecting a private ATM network to a public network, only the UNIs connected directly to the public network have an E.164 address.

Private ATM addresses are conceptually based on hierarchical addressing domains. The address format is twenty bytes long and consists of two parts: the *initial domain part (IDP)* and the *domain-specific part (DSP)*. (See Figure 5.9.) The IDP specifies an administration authority which has the responsibility for allocating and assigning values for the DSP. It is subdivided into the *authority and format identifier (AFI)* and the *initial domain identifier (IDI)*. AFI specifies the format of the IDI, and the abstract syntax of the DSP field. The length of the AFI field is 1 byte. The IDI specifies the network addressing domain, from which the DSPs are allocated and the network addressing authority responsible for allocating values of the DSP from that domain. The following three IDIs have been defined by the ATM Forum:

a. *DCC (data country code)*: This field specifies the country in which the address is registered. The country codes are specified in ISO 3166. These addresses are administered by the ISO's national member body in each country. The length of this field is two bytes, and the digits of the data country code are encoded using BCD.
b. *ICD (international code designator)*: The ICD field identifies an authority which administers a coding scheme. This authority is responsible for the allocation of identifiers within this coding scheme to organizations. The registration authority for the international code designator is maintained by the British Standards Institute. The length of the field is two bytes and the digits of the international code designator are encoded using BCD.
c. *E.164 addresses.*

The DSP field consists of the *high-order DSP (HO-DSP)* field, the *end system identifier (ESI)* field and the *selector (SEL)* field. The coding for the HO-DSP field is specified by the authority or the coding scheme identified by the IDP. The authority determines how identifiers will be assigned and interpreted within that domain. The authority can create further subdomains. That is, it can define a number of subfields of the HO-DSP and use them to identify a lower authority which in turn defines the balance of HO-DSP. The content of these subfields describe a hierarchy of addressing authorities and convey a topological structure.

The *end system identifier (ESI)* is used to identify an end device. This identifier must be unique for a particular value of the IDP and HO-DSP fields. The end system identifier can also be globally unique by populating it with a 48-bit IEEE MAC address. Finally, the *selector (SEL)* field has only local significance to the end device; it is not used in routing. It is used to distinguish different destinations reachable at the end device.

Of interest is how IP addresses can be mapped to the NSAP structure. Figure 5.10 shows a mapping which can eliminate the use of ATMARP.

To show how the NSAP ATM addresses can be used, we refer to the ATM addressing scheme for the private ATM network, *North Carolina Advanced Network (NCANet)*. NCANet is a production network that is used for research and education purposes. The ICD format was selected (see Figure 5.9). The US GOSIP coded HO-DSP field was used, which consists of: a 1-byte *domain format identifier (DFI)* field; a 3-byte *administrative authority (AA)* field; a 2-byte reserved field; a 2-byte *routing domain number (RDN)* field; and a 2-byte AREA field. The fields were populated as follows (in hexadecimal):

- *AFI* = 47: Indicates that an ICD ATM format is used.
- *ICD* = 0005: Indicates that a GOSIP (NIST) coded HO-DSP field is used.
- *DFI* = 80: Indicates that the next 3 bytes represent the administrative authority; in this case, it indicates the Micro Electronics Center of North Carolina (MCNC), which is responsible for handling the regional traffic.
- *AA* = FFEA00: Assigned to MCNC by GOSIP (NIST).
- *Reserved field* = 0000.
- *RDN* = *xxxx*: To be assigned by MCNC. For instance, North Carolina State University is part of NCANet, and it has been assigned the RND value of 0101.
- *AREA* = *yyyy*: To be assigned by the RDN owner. For instance, a group of ATM addresses at North Carolina State University have been assigned the AREA value of 1114.

As a result, all ATM addresses of ATM end devices and ATM switches in NCANet have the following NSAP prefix (in hexadecimal):

47.0005.80.FFEA00.0000.xxxx.yyyy.

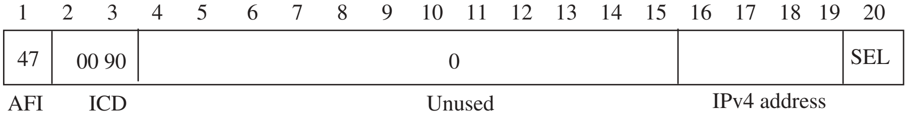


Figure 5.10 The NSAP ATM format for IP addresses.

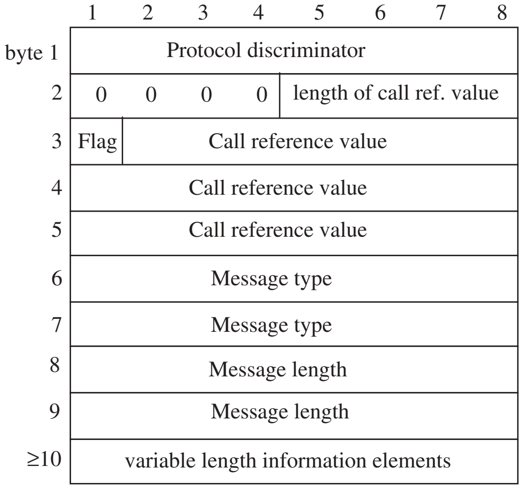
The following address (in hexadecimal) is an example of the complete ATM address of an ATM switch in the ATM Lab of North Carolina State University:

47.0005.80.FFEA00.0000.0101.1114.400000000223.00.

The first thirteen bytes provide the prefix, equal to: 47.0005.80.FFEA00.0000.0101.1114. The next field is the *end system identifier (ESI)*; it is populated with the value of 400000000223, which is the IEEE MAC address of the switch. The final field is the *selector (SEL)*, which is populated with the value 00.

5.6 THE FORMAT OF THE SIGNALING MESSAGE

The format of the signaling message is shown in Figure 5.11. This message format is used by the signaling protocols Q.2931, Q.2971, and PNNI. The protocol discriminator field is used to identify the signaling protocol. Bytes 3 to 5 give the call reference number to which the signaling message pertains. This is simply a number assigned to each call (i.e., connection) by the side that originates the call. It is a unique number that has local significance, and it remains fixed for the lifetime of the call. After the call ends, the call reference value is released and it can be used for another call. The call reference value is used by the signaling protocol to associate messages to a specific call, and it has nothing to do with the VPI/VCI values that will be assigned to the resulting ATM connection. The length of the call reference value is indicated in byte 2. For instance, 0011 indicates a 3-byte length. Since the call reference value is selected by the side that originates the call, two calls originating at the opposite sides of the interface might have the same call reference value. The call reference flag, in byte 3, is used to address this problem. Specifically, the side that originates the call sets the flag to 0 in its message, whereas the destination sets the flag to 1 when it replies to a message sent by the originating side.



	1	2	3	4	5	6	7	8
byte 1	Protocol discriminator							
2	0	0	0	0	length of call ref. value			
3	Flag	Call reference value						
4	Call reference value							
5	Call reference value							
6	Message type							
7	Message type							
8	Message length							
9	Message length							
≥10	variable length information elements							

Figure 5.11 The signaling message format.

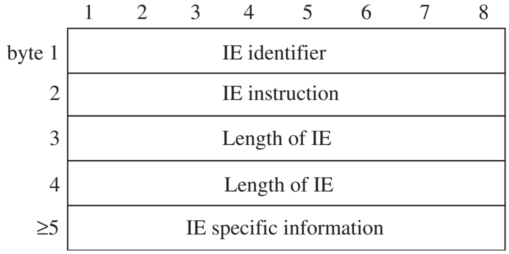


Figure 5.12 The structure of an information element.

The message type field of the signaling message, bytes 6 and 7, is used to identify the type of the signaling message.

The message length field, bytes 8 and 9, is used to indicate the length of the signaling message, excluding the first nine bytes. Typically, there is a variety of information that has to be provided with each signaling message. This information is organized into different groups, known as *information elements (IE).* Each signaling message contains a variable number of information elements, which are appended to the signaling message starting at byte 10. The total length of all of the information elements appended to a signaling message is given in the message length field. The structure of an information element is shown in Figure 5.12. The first byte contains the IE identifier, which is used to uniquely identify the information element. The second byte contains various fields, such as the coding standard, i.e., ITU-T, ISO/IEC, national, network specific standard (private or public), and the IE action indicator. Bytes 3 and 4 give the length of the information element, excluding the first four bytes, and the remaining bytes starting at byte 5 contain the information specific to the IE.

5.7 THE SIGNALING PROTOCOL Q.2931

This protocol establishes a point-to-point SVC over the private UNI in real-time. In this section, we first examine the information elements used in the Q.2931 messages, and then we describe the Q.2931 messages and we show how they are used to establish and terminate a call. The terms *calling user* or *party* and, conversely, *called user* or *party* are used interchangeably. The *calling user* is a user in the end device that initiates a call, and the *called user* is the user at the end device that is being called.

5.7.1 Information Elements (IE)

Each signaling message contains a variety of information organized into different groups, known as *information elements (IE).* The following are some of the information elements used in Q.2931 messages:

- *AAL parameter IE*: Indicates the AAL parameter values used between the end devices.
- *ATM traffic descriptor IE*: Specifies the traffic parameters in the forward and backward direction of the connection.
- *Broadband bearer capability IE*: Used to define the ATM service requested for a new connection.

- *Broadband high-layer IE, broadband low-layer IE*: Used to check compatibility by the called user.
- *Broadband repeat indicator IE*: Indicates how repeated IEs are to be interpreted.
- *Call state*: Describes the current status of the call.
- *Called party number IE, and called party subaddress IE*: Identify the called user.
- *Calling party number IE, and calling party subaddress IE*: Identify the calling user.
- *Cause IE*: Describes the reason for generating certain messages and indicates the location of the cause originator.
- *Connection identifier IE*: Identifies the VPI/VCI allocated to the connection at the UNI.
- *End-to-end transit delay IE*: Indicates the maximum acceptable transit delay and the cumulative transit delay to be expected for the connection.
- *Extended QoS parameters IE*: Specifies the acceptable values and the cumulative values of some of the QoS parameters.
- *Transit network selection IE*: Identifies a transit network that the call can cross.

An ATM end device or an ATM switch might not be able to process every information element included in a signaling message. In this case, the ATM equipment simply uses only the information elements that it needs, and it ignores the rest of them.

5.7.2 Q.2931 Messages

The Q.2931 messages can be grouped into the following three categories: call establishment, call clearing, and miscellaneous. The messages for each category are given in Table 5.2. Each Q.2931 message uses the signaling message format described in Section 5.6, with the protocol discriminator set to 00001001, and contains a set of information elements. Below, we describe the function of each message, and then we show how they are used to establish and clear a call.

- *ALERTING*: This message is sent by the called user to the network and by the network to the calling user to indicate that "called user alerting" has been initiated. Called user

Table 5.2 Q.2931 messages.

Message category	Message
Call establishment messages	ALERTING
	CALL PROCEEDING
	CONNECT
	CONNECT ACKNOWLEDGEMENT
	SETUP
Call clearing messages	RELEASE
	RELEASE COMPLETE
Miscellaneous messages	NOTIFY
	STATUS
	STATUS ENQUIRY

alerting is used for calls that require human interface, such as voice. The information element used in this message is the connection identifier.

- *CALL PROCEEDING*: The message is sent by the called user to the network or by the network to the calling user, to indicate that the requested establishment of the call has been initiated and no more call information is accepted. The information element used in this message is the connection identifier.
- *CONNECT*: The message is sent by the called user to the network, or by the network to the calling user to indicate that the called user has accepted the call. The following information elements are used in this message: AAL parameter, broadband low-layer, connection identifier, and end-to-end transit delay.
- *CONNECT ACKNOWLEDGEMENT*: This message is sent by the network to the called user to indicate that the user has been awarded the call. It is also sent by the calling user to the network to allow symmetrical call control procedures.
- *RELEASE*: This message is sent by the user to request the network to clear an end-to-end connection. It is also sent by the network to indicate that an end-to-end connection is cleared. It also indicates that the receiving equipment is to release the connection identifier, and to prepare to release the call reference value after sending RELEASE COMPLETE. The cause information element is carried in this message.
- *RELEASE COMPLETE*: This message is sent by the calling user or the network to indicate that the equipment sending the message has released its call reference value and, if appropriate, the connection identifier. The cause information element is carried in this message.
- *SETUP*: This message is sent by the calling user to the network and by the network to the called user to initiate the establishment of a new call. The following information elements are used: AAL parameter, ATM traffic descriptor, broadband bearer capability, broadband high-layer, broadband low-layer, called party number, called party subaddress, calling party number, calling party subaddress, connection identifier, end-to-end transit delay, extended QoS parameters, and transit network selection.

The NOTIFY message is sent by the user or the network to indicate information pertaining to a call. The STATUS message is sent by the user or the network in response to a STATUS ENQUIRY message. Finally, the STATUS ENQUIRY message is sent by the user or the network to solicit a STATUS message from the peer Q.2931 protocol.

a. Call establishment

The steps involved in establishing a call are shown in Figure 5.13. The calling user initiates the procedure for establishing a new call by sending a SETUP message to its ingress ATM switch across the UNI. The ingress switch sends a CALL PROCEEDING message to the calling user if it determines that it can accommodate the new call. (If it cannot accommodate the new call, it rejects it by responding with a RELEASE COMPLETE message.) The ingress switch calculates a route to the destination end device over which the signaling messages are transferred. The same route is used to set up a connection over which the data will flow. It then forwards the SETUP message to the next switch on the route. The switch verifies that it can accommodate the new connection, and the forwards the SETUP message to the next switch, and so on, until it reaches the end device of the called user. The PNNI protocol is used to progress the SETUP message across the network.

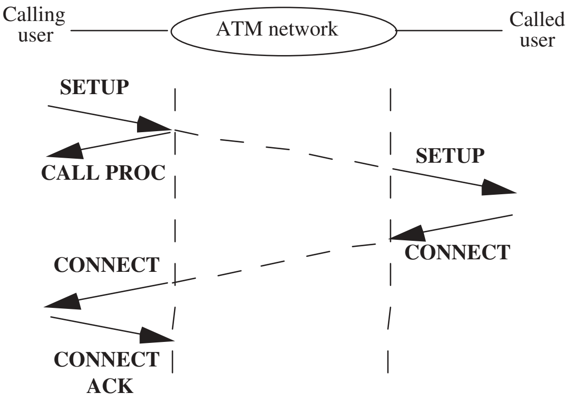


Figure 5.13 Call establishment.

If the called user can accept the call it responds with CALL PROCEEDING, ALERTING, or CONNECT message. (Otherwise, it sends a RELEASE COMPLETE message.) Upon receiving an indication from the network that the call has been accepted, the ingress switch sends a CONNECT message to the calling user, who responds with a CONNECT ACKNOWLEDGMENT.

b. Call clearing

Call clearing is initiated when the user sends a RELEASE message. When the network receives the RELEASE message, it initiates procedures for clearing the connection to the remote user. Once the connection has been disconnected, the network sends a RELEASE COMPLETE message to the user, and releases both the call reference value and the connection identifier. Upon receipt of RELEASE COMPLETE message the user releases the connection identifier and the call reference value.

PROBLEMS

1. Why does the HDLC selective-reject ARQ does not work well in a network with a high bandwidth-delay product?
2. What are the basic differences between the error recovery scheme in the SSCOP and the more traditional ARQ schemes, such as go-back-n and selective reject?
3. Describe the sequence of primitives issued to set up a connection between two peer signaling protocols.
4. What is the purpose of the call reference flag in the signaling message?
5. In which information element, the calling user indicates its traffic parameters?
6. In which information elements the calling user indicates the QoS parameters?
7. Trace the sequence of the signaling messages issued to set up a connection.

6

The Multi-Protocol Label Switching (MPLS) Architecture

The *multi-protocol label switching (MPLS)* scheme is based on Cisco's *tag switching*, which in turn was inspired by the *IP switching scheme*, an approach to switching IP packets over ATM proposed by Ipsilon Networks (Ipsilon was later on purchased by Nokia). MPLS was standardized by IETF, and it introduces a connection-oriented structure into the otherwise connectionless IP network. MPLS circumvents the CPU-intensive table look-up in the forwarding routing table necessary to determine the next hop router of an IP packet. Also, it can be used to introduce QoS in the IP network. Interestingly enough, since the introduction of tag switching, and subsequently of MPLS, several CPU-efficient algorithms for carrying out table look-ups in the forwarding routing table were developed. The importance of MPLS, however, was by no means diminished since it is regarded as a solution for introducing QoS into the IP networks.

MPLS requires a set of procedures for the reliable distribution of label bindings. MPLS does not require that a single label distribution protocol is used. In view of this, various schemes have been proposed for the distribution of labels, of which the *label distribution protocol (LDP)* and the *resource reservation protocol – traffic engineering (RSVP–TE)* are the most popular.

In this chapter, we describe the basic features of the MPLS architecture. The label distribution protocols LDP and its extension CR-LDP, and RSVP and its extension RSVP-TE are presented in the following chapter. MPLS has been extended to *generalized MPLS (GMPLS)*, which is described in Section 9.5. Before we proceed to describe the MPLS architecture, we review some basic concepts of IP networks in the following section. This section can be skipped by the knowledgeable reader.

6.1 THE INTERNET PROTOCOL (IP): A PRIMER

IP is part of the TCP/IP suite of protocols used in the Internet. TCP corresponds to the transport layer of the OSI model, and IP corresponds to the network layer of the OSI model. In this section, we describe the current version of IP, known as *IP version 4 (IPv4)*.

IP provides a connectionless service using packet switching with datagrams. Packets in a connectionless network, such as the IP network, are referred to as *datagrams*. An IP host can transmit datagrams to a destination IP host without having to set up a connection to the destination, as in the case of X.25, frame relay, and ATM networks. IP datagrams

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

are routed through the IP network independently from each other, and in theory, they can follow different paths through the IP network. In practice, however, the IP network uses routing tables that remain fixed for a period of time. In view of this, all IP packets from a sender to a receiver typically follow the same path. These routing tables are refreshed periodically, taking into account congested links and hardware failures of routers and links.

IP does not guarantee delivery of IP datagrams. In view of this, if the underlying network drops an IP datagram, IP will not be aware of that. Also, as in the ATM networks, IP does not check the payload of an IP datagram for errors, but it only checks its IP header. IP will drop an IP datagram, if it finds that its header is in error. Lost or erroneous data is recovered by the destination's TCP.

6.1.1 The IP Header

An IP datagram consists of a header and a payload. The IP header is shown in Figure 6.1, and it consists of a 20-byte fixed part and an optional part which has a variable length. The following fields are defined in the IP header:

- *Version*: A 4-bit field used to indicate which version of the protocol is used.
- *Internet Header Length (IHL)*: This is a 4-bit field that gives the length of the header in 32-bit words. The minimum header length is five 32-bit words (or 20 bytes).
- *Type of service*: This is an 8-bit field that indicates whether the sender prefers for the datagram to travel over a route with minimal delay, or a route with maximal throughput.
- *Total length*: A 16-bit field used to indicate the length of the entire datagram (i.e., header and payload). The default value for the maximum length is 65,535 bytes.
- *Identification*: A 16-bit field used by the receiver to identify the datagram that the fragment belongs to. All fragments of a datagram have the same value in the identification field.
- *Flags*: This is a 3-bit field, but only two bits – *more fragments* and *don't fragment* – are used. All fragments, except the last one, have the more fragments bit set. This information permits the receiver to know when all of the fragments have arrived. The don't fragment bit is used to disallow fragmentation.

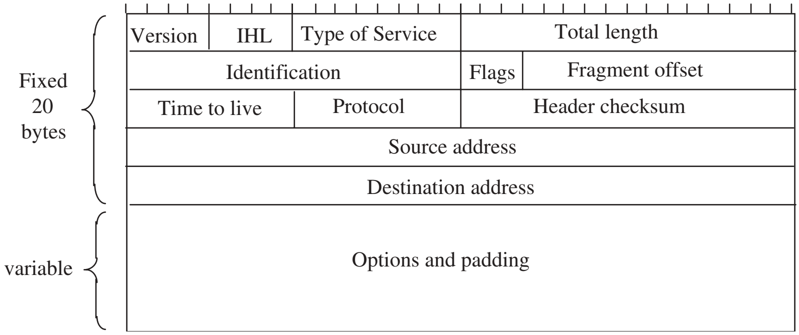


Figure 6.1 The IPv4 header.

- *Fragment offset*: This 13-bit field contains an offset that points to where this fragment belongs to in the original datagram.
- *Time to live*: This is an 8-bit field that specifies in seconds how long a datagram is allowed to live in the network. The maximum lifetime is 255 sec. Every router that processes the datagram must decrease this field by one second, and by several seconds if the datagram is queued in the router for a long time. This field can be seen as being similar to a hop count. When the time to live field becomes equal to 0, the datagram is discarded. This prevents a datagram from moving around in the network forever.
- *Protocol*: This field is 8 bits long; it specifies the next higher level protocol (e.g. TCP and UDP) to which the datagram should be delivered.
- *Header checksum*: A 16-bit field used to verify whether the IP header has been correctly received. The transmitting host adds up all of the 16-bit half-words of the header using 1's compliment arithmetic, assuming that the checksum field is 0. The 1's compliment of the final result is then computed and placed in the checksum field. The receiving host calculates the checksum, and if the final result is 0, then the header has been correctly received. Otherwise, the header is erroneous and the datagram is dropped. The checksum is recomputed at each router along the path of the datagram, since at least one field of the header (the time to live field) is changed.
- *Source address*: A 32-bit field populated with the network and host number of the sending host.
- *Destination address*: A 32-bit field populated with the network and host number of the destination host. The IP addressing scheme is discussed below.
- *Options*: A variable-length field used to encode the options requested by the user (e.g. security, source routing, route recording, and time stamping).
- *Padding*: A variable-length field used to make the header of the datagram an integral multiple of 32-bit words.

6.1.2 IP Addresses

As we saw above, IP addresses are 32 bits long. An IP address is divided into two parts: a network and a suffix. The network identifies the physical network that the host computer is attached to. The suffix identifies the host computer itself. The size of these two fields vary according to the class of the IP address. Specifically, five different classes of addresses – A, B, C, D, and E – have been defined (see Figure 6.2).

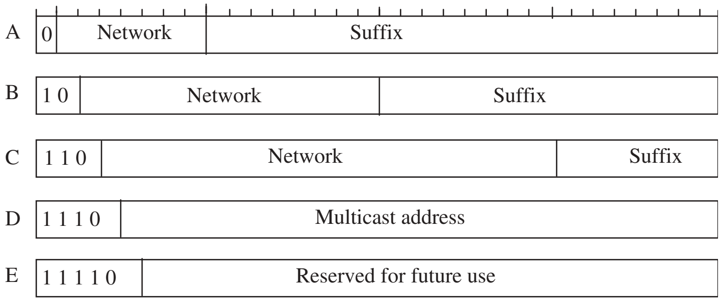


Figure 6.2 The IP address classes.

Classes A, B and C are called the *primary classes* because they are used for host addresses. Class D is used for multicasting; class E is reserved for future use. The first field determines the class of the IP address, and it ranges from 1 bit for a class A address to 5 bits for a class E addresses. The second field gives the network address, and the third field is the suffix which gives the host address.

In class A, there is a 7-bit network address and a 24-bit host address, resulting to 128 network addresses and 16,777,216 host addresses. In class B, there is a 14-bit network address and a 16-bit host address, resulting to 16,384 network addresses and 65,536 host addresses. In class C, there is a 21-bit network address and an 8-bit host address, resulting to 2,097,152 network addresses and 256 host addresses.

Network addresses are usually written in the *dotted decimal notation*. That is, each byte is written in decimal, ranging from 0 to 255. As an example, the IP address 00000111 00000010 00000000 00000010 will be written as 7.2.0.2. Using this notation, we have that the range of class A addresses is from 1.0.0.0 to 127.255.255.255, for class B we have a range of values from 128.0.0.0 to 191.255.255.255, and for class C we have a range of 192.0.0.0 to 233.255.255.255.

Class C is very common, whereas class A is rarely used since there are only few networks with that large number of hosts. IP reserves host address 0 to denote the address of a network. For instance, in the class B address 128.32.0.0 the network field is 128.32 and the suffix is 0.0. This indicates the address of the network 128.32. For broadcasting within the network, IP uses the address 128.32.255.255.

IP assigns multiple IP addresses to routers, since a router is attached to multiple networks. Specifically, a router has one IP address for each network that it is attached to. An individual host connected to multiple networks has also multiple IP addresses, one for each network connection. Such a host is referred to as *multihomed*.

Subnetting

The IP address structure described above introduces a two-level hierarchy. The first level is the network address and the second level is the host address carried in the suffix. In many cases, these two levels of addressing is not enough. For instance, if we consider an organization with a B class address, then all of the hosts appear to be organized into a single group, described by the network address. However, hosts within an organization are typically grouped together to form a number of different LANs. In order to distinguish the LANs the suffix of the IP address is subdivided into a subnet part and a host part. Each LAN is assigned a subnet address carried in the subnet part, and a host in the LAN is assigned an address which is carried in the host part. The actual parsing of the suffix in these two subfields is dictated by a subnet mask. The subnet mask is only known to the routers within the network since the subnets are not visible outside the network. This technique is known as *subnetting*.

Classless inter-domain routing (CIDR)

In the early 90s, it became apparent that the rapid expansion of the Internet would cause a depletion of IP addresses and an explosion of the routing tables. The main cause for address depletion was the wasteful usage of class B addresses; many organizations used a class B address, but only had a small number of hosts, thus leaving the host address space largely unused. The routing table explosion was due to the fact that a router keeps all of the addresses of all of the registered networks.

In order to alleviate these two problems the *classless inter-domain routing (CIDR)* scheme was proposed. This scheme permits the assignment of contiguous class C addresses and at the same time it reduces the number of entries required in a routing table.

The basic idea in CIDR is to allocate blocks of class C network addresses to each ISP. Organizations using the ISP are suballocated a block of 2^n contiguous addresses. For instance, if an organization requires 2000 addresses, then it will be allocated a block of 2048 (i.e. 2^8) contiguous class C addresses.

Hierarchical suballocation of addresses in this manner implies that clients with addresses allocated out of a given ISP will be routed via the ISP's network. This permits all of these addresses to be advertised outside the ISP's network in an aggregate manner. As an example, let us assume that an ISP was allocated 131,072 class C network addresses starting at 194.0.0.0. That means that the lowest network address is 194.0.0.0 or 11000010 00000000 00000000 00000000 and the highest network address is 195.255.255.255 or 11000011 11111111 11111111 11111111. Any address whose first seven bits are 1100001 belongs to the group of addresses allocated to the ISP. This *prefix* can be calculated by performing a bit-wise AND operation between the lowest address and the mask 254.0.0.0 or 11111110 00000000 00000000 00000000. Routers outside the ISP's network are provided, therefore, only with the base address 194.0.0.0 and the mask 254.0.0.0. This information suffices in order to identify whether an address of an IP packet has the same prefix as the ISP. Calculating a prefix using a base network address and a mask is known as *supernetting*. Supernetting is the converse of subnetting.

The above use of contiguous addresses gives rise to better usage of the address space. Also, by only advertising a base address and a mask, the amount of information that a router has to keep in its routing table is minimized. Note that some network addresses were allocated prior to CIDR, and a router has to keep these addresses in its table as well.

To further simplify routing, blocks of addresses were also allocated according to geographic regions (see Table 6.1).

Finally, note that the class A, B, and C addresses are no longer used for routing. Instead, CIDR is applied to all addresses, which explains why this scheme is called *classless*.

6.1.3 ARP, RARP, and ICMP

The TCP/IP protocol suite includes other protocols such as the *address resolution protocol (ARP)*, the *reverse address resolution protocol (RARP)* and the *Internet control message protocol (ICMP)*.

Table 6.1 Allocation of addresses per region.

Region	Lower address	Higher address
Europe	194.0.0.0	195.255.255.255
North America	198.0.0.0	199.255.255.255
Central/South America	200.0.0.0	201.255.255.255
Pacific Rim	202.0.0.0	203.255.255.255

ARP is used to translate a host's IP address to its corresponding hardware address. This address translation is known as *address resolution*. The ARP standard defines two basic messages: a *request* and a *response*. A request message contains an IP address and requests the corresponding hardware address. A reply message contains the IP address sent in the request and the hardware address.

RARP does the opposite to ARP. It identifies the IP address of a host that corresponds to a known hardware address.

ICMP defines several error and information messages used in the Internet to report various types of errors or send various types of information. Some of the principal messages are: *source quench, time exceeded, destination unreachable, redirect, fragmentation required, parameter problem, echo request/rep*ly, and *timestamp request/reply*.

A *source quench* message is sent by a router when it has run out of buffer space and it cannot accept more datagrams. A *time exceeded* message is sent by a router when the time to live field in a datagram is 0. The datagram is dropped by the router. The same message is also used by a host if the reassembly timer expires before all fragments from a given datagram have arrived. A *destination unreachable* message is sent by a router to a host that created a datagram, when it decides that the datagram cannot be delivered to its final destination. A *redirect message* is sent by a router to the host that originated a datagram, if the router believes that the datagram should have been sent to another router. A *fragmentation required* message is sent by a router to the host of a datagram, if it finds that the datagram is larger than the *maximum transfer unit (MTU)* of the network over which it must be sent. The datagram is rejected by the router. A *parameter problem* message is used to indicate that an illegal value has been discovered in the IP header of a datagram. *Echo reply* and *echo request* is used to test if a user destination is reachable and alive. *Timestamp request* and *timestamp reply* are similar to the echo request/reply messages except that the arrival time of the request message and the departure time of the reply message are also recorded.

6.1.4 IP Version 6 (IPv6)

Due to the rapid growth of the Internet, it was felt that the address space of the current IP will soon be inadequate to cope with the demand for new IP addresses. This consideration coupled with the need to provide new mechanisms for delivering real-time traffic, such as audio and video, led to the development of a new IP, known as *IPv6*.

IPv6 retains many of the basic concepts from IPv4. The new features are 128-bit addresses, new header format, extension headers, support for audio and video, and extensible protocol.

6.2 THE MULTI-PROTOCOL LABEL SWITCHING (MPLS) ARCHITECTURE

MPLS is an IETF standard based on Cisco's tag switching. The original intention was to be used in conjunction with different networking protocols, such as IPv4, IPv6, IPX and AppleTalk. However, MPLS has been developed exclusively for IP networks, which makes the name of the protocol more general than it is in reality.

In order to understand the basic concept behind MPLS, we need to take a look at how an IP router works. An IP router implements both control and forwarding components. The control component consists of routing protocols, such as the *open shortest path* *first (OSPF)*, the *border gateway protocol (BGP)*, and the *protocol independent multicast (PIM)*, used to construct routes and exchange routing information among IP routers. This information is used by the IP routers to construct the forwarding routing table, referred to as the *forwarding information base (FIB)*. The forwarding component consists of procedures that a router uses to make a forwarding decision on an IP packet. For instance, in unicast forwarding, the router uses the destination IP address to find an entry in the FIB, using the longest match algorithm. The result of this table look-up is an interface number, which is the output port connecting the router to the next hop router, to which the IP packet should be sent.

A router forwards an IP packet according to its prefix. In a given router, the set of all addresses that have the same prefix, is referred to as the *forwarding equivalent class (FEC*, pronounced as *fek)*. IP packets belonging to the same FEC have the same output interface. In MPLS, each FEC is associated with a different *label*. This label is used to determine the output interface of an IP packet without having to look-up its address in the FIB. A label is a short fixed-length identifier that has local significance. That is, it is valid on a single hop interconnecting two routers. A label is similar in functionality to the VPI/VCI value associated with an ATM cell.

In IPv6, the label can be carried in the flow label field. In IPv4, however, there is no space for such a label in the IP header. If the IP network runs on top of an ATM network, then the label is carried in the VPI/VCI field of an ATM cell. If it is running over frame relay, the label is carried in the DLCI field. For Ethernet, token ring, and point-to-point connections that run a link layer protocol (e.g. PPP), the label is encapsulated and inserted between the LLC header and the IP header (see Figure 6.3). (Note that in tag switching, the encapsulated label was referred to as a *shim header*.) The first field of the label encapsulation is a 20-bit field used to carry the label. The second field is a 3-bit field used for experimental purposes. It can for instance carry a *class-of-service (CoS)* indication, which can be used to determine the order in which IP packets will be transmitted out of an interface. The S field is used in conjunction with the label stack, which will be discussed in detail later on in this chapter. Finally, the *time-to-live (TTL)* field is similar to the TTL field in the IP header.

An MPLS network consists of *label switching routers (LSR)* and *MPLS nodes*. An LSR is an IP router that runs the MPLS protocol. It can bind labels to FECs, forward IP packets based on their labels, and carry the customary IP forwarding decision by carrying out a table look-up in the FIB using a prefix. An MPLS node is an LSR, except that it does not necessarily have the capability to forward IP packets based on prefixes.

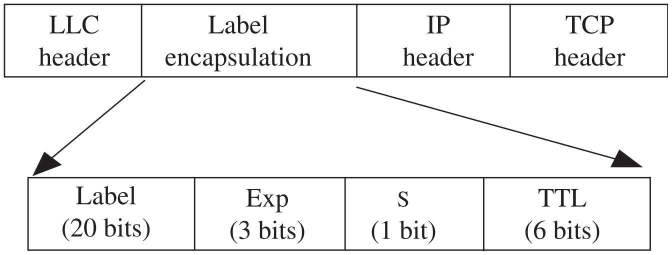


Figure 6.3 Label encapsulation.

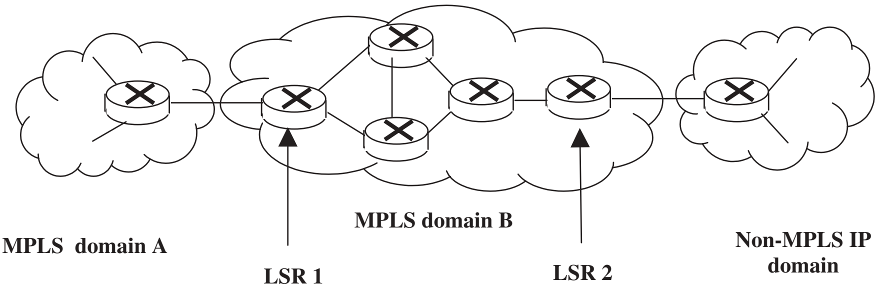


Figure 6.4 MPLS domains, LSRs, and MPLS nodes.

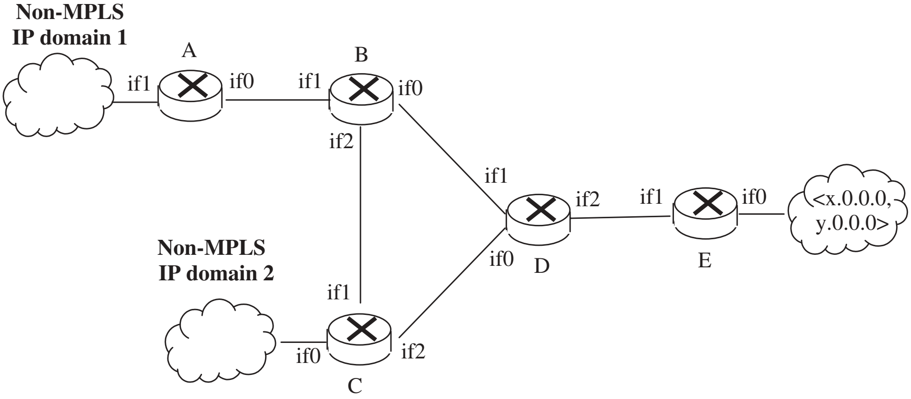


Figure 6.5 An example of multi-protocol label switching.

A contiguous set of MPLS nodes that are in the same routing or administrative domain forms an *MPLS domain*. Within an MPLS domain, IP packets are switched using their MPLS label. An MPLS domain can be connected to a node outside the domain, which might belong to an MPLS or a non-MPLS IP domain (that is, an IP domain where the routers use the customary forwarding decision based on prefixes). As shown in Figure 6.4, the MPLS domain B consists of five routers, two of which are LSRs (LSR 1 and LSR 2); the remaining three might be either LSRs or MPLS nodes. MPLS domain B is connected to the MPLS domain A via LSR 1, and is connected to the non-MPLS IP domain C via LSR 2. LSRs 1 and 2 are referred to as *MPLS edge node*s. For simplicity, we will assume that all nodes within an MPLS domain are LSRs.

To see how MPLS works, let us consider an MPLS domain consisting of five LSRs (LSRs A, B, C, D, and E), all linked with point-to-point connections as shown in Figure 6.5. LSRs A and C are connected to non-MPLS IP domains 1 and 2, respectively. Assume that a new set of hosts with the prefix ⟨x.0.0.0, y.0.0.0⟩, where x.0.0.0 is the base network address and y.0.0.0 is the mask, is directly connected to E. The flow of IP packets with this prefix from A to E is via B and D. That is, A's next hop router for this prefix is B, B's next hop router is D, and D's next hop router is E. Likewise, the flow of IP packets with the same prefix from C to E is via D. That is, C's next hop router for this prefix is D, and D's next hop router is E. The interfaces in Figure 6.5 show how these routers are interconnected. For instance, A is connected to B via if0, and B is connected to A, C, and D via if1, if2, and if0, respectively.

When a LSR identifies the FEC associated with this new prefix ⟨x.0.0.0, y.0.0.0⟩, it selects a label from a pool of free labels and it makes an entry into a table referred to as the *label forward information base (LFIB)*. This table contains information regarding the incoming and outgoing labels associated with a FEC and the output interface, i.e. the FEC's next hop router. The LSR also saves the label in its FIB in the entry associated with the FEC.

The entry in the LFIB associated with this particular FEC for each LSR is shown in Table 6.2. (For presentation purposes we have listed all of the entries together in a single table). Note that B has selected an incoming label equal to 62; D has selected 15; and E has selected 60. Since A and C are MPLS edge routers and do not expect to receive labeled IP packets, they have not selected an incoming label for this FEC. The remaining information in each entry gives the next hop LSR and the output interface for the FEC. So for this FEC, the next hop router for A is B, and is through if0.

An incoming label is the label that a LSR expects to find in all of the incoming IP packets that belong to a FEC. For instance, in the above example, LSR B expects all of the incoming IP packets belonging to the FEC associated with the prefix ⟨x.0.0.0, y.0.0.0⟩ to be labeled with the value 62. The labeling of these packets has to be done by the LSRs which are upstream of B. That is, they are upstream in relation to the flow of IP packets associated with this FEC. In this example, the only LSP that is upstream of B is A. In the case of D, both B and C are upstream LSRs.

In order for an LSR to receive incoming IP packets labeled with the value that it has selected, the LSR has to notify its neighbours about its label selection for a particular FEC. In the above example, LSR B sends its information to A, D, and C. A recognizes that it is upstream from B, and it uses the information to update the entry for this FEC in its LFIB. As far as this FEC is concerned, D and C are not upstream from B, and they do not use this information in their LFIBs. However, they can chose to store it for future use. For example, a C-to-D link failure could make B the next hop LSR for this FEC. In this case, C will use the label advertised by B to update the entry in its LFIB.

D sends its information to B, C, and E. Since B and C are both upstream of D, they use this information to update the entries in their LFIB. Finally, E sends its information

Table 6.2 FEC entry in each LFIB.

LSR	Incoming label	Outgoing label	Next hop	Outgoing interface
A	–	–	LSR B	if0
B	62	–	LSR D	if0
C	–	–	LSR D	if2
D	15	–	LSR E	if2
E	60	–	LSR E	if0

Table 6.3 FEC entry in each LFIB with label binding information.

LFIB	Incoming label	Outgoing label	Next hop	Outgoing interface
A	–	62	LSR B	if0
B	62	15	LSR D	if0
C	–	15	LSR D	if2
D	15	60	LSR E	if2
E	60	–	LSR E	if0

to D, which uses it to update its entry in its LFIB. As a result, each entry in the LFIB of each LSR will be modified (see Table 6.3).

For LSR E, the next hop is E itself. This indicates that the IP packets associated with the prefix ⟨x.0.0.0, y.0.0.0⟩ will be forwarded to the local destination over if 0 using their prefix.

Once the labels have been distributed and the entries have been updated in the LFIBs, the forwarding of an IP packet belonging to the FEC associated with the prefix ⟨x.0.0.0, y.0.0.0⟩ is done using solely the labels. Let us assume that A receives an IP packet from the non-MPLS IP domain 1 with a prefix ⟨x.0.0.0, y.0.0.0⟩. A identifies that the packet's IP address belongs to the FEC, and it looks up its LFIB to obtain the label value and the outgoing interface. It sets the label value to 62, encapsulates it using the format shown in Figure 6.3, and forwards it to the outgoing interface if0. When the IP packet arrives at LSR B, its label is extracted and looked up in B's LFIB. The old label is replaced by the new one, which is 15, and the IP packet is forwarded to interface if0. LSR D follows exactly the same procedure. When it receives the IP packet from B, it replaces its incoming label with the outgoing label, which is 60, and forwards it to interface if2. Finally, E forwards the IP packet to its local destination. The same procedure applies for an IP packet with a prefix ⟨x.0.0.0, y.0.0.0⟩ that arrives at C from the non-MPLS domain 2.

In Figure 6.6, we show the labels allocated by the LSRs. These labels are similar to the VPI/VCI values in ATM. They have *local significance*; that is, each label is valid only for

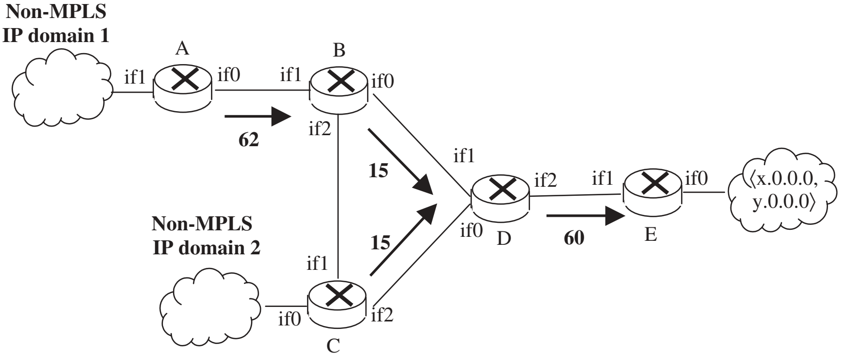


Figure 6.6 Label switched paths.

one link. The sequence of labels 62, 15, 60 form a path known as the *label switched path (LSP)*. This path is analogous to a point-to-point ATM connection, which is defined by a sequence of VPI/VCI values. An ATM connection is associated with two end devices, whereas a label switched path is associated with a FEC. Several label switched paths are typically associated with the same FEC, forming a tree diagram (see Figure 6.6). Each LSP has an *ingress LSR* and an *egress LSR*. For instance, in Figure 6.6, LSRs A and E are the ingress and egress LSRs, respectively, for the LSP from the LSR A to LSR E. Likewise, LSRs C and E are ingress and egress LSRs for the LSP from LSR C to LSR E.

Label switching eliminates the CPU-intensive table look-up in the FIB, necessary to determine the next hop router of an IP packet. A table look-up in the LFIB is not as time-consuming since an LFIB is considerably smaller than a FIB. Since the introduction of label switching, however, several CPU-efficient algorithms for carrying out table look-ups in the FIB were developed. This did not diminish the importance of label switching since it was seen as a means of introducing QoS in the IP network.

One way that QoS can be introduced in the network is to associate each IP packet with a priority. This priority can be carried in the 3-bit experimental field of the label encapsulation (see Figure 6.3). Priorities can be assigned by an MPLS edge node. Labeled IP packets within an LSR are served according to their priority as in the case of an ATM switch. Recall that, in ATM networks, each VC connection is associated with a QoS category. An ATM switch can determine the QoS of an incoming cell from its VPI/VCI value, and accordingly it can queue the cell into the appropriate QoS queue. An ATM switch maintains different QoS queues for each output interface. These queues are served using a scheduling algorithm, so that VC connections can be served according to their requested QoS. A similar queueing structure can now be introduced in an IP router. IP packets can now be queued at an output interface according to their priority, and they can be transmitted out in an order determined by a scheduler.

6.2.1 Label Allocation Schemes

In the label switching example described above, an LSR binds (i.e., allocates) a label to a FEC and saves this information in its LFIB as the incoming label. It then advertises the binding between the incoming label and the FEC to its neighboring LSRs. An *upstream LSR* (i.e., an LSR that is upstream of the link as the traffic flows) places the label in the outgoing label field of the entry in its LFIB that is associated with this FEC. A *non-upstream LSR* can either ignore the label advertisement or store it for future use. Because the LSR, which is downstream of the link with respect to traffic flow, creates the label, and because the label is advertised to its neighbors in an unsolicited manner, the label allocation scheme is known as the *unsolicited downstream scheme*.

As an example, let us consider LSR B in Figure 6.6. LSR B advertises its incoming label 65 for the FEC ⟨x.0.0.0,y.0.0.0⟩ to its neighboring LSRs A, C, and D. Of these, only LSR A is upstream to LSR B as far as the flow of IP packets towards the destination ⟨x.0.0.0,y.0.0.0⟩ is concerned. In view of this, LSR A will use this label to update its LFIB. LSRs C and D can elect to store this label binding, in case they do become upstream to LSR B as far as the flow of these IP packets is concerned. This can happen if a link or an LSR goes down. For instance, if the link between LSRs C and D is broken, LSR C might have to reroute its traffic through LSR B; in which case, it will become upstream to B. (Given the topology in Figure 6.6, LSR D will never become upstream to LSR B). LSRs C and D can also elect to ignore LSR B's advertised label binding.

A non-upstream LSR will store or ignore a label binding depending on whether the *conservative label retention mode* or the *liberal retention mode* is used. In the conservative retention mode, a label is retained only if the LSR is upstream of the LSR advertising the label binding. In the liberal label retention mode, all labels are retained irrespective of whether the LSR is upstream or not of the LSR advertising the label binding.

MPLS can also use the *downstream on demand* label allocation. In this case, each LSR binds an incoming label to a FEC and creates an appropriate entry in its LFIB. However, it does not advertise its label binding to its neighbors as in the unsolicited downstream allocation scheme. Instead, an upstream LSR obtains the label information by issuing a request.

6.2.2 The Next Hop Label Forwarding Entry (NHLFE)

So far, for presentation purposes we have assumed that an LSR maintains a single entry for each incoming label. In this entry, it binds the incoming label with an outgoing label and it provides information regarding the next hop, such as the next LSR and the output interface.

The MPLS architecture permits an LSR to maintain multiples entries for each incoming label. Each entry is known as the *next hop label forwarding entry (NHLFE)*, and it provides the following information: the packet's next hop, and the operation to be performed on the packet's label. Each NHLFE entry can also contain additional information necessary in order to properly dispose of the packet.

MPLS permits a packet to carry multiple labels which are organized as a stack. An example of the label stack is given in Figure 6.7. Each row contains a different label encapsulation. The S bit indicates whether the current label encapsulation is the last one (S = 1) or not (S = 0).

The following three operations can be performed on the packet's label:

- Replace the label at the top on the packet's label stack with a new label.
- Pop the label stack.
- Replace the label at the top of the packet's label stack with a new label, and then push one or more new labels on to the stack.

Figure 6.6 only illustrates the first operation. When LSR B receives a packet from LSR A, it will replace the incoming label 62 with a new outgoing label 15, using the first

Label (20 bits)	Exp (3 bits)	S = 0	TTL (8 bits)
Label (20 bits)	Exp (3 bits)	S = 0	TTL (8 bits)
• • •			
Label (20 bits)	Exp (3 bits)	S = 1	TTL (8 bits)

Figure 6.7 The label stack.

operation. The same happens at the LSR D. The other two operations will be described below, where we discuss the use of the label stack.

In the case where the next hop of an LSR is the LSR itself, the LSR pops the top level label, and the resulting packet is forwarded based on whatever remains after the label stack was popped. This might still be a labeled packet or it might be a native IP packet that has to be forwarded based on its prefix. In the example in Figure 6.6, LSR E will pop the stack, and then will forward the native IP packet using its prefix.

The *incoming label map (ILM)* maps an incoming label to a set of NHLFEs associated with the incoming label. Having multiple entries for each incoming label can be useful because that allows multi-pathing for load balance and protection to be implemented. The procedure for choosing one of the NHLFE entries is beyond the scope of the MPLS architecture.

Finally, there is the *FEC-to-NHLFE map (FTN)*, which is used to map a FEC to a set of NHLFEs. This is used when a packet arrives unlabeled, and it has to be labeled before it is forwarded. As in the case of the ILM, if the FTN maps a FEC to multiple NHLFEs, a procedure is required to select one of them. Such a procedure is not defined in the MPLS architecture. In the example in Figure 6.6, LSRs A and C make use of the FTN to determine the appropriate entry from where the outgoing label and the outgoing interface can be obtained.

6.2.3 Explicit Routing

An IP router makes a forwarding decision by using the destination IP address of a packet in its FIB in order to determine the next hop IP router. When using a link-state protocol such as OSPF, each IP router learns about the topology of its domain by exchanging information with the other IP routers. It then calculates the next hop IP router for each destination using the shortest path algorithm. This next hop is stored in its FIB. MPLS uses the same next hop information in order to set up an LSP. In view of this, this type of routing is known *hop-by-hop* routing. For instance, in the example in Figure 6.6, the LSP between LSRs A and E was chosen using the next hop information in each LSR.

In addition to the hop-by-hop LSPs the MPLS architecture permits the creation of an LSP that follows an explicit route through a network which might not necessarily correspond to the hop-by-hop path. This type of routing is referred to as *explicit routing*. An explicitly routed LSP in MPLS is the equivalent of a point-to-point connection in ATM networks. An explicit route might be set up to satisfy a QoS criterion, such as minimizing the total end-to-end delay and maximizing throughput. Such a QoS criterion might not be necessarily satisfied by the hop-by-hop routing, which in general strives to minimize the number of hops only. Also, explicit routing can be used to provide load-balancing, by forcing some of the traffic to follow different paths through a network, so that the utilization of the network links is as even as possible. Finally, explicit routing can be used to set up MPLS-based *tunnels* and *virtual private networks (VPN)*.

An explicit route can be *strictly explicitly routed* or *loosely explicitly routed*. In the strictly explicitly routed case, the path for the ingress LSR to the egress LSR is defined precisely. That is, all of the LSRs through which the path will pass are explicitly specified. In the loosely explicitly routed case, not all of the LSRs through which the path will pass are specified. For instance, if a path has to go through several domains, the actual path through a domain might not be defined precisely. In this case, the MPLS edge LSR will calculate a path through its domain.

The strictly explicitly and loosely explicitly schemes are also used in PNNI. For instance, if an ingress ATM switch wants to establish an ATM connection to an egress ATM switch that belongs to the same peer group, then it will specify all of the ATM switches along the path. This is similar to the strictly explicitly routed scheme. However, if the egress ATM switch belongs to a different peer group, then the ingress ATM switch will specify all of the ATM switches along the path through its peer group, and then it will give a sequence of logical group nodes to be transited. When the setup message arrives at a logical group node, the node itself is responsible for calculating the path across its peer group. This is similar to the loosely explicitly route scheme.

6.2.4 An Example of the Use of the Label Stack

An example of the use of the label stack is shown on Figure 6.8. There are three MPLS domains (A, B, and C), and an explicit route has been established between LSR 1 in the MPLS domain A and LSR 6 in the MPLS domain C.

The label stack on each hop and the label operation carried out at each LSR along the path is also shown in Figure 6.8. The outgoing label from LSR 1 to LSR 2 is 60. (For simplicity, we are not concerned how LSR 1 assigns this label, and from where the IP packets originate).The label operation at LSR 2 is: replace the label at the top of the packet's label stack with a new label, and then push one new label on to the stack. As a result of this operation, label 60 is replaced by 70, and a new label with the value 40 is pushed on top. From LSR 3 to LSR 4, the packet is forwarded using the operation:

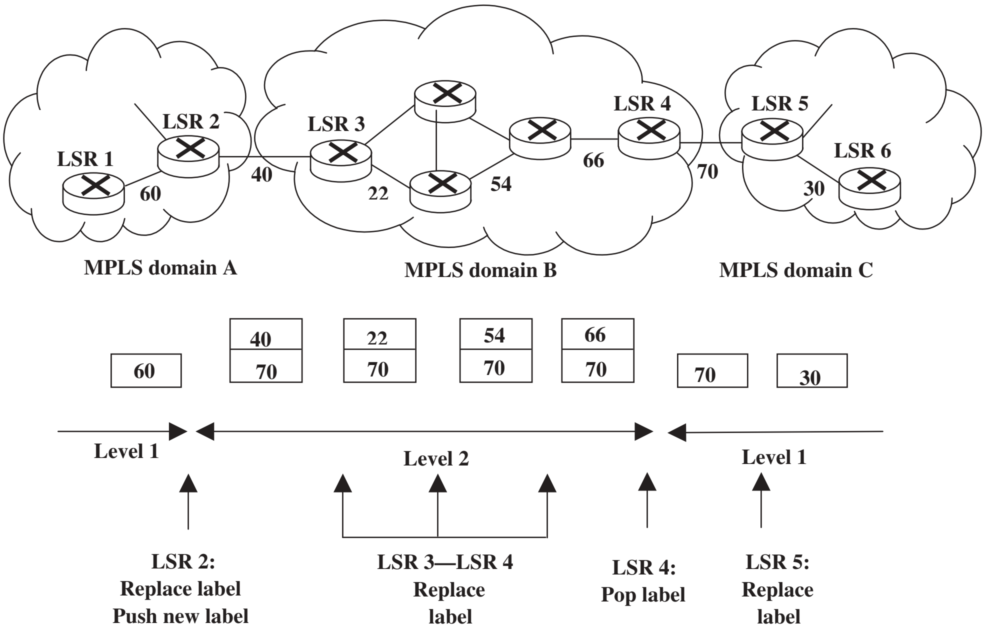


Figure 6.8 An example of the use of the label stack.

replace the label at the top on the packet's label stack with a new label. As a result, the top label of the label stack is first replaced with the value 22, then 54, and subsequently 66. At LSR 4, the label operation is: pop the label stack. As a result, the top label 66 is removed from the label stack, and now the packet is forwarded to LSR 5 using the label 70. Finally, LSR 5 forwards the packet to LSR 6 using the label operation: replace the label at the top on the packet's label stack with a new label. As a result, the packet arrives at LSR 6 with a label 30.

As we can see, when a packet is forwarded within the MPLS domain B, it contains two labels. The top label is used for label switching within the MPLS domain B, and the bottom label is used between the two edge nodes connecting the MPLS domains B and C.

This use of the label stack permits the establishment of LSP tunnels. For instance, we can think that the path between LSRs 3 and 4 through the MPLS domain B, defined by the labels 22, 54, and 66, is a tunnel that interconnects LSRs 2 and 5. For LSR 2 to forward packets to this tunnel, it has to use the label 40. At the other side of the tunnel, in order for the packets to be forwarded to LSR 6, the incoming label to LSR 5 has to have the value 70, so that LSR 5 can switch it out to LSR 6. This label is carried in the bottom label of the label stack.

6.2.5 Schemes for Setting up an LSP

In the MPLS architecture it is possible to force an LSP to be set up through LSRs in a particular order. Specifically, the following two schemes can be used to set up an LSP: *independent LSP control*, and *ordered LSP control*. In the independent LSP control scheme, each LSR binds a label to a FEC and advertises the binding to its neighbors as soon as it recognizes a new FEC. In the ordered control case, the allocation of labels proceeds backwards from the egress LSR. The following rules are used: an LSR only binds a label to a FEC if it is the egress LSR for that FEC, or if it has already received a label binding for that FEC from its next hop LSR.

In the example shown in Figure 6.6, we can assume that the independent LSP control scheme was used. That is, each LSR independently from the other LSRs binds a label to the FEC ⟨x.0.0.0,y.0.0.0⟩ and advertises the binding to its neighbors without having to wait to hear from its next hop LSR for that FEC. The ordered LSP control scheme is used to set up an explicit route, as will be seen later on in this chapter.

Within an MPLS domain, it is possible that IP packets belonging to two or more different FECs follow the same route. This can happen when these FECs have the same egress node. In this case, it is possible to aggregate these FECs in to one or more FECs, or to not aggregate them at all and simply keep them separate.

6.3 MPLS OVER ATM

MPLS was defined to run over different networking schemes including ATM and frame relay. In the case of MPLS over ATM, the signaling protocols that typically run on an ATM switch, such as Q.2931 and PNNI, are replaced by IP protocols such as OSPF, BGP, PIM and RSVP. The ATM switch is used simply as a cell-switching device.

In an ATM network, a connection is set up using Q.2931 and PNNI. Using PNNI, an ATM switch becomes aware of the topology of its peer group and of the logical nodes in the higher-level peer groups. When a calling user wishes to establish a connection to a destination user, it sends a SETUP message to its ingress ATM switch using the Q.2931 signaling protocol. The ingress ATM switch calculates a path through the ATM network, and then using the PNNI protocol it forwards the SETUP message to the next switch along the path, which forwards it to the next ATM switch on the path, and so on, until the SETUP message reaches the egress ATM switch which serves the called user. The egress switch forwards the SETUP message to the called user using the Q.2931 signaling protocol, and if the called user accepts it, a confirmation is returned back to the calling user. At that time, the calling user can start transmitting data to the called user.

In MPLS over ATM, this entire signaling functionality is removed from the ATM switches. Instead, each ATM switch is identified by an IP address and runs IP routing protocols. Such an ATM switch is referred to as an ATM-LSR. As in an IP router, using IP routing protocols an ATM-LSR can learn about its neighbors and about the topology of its IP domain, and it can calculate the next hop ATM-LSR for each IP destination.

In MPLS over ATM, an LSP is nothing else but an ATM connection which is set up using MPLS. The MPLS label is carried in the VPI/VCI field of the cell. If a label stack is used, then only two labels can be carried. The top label is carried in the VPI field and the bottom one in the VCI field. The advertising of label bindings is done using downstream allocation on demand. That is, when an ATM-LSR identifies a new FEC it allocates a label, but it does not advertise it to its neighbors. An upstream ATM-LSR obtains the label binding by sending a request. A predefined ATM VC connection is used for exchanging label binding information.

Let us consider now a network of ATM-LSRs. An IP packet at the ingress ATM-LSR, is first encapsulated into a CPS-PDU using AAL 5. Then, it is segmented into an integer number of 48-byte blocks and each block is carried in a different ATM cell. The label associated with the particular LSP is carried in the VPI/VCI field of the cell. When an ATM cell reaches the next hop ATM-LSR, its label is replaced by the outgoing label and the cell is switched to the appropriate output from where it is transmitted out to the next ATM-LSR. This continues until the cell reaches the egress ATM-LSR. There, the cell is assembled with other cells into the original AAL 5 CSC-PDU, and the IP packet is recovered from its payload and is delivered to the IP protocol. Therefore, an IP packet traverses the network of ATM-LSRs in a sequence of ATM cells that are switched through each ATM-LSR without ever having to reconstruct the original IP packet at each intermediary ATM-LSR, except at the egress ATM-LSR. This is very similar to the IP switching scheme.

6.3.1 VC Merging

An interesting problem that arises in label switching over ATM is *VC merging*. This problem arises when two ATM-LSRs are both connected to the same downstream ATM-LSR. Let us consider the four ATM-LSRs (A, B, C, and D) in Figure 6.9, and let us assume that the flow of IP packets for a specific FEC is from A to C and then to D, and from B to C and then to D. Assume that D is the edge LSR that serves the hosts associated with the FEC. (The allocated labels are shown in Figure 6.9 in bold.) Now let us see what happens when A has an IP packet (call it packet 1) to transmit that belongs to this FEC. This packet will be encapsulated by AAL 5 and then it will be segmented into an integer number of 48-byte blocks. Each block will then be carried in the payload of an

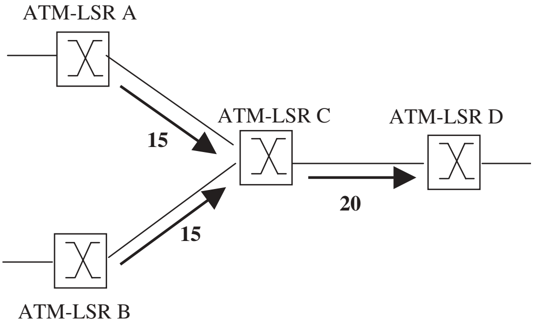


Figure 6.9 VC merging.

ATM cell labeled with the value 15. The cells will be transmitted to C, where they will have their label changed to 20 and then they will be forwarded to the buffer of the output port that connects to D. In this buffer, it is possible that these cells will get interleaved with the cells belonging to an IP packet, call it packet 2, associated with same FEC and transmitted from B. That is, as the cells are queued-up into the buffer, packet 1 cells can find themselves between two successive packet 2 cells. Since all of these cells will be sent to D with the label of 20, D will not be able to identify which of these cells belong to packet 1 or packet 2. Consequently, D will be not be able to reconstruct the original AAL 5 PDUs.

A simple solution to this problem is to first collect all of the cells belonging to the same IP packet in the buffer of the output port of C. Once all of the cells have arrived, then they can be transmitted out back-to-back to D. To do this, the switch will need to be able to identify the beginning cell and last cell of an AAL 5 PDU. If the switch is set up with the early-packet and partial-packet discard policies, then the mechanism to do this might be in place. Otherwise, multiple labels can be used, so that the path from A to D is associated with a different set of labels than the path from B to D.

6.3.2 Hybrid ATM Switches

It was mentioned above that in MPLS over ATM, the entire ATM signaling functionality is removed from the ATM switches. Instead, each ATM switch runs the MPLS control plane. That is, the MPLS protocol, a label distribution protocol, and IP routing protocols. However, the existence of MPLS on an ATM switch does not preclude the switch from running the ATM signaling protocol. In fact, both schemes can coexist on the same ATM switch and on the same ATM interface.

PROBLEMS

1. Make-up an arbitrary IP header and calculate its checksum. Introduce errors in the bit stream (i.e. flip single bits) so that the checksum when calculated by the receiver will fail to detect that the IP header has been received in error.
2. Consider the IP address: 152.1.213.156.
   a) What class address is it?
   b) What is the net and host address in binary?

3. A class B IP network has a subnet mask of 255.255.0.0.
   a) What is the maximum number of subnets that can be defined?
   b) What is the maximum number of hosts that can be defined per subnet?

4. Consider the label allocations in Table 6.3. How would the labels change if we unplugged the link between C and D?

5. The MPLS architecture permits an LSR to maintain multiple entries for each incoming label. Give an example where this feature can be used.

6. Describe the difference between an explicit route and a hop-by-hop route. Under what conditions an explicit route between an ingress LSR and an egress LSR coincides with the hop-by-hop route?

7. In MPLS, label allocation can be done using either the unsolicited downstream scheme or the downstream on demand scheme. In Cisco's tag switching, *upstream tag allocation* was also possible. Describe how this scheme works. (Hint: it is the opposite of the unsolicited downstream scheme.)

8. Consider the example of the LSP shown in Figure 6.8. Let us assume that a second LSP has to be set up from LSRs 1 to 6 over the tunnel connecting LSRs 3 and 4. Show the label stack on each hop and the label operation that has to be performed at each LSR.

7

Label Distribution Protocols

MPLS requires a set of procedures for the reliable distribution of label bindings between LSRs. MPLS does not require the use of a single label distribution protocol. In view of this, various schemes have been proposed for the distribution of labels, of which the *label distribution protocol (LDP)* and the *resource reservation protocol – traffic engineering (RSVP–TE)* are the most popular.

LDP is a new signaling protocol. It is used to distribute label bindings for an LSP associated with a FEC. It has been extended to the *constraint-based routing label distribution protocol (CR-LDP)*, which is used to set up an *explicit route* (i.e. an LSP between two LSRs). LDP and CR-LDP are described below in Sections 7.1 and 7.2.

An alternative method to distributing label bindings is to extend an existing IP control protocol, such as BGP, PIM, and RSVP, so that it can carry label bindings. The extended version of RSVP is known as RSVP-TE, and is the most popular protocol of the above three for distributing label bindings. RSVP-TE has functionality for setting up LSPs using the next hop information in the routing table of an LSR and explicitly routed LSPs. RSVP and RSVP-TE are described in Sections 7.3 and 7.4.

Typically, an LSR will run both LDP and RSVP-TE. The two label distribution protocols are not compatible, however. In order to establish an LSP, either LDP or RSVP-TE has to be used.

7.1 THE LABEL DISTRIBUTION PROTOCOL (LDP)

LDP is used to establish and maintain label bindings for an LSP associated with a FEC. Two LSRs that use LDP to exchange label bindings are known as *LDP peers*. LDP provides several LDP messages, which are classified as follows:

- *Discovery messages*: These messages are used to announce and maintain the presence of an LSR in the network.
- *Session messages*: In order for two LDP peers to exchange information, they have to first establish an *LDP session*. The session messages are used to establish, maintain, and terminate LDP sessions between LDP peers.
- *Advertisement messages*: These messages are used to create, change, and delete label bindings to FECs.
- *Notification messages*: These messages are used to provide advisory information and to signal error information.

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

LDP runs on top of TCP for reliability, with the exception of the LDP discovery messages that run over UDP.

Before we proceed to describe the LDP messages and their format, we divert to discuss in the following section several LDP concepts, such as the *per platform* and *per interface label space, LDP session*, and *hello adjacencies*. (The following section can be skipped in the first reading.)

7.1.1 Label Spaces, LDP Sessions, and Hello Adjacencies

LDP makes use of the concept of the label space, which is the set of all labels. Two types of label space have been defined: *per interface label space* and *per platform label space*. The per interface label space is a set of labels which are specific to a particular interface. For instance, an ATM interface uses VPI/VCI numbers which are specific to the interface. Likewise, a frame relay interface uses DLCI values which are also specific to the interface. The per platform label space is a set of labels shared by all interfaces other than ATM and frame relay interfaces, such as *packet-over-SONET (PoS)* and *gigabit Ethernet (GbE)*. When transmitting packets over these links, the MPLS labels are carried in the special label encapsulation (see Figure 6.3). These labels are processed by the same MPLS software, and they all belong to the same per-platform label space.

An LSR label space is identified by a 6-byte value. The first four bytes carry a globally unique value identifying the LSR, such as a 32-bit router id that has been assigned to the LSR by the autonomous system administrator. The last two bytes identify the label space. If the label space is per platform, then the last two bytes are set to 0. A label space id, formally referred to as an *LDP id*, is expressed as ⟨LSRDid, label space number⟩. An example of label space ids is given in Figure 7.1. The LSR id number for LSR A is *lsr170*. LSR A is connected to LSRs B, C, D, E, and F. It connects to B via the A GbE interface. It connects to C via a GbE interface, and also via a separate PoS interface. It connects to D via an ATM interface. It connects to E via two separate ATM interfaces. It connects to F via a GbE interface and a separate ATM interface. LSR A advertises the per-platform label space id ⟨lsr170,0⟩ to B and C. It advertises the non-zero label space ⟨lsr170,1⟩ to D. It advertises the non-zero label space id ⟨lsr170,2⟩ over the first ATM link and ⟨lsr170,3⟩ over the second ATM link to E. It advertises the non-zero label space ⟨lsr170,4⟩ over the ATM link and the per-platform label space id ⟨lsr170,0⟩ over the GbE interface to F. In summary, the label space ids that A advertises to its

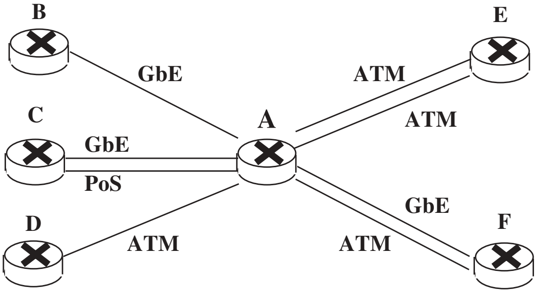


Figure 7.1 An example of label space ids.

neighbors are:

- ⟨lsr170,0⟩ for LSR B, LSR C (both interfaces), and LSR F (GbE interface)
- ⟨lsr170,1⟩ for LSR D (for the ATM interface)
- ⟨lsr170,2⟩ for LSR E (for the first ATM interface)
- ⟨lsr170,3⟩ for LSR E (for the second ATM interface)
- ⟨lsr170,4⟩ for LSR F (for the ATM interface)

An LDP session is set up between two directly connected LSRs to support the exchange of LDP messages between them. An LDP session between two LSRs is associated with a label space. For the example given above, the following sessions are set up:

- A-B: 1 LDP session for ⟨lsr170,0⟩
- A-C: 1 LDP session for ⟨lsr170,0⟩
- A-D: 1 LDP session for ⟨lsr170,1⟩
- A-E: 2 LDP sessions; one for ⟨lsr170,2⟩ and one for ⟨lsr170,3⟩
- A-F: 2 LDP sessions; one for ⟨lsr170,0⟩ and one for ⟨lsr170,4⟩

It is also possible to set up an LDP session between two non-directly connected LSRs. This can be useful when two distant LSRs might want to communicate via an LSP. The two LSRs can establish a session in order to communicate a label binding. This label can be pushed down the label stack as in the example discussed 6.2.4.

An LDP discovery mechanism enables an LSR to discover potential LDP peers (i.e., other LSRs which are directly connected to it). An LSR sends periodically *LDP link hellos* out of each interface. Hello packets are sent over UDP addressed to a well-known LDP discovery port for the "all routers on this subnet" group multicast address. An LDP link hello sent by an LSR carries the label space id that the LSR wants to use for the interface and possibly additional information. Receipt of an LDP link hello identifies a *hello adjacency*. For each interface, there is one hello adjacency.

An extended discovery mechanism can be used for non-directly connected LSRs. An LSR periodically sends *LDP targeted hellos* to a specific address over UDP. Receipt of an LDP targeted hello identifies a hello adjacency.

The exchange of LDP link hellos between two LSRs triggers the establishment of an LDP session. If there is a single link between two LSRs, then a single hello adjacency and a single LDP session are set up. If there are parallel links with per platform label space, then there are as many hello adjacencies as the number of links, but only one LDP session. If there are parallel links, one with per platform label space and the others per interface label space, then one session per interface and one adjacency per session is set up. For the example in Figure 7.1, the following sessions and hello adjacencies are set up:

- A-B: One LDP session with one hello adjacency
- A-D: One LDP session with one hello adjacency
- A-C: One LDP session with two hello adjacencies
- A-E: Two LDP sessions, each associated with one hello adjacency
- A-F: Two LDP sessions, each associated with one hello adjacency

The establishment of an LDP session involves two steps. First, a TCP session is established. Second, an LDP session is initialized, during which the two LSRs negotiate session parameters (such as protocol version, label distribution method, timer values, range of VPI/VCI values for ATM, and range of DLCI values for frame relay).

An LSR maintains a timer for each hello adjacency, which it restarts each time it receives a hello message. If the timer expires without having received a hello message from the peer LSR, the hello adjacency is deleted. The LDP session is terminated, if all hello adjacencies associated with an LDP session are deleted.

An LSR maintains a *keepAlive* timer for each session. The timer is reset each time it receives any LSP PDU from its LDP peer. If the LDP peer has nothing to send, it sends a keepAlive message.

7.1.2 The LDP PDU Format

An *LDP PDU* consists of an LDP header followed by one or more LDP messages which might not be related to each other. The LDP PDU format is shown in Figure 7.2. The LDP header consists of the fields:

- *Version*: A 16-bit field that contains the protocol version.
- *PDU length*: A 16-bit field that gives the total length of the LDP PDU in bytes, excluding the version and PDU length fields of the LDP PDU header.
- *LDP id*: A 48-bit field that contain the LDP id (that is, the label space id) which has the form ⟨32-bit router id, label space number⟩.

The LDP message format consists of a header followed by mandatory and optional parameters. The header and the parameters are all encoded using the *type-length-value (TLV)* scheme shown in Figure 7.3. The following fields have been defined:

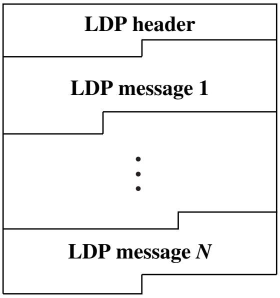


Figure 7.2 The LDP PDU format.

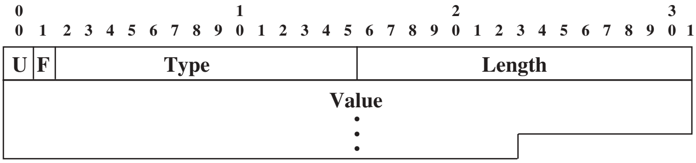


Figure 7.3 The TLV format.

- *U (Unknown TLV bit)*: Used when an unknown TLV is received. If U = 0, then a notification is returned to the message originator and the entire message is ignored. If U = 1, then the TLV is silently ignored and the rest of the message is processed as if the TLV did not exist.
- *F (Forward unknown TLV bit)*: This bit applies only when U = 1, and the LDP message containing the unknown TLV has to be forwarded. If F = 0, then the unknown TLV is not forwarded with the rest of the message. If F = 1, then the unknown TLV is forwarded with the rest of the message.
- *Type*: A 14-bit field that describes how the Value field is to be interpreted.
- *Length*: A 16-bit field that gives the length of the Value field in bytes.
- *Value*: Contains information that is interpreted as specified in the Type field. It might contain TLV encoding itself.

7.1.3 The LDP Message Format

The LDP message format is shown in Figure 7.4. The following fields have been defined:

- *U (Unknown message bit)*: Upon receipt of an unknown message, if U = 0, then a notification is returned to the message originator. If U = 1, then the unknown message is silently ignored.
- *Message type*: A 15-bit field that is used to identify the type of message.
- *Message length*: A 16-bit field that gives the total length (in bytes) of the message ID field and the mandatory and optional parameters fields.
- *Message ID*: A 32-bit value used to identify this message. Subsequent messages related to this one have to carry the same message ID.

The mandatory fields will be discussed separately for each individual LDP message.

7.1.4 The LDP Messages

The following LDP messages have been defined: *notification, hello, initialization, keepAlive, address, address withdraw, label mapping, label request, label abort request, label withdraw*, and *label release*.

Notification message

This message is used to inform an LDP peer of a fatal error, or to provide advisory information regarding the outcome of processing an LDP message or the state of an LDP

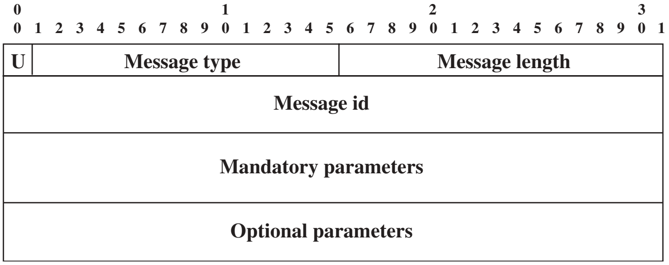


Figure 7.4 The LDP message format.

session. Some of the notification messages are:

- Malformed PDU or message
- Unknown or malformed TLV
- Session keepAlive timer expiration
- Unilateral session shutdown
- Initialization message events
- Events resulting from other errors

Hello message

LDP hello messages are exchanged as part of the LDP discovery mechanism. The format of the hello message is shown in Figure 7.4, with the U bit set to 0, and the message type set to hello (0x0100). The mandatory parameters field, referred to as *the common hello parameters TLV*, is shown in Figure 7.5. The following fields have been defined for the common hello parameters TLV:

- *Hold time*: Specifies the *hello hold time* in seconds. This is the time that the sending LSR will maintain its record of hellos from the receiving LSR without receipt of another hello. If hold time = 0, then the default value – 15 sec for link hellos or 45 sec for targeted hellos – is used. A value of 0xffff means infinite. The hold timer is reset each time a hello message is received. If it expires before a hello message is received, then the hello adjacency is deleted.
- T: Specifies the hello type: a targeted hello ($T = 1$) or a link hello ($T = 0$).
- R: This field is known as the *request send targeted hellos*. A value of 1 indicates that the receiver is requested to send periodic targeted hellos to the source of this hello. A value of 0 makes no such request.

Initialization message

This message is used to request the establishment of an LDP session. The format of the initialization message is shown in Figure 7.4, with the U bit set to 0, and the message type set to initialization (0x0200). The format for the mandatory parameters field, referred to as the *common session parameters TLV*, is shown in Figure 7.6. The following fields have been defined:

- *KeepAlive time*: Indicates the maximum number of seconds that can elapse between the receipt of two successive LDP PDUs. The keepAlive timer is reset each time an LDP PDU is received.
- *A*: Indicates the type of label advertisement: downstream unsolicited (A = 0) or downstream on demand (A = 1). Downstream on demand is used only for an ATM or a frame relay link. Otherwise, downstream unsolicited must be used.

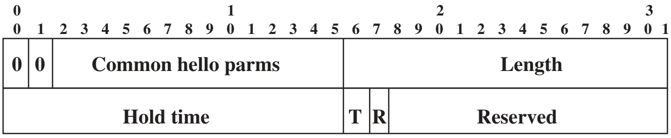


Figure 7.5 Common hello parameters TLV.

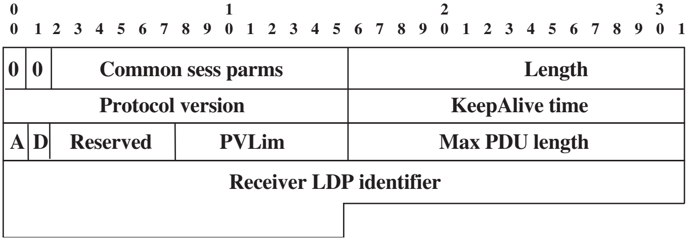


Figure 7.6 Common session parameters TLV.

- *D*: Enables loop detection.
- *PVLim (Path vector limit)*: Gives the maximum number of LSRs recorded in the path vector used for loop detection.
- *Max PDU length*: Default value of the maximum allowable length is 4096 bytes.
- *Receiver LDP identifier*: Identifies the receiver's label space.

The optional session parameters field can be used to provide for ATM and frame relay session parameters.

KeepAlive message

An LSR sends keepAlive messages as part of the a mechanism that monitors the integrity of an LDP session. The format of the keepAlive message is shown in Figure 7.4, with the U bit set to 0, and the message type set to keepALive (0x0201). No mandatory or optional parameters are provided.

Address and address withdraw messages

Before sending a label mapping and a label request messages, an LSR advertises its interface addresses using the address messages. Previously advertised addresses can be withdrawn using the address withdraw message.

Label mapping message

An LSR uses the message to advertise a mapping (i.e., a binding) of a label to a FEC to its LDP peers. The format of the label mapping message has the same structure as the one shown in Figure 7.4, with the U bit set to 0 and the message type set to label mapping (0x0400). The mandatory parameters field consists of a FEC TLV and a label TLV.

In LDP a FEC element could be either a prefix of an IP address or it could be the full IP address of a destination host. The FEC TLV is shown in Figure 7.7. LDP permits a FEC to be specified by a set of *FEC elements*, with each FEC element identifying a set of packets that can be mapped to the corresponding LSP. (This can be useful, for instance, when an LSP is shared by multiple FEC destinations all sharing the same path).

The label TLV gives the label associated with the FEC given in the FEC TLV. This label could be a 20-bit label value, or a VPI/VCI value in the case of ATM, or a DLCI

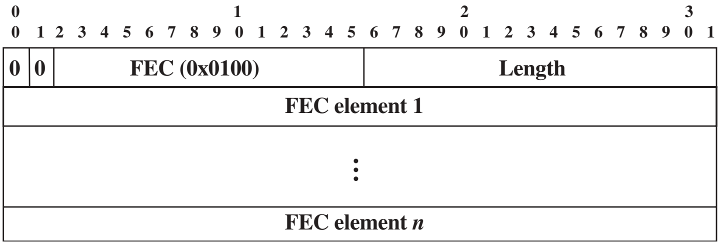


Figure 7.7 The FEC TLV.

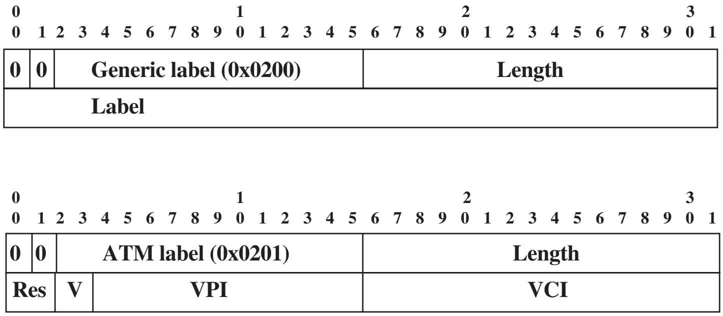


Figure 7.8 The generic label and ATM label TLVs.

value in the case of frame relay. The generic label and ATM label TLVs are shown in Figure 7.8. The 2-bit V field in the ATM label TLV, referred to as the *V-bits*, is used as follows. If the V-bits is 00, then both the VPI and the VCI fields are significant, If the V-bits is 10, then only the VCI is significant.

Label request message

An LSR sends a label request message to an LPD peer to request a mapping to particular FEC. The label request message has the format shown in Figure 7.4, with the U bit set to 0, and the message type set to label request (0x0401). The mandatory parameters field contains the FEC TLV shown in Figure 7.7.

An LSR can transmit a label request message under the following conditions:

- The LSR recognizes a new FEC via its forwarding routing table; the next hop is an LDP peer; and the LSR does not already have a mapping from the next hop for the given FEC.
- The next hop to the FEC changes, and the LSR does not already have a mapping from the next hop for the given FEC.
- The LSR receives a label request for a FEC from an upstream LDP peer; the FEC next hop is an LDP peer; and the LSR does not already have a mapping from the next hop.

Label abort, label withdraw, and label release messages

An LSR A can send a *label abort* message to an LDP peer LSR B to abort an outstanding label request message. This might happen, for instance, if LSR A's next hop for the FEC has changed from LSR B to a different LSR.

An LSR A uses a *label withdraw* message to signal to an LDP peer LSR B that it cannot continue using a specific FEC-label mapping that LSR A had previously advertised.

An LSR A sends a *label release* message to an LDP peer LSR B to signal to LSR B that LSR A no longer needs a specific FEC-label mapping that was previously requested of and/or advertised by the peer.

7.2 THE CONSTRAINED-BASED ROUTING LABEL DISTRIBUTION PROTOCOL (CR-LDP)

CR-LDP is a label distribution protocol based on LDP. As described above, LDP can be used to set up an LSP associated with a particular FEC. CR-LDP is used to set up a unidirectional point-to-point explicitly routed LSP, referred to as the *constrained-based routed label switched path* (*CR-LSP).*

An LSP is set up as a result of the routing information in an IP network using the shortest path algorithm. A CR-LSP is calculated at the source LSR based on criteria not limited to routing information, such as explicit routing and QoS-based routing. The route then signaled to the other nodes along the path which obey the source's routing instructions. This routing technique, referred to as *source routing*, is also used in ATM.

A CR-LSP in MPLS is analogous to a connection in ATM, only it is unidirectional. The ATM signaling procedures will automatically set up a bidirectional connection between two ATM hosts, where each direction of the connection can be associated with different traffic and QoS parameters. A bidirectional CR-LSP between LSRs 1 and 2 can only be created by setting up one CR-LSP from LSR 1 to LSR 2 and a separate one from LSR 2 to LSR 1. As in the case of an LSP, a CR-LSP has an ingress and an egress LSR.

CR-LSPs can be used in a variety of ways. For instance, they can be used in an IP network to do load balancing. That is, the traffic among its links can be evenly distributed by forcing some of the traffic over CR-LSPs, which pass through lesser-utilized links. CR-LSPs can also be used to create tunnels in MPLS, and introduce routes based on a QoS criterion, such as minimization of the total end-to-end delay, and maximization of throughput. For example, let us consider the MPLS network in Figure 7.9, and let us

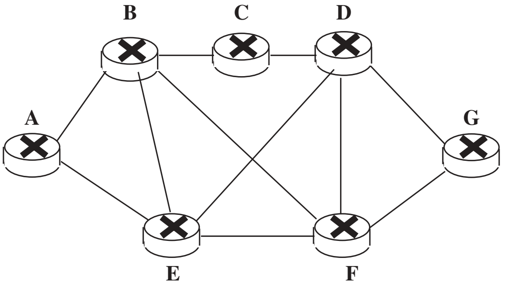


Figure 7.9 An example of a CR-LSP.

assume that the path between the ingress LSR A and the egress LSR G, calculated using OSPF, passes through E and F. Using CR-LDP we can set up a CR-LSP that satisfies a QoS criterion, such as minimize the end-to-end delay. For instance, if LSRs B, C, and D are not heavily utilized, routing the CR-LSP through these LSRs will reduce the end-to-end delay, even though the number of hops will be higher than the E-to-F path.

The following are some of the features of CR-LDP:

- CR-LDP is based on LDP, and runs on top of TCP for reliability.
- The CR-LDP state-machine does not require periodic refreshment.
- CR-LDP permits strict and loose explicit routes. This allows the ingress LSR some degree of imperfect knowledge about the network topology (see Section 6.2.3). The source LSR might also request *route pinning*, which fixes the path through a loosely defined route so that it does not change when a better next hop becomes available.
- CR-LDP permits path preemption by assigning setup/holding priorities to CR-LSPs. If a route for a high-priority CR-LSP cannot be found, then existing lower-priority CR-LSPs can be rerouted to permit the higher-priority CR-LSP to be established.
- The network operator can classify network resources in various ways. CR-LDP permits the indication of the resource classes that can be used when a CR-LSP is being established.
- As in the case of ATM, CR-LDP allows the specification of traffic parameters on a CR-LSP and how these parameters should be policed.

CR-LDP depends on the following minimal LDP functionality:

- Basic and/or extended discovery mechanism
- Label request message for downstream on demand with ordered control
- Label mapping message for downstream on demand with ordered control
- Notification messages
- Label withdraw and release messages
- Loop detection for loosely routed segments

7.2.1 CR-LSP Setup Procedure

A CR-LSP is set up using downstream on demand allocation with ordered control. Recall that in the downstream on demand allocation scheme, each LSR binds an incoming label to a FEC and creates an appropriate entry in its LFIB. However, it does not advertise its label mapping to its neighbors as in the unsolicited downstream allocation scheme. Instead, an upstream LSR obtains the label mapping by issuing a request. In the ordered control scheme, the allocation of labels proceeds backwards from the egress LSR towards the ingress LSR. Specifically, an LSR only binds a label to a FEC if it is the egress LSR for that FEC, or if it has already received a label binding for that FEC from its next hop LSR.

An example of how a CR-LSP is set up is shown in Figure 7.10. Let us assume that LSR A has been requested to establish a CR-LSP to LSR E. A request to set up a CR-LSP to LSR E might originate from a management system or an application. LSR A calculates the explicit route using information provided by the management system, or the application, or from a routing table, and creates the label request message. The explicit

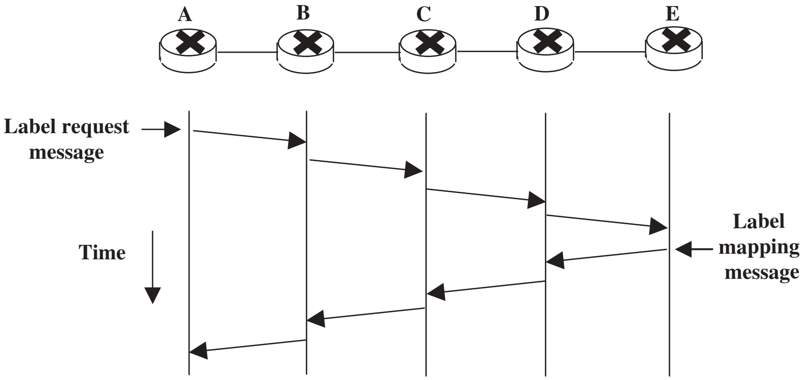


Figure 7.10 An example of a CR-LSP setup.

route in this case is given by the series of LSRs B, C, and D. It is carried in a special TLV in the label request message called the *explicit route TLV (ER-TLV)*.

The ingress LSR A sends the label request message to LSR B, the first LSR indicated in the ER-TLV, requesting a label mapping for the FEC associated with the CR-LSP. Because of the ordered control scheme, LSR B cannot create a label mapping to the FEC until it has received a label mapping from its next hop LSR C. Also, because of the downstream on demand allocation scheme, LSR C does not advertise its label mappings to its neighbors. In view of this, LSR B forwards the label request message to LSR C requesting a label mapping for the FEC. LSR C forwards the label mapping request to LSR D for the same reasons, which then forwards the label request message to the egress LSR E. The egress LSR E is now allowed to create a label mapping to the FEC. It does so, and it responds to LSR D's label request message with a label mapping message that contains the allocated label. When LSR D receives the label mapping message form LSR E it responds to LSR C's label request message with a label mapping message that contains its own incoming label, and so on, until LSR A receives a label mapping message form LSR B. At that time, the CR-LSP has been set up.

Next, the label request message and the label mapping message are described.

7.2.2 The Label Request Message

The label request message is shown in Figure 7.11. The U bit is set to 0, and the message type set to label request (0x0401). The FEC TLV (see Figure 7.7) must be included in the label request message, and it contains a single new FEC element, named CR-LSP.

The *LSPID TLV* is required and is used to give a unique identifier of a CR-LSP. It is composed of the ingress LSR router id (or any of its IPv4 addresses) and a CR-LSP id which is locally unique to that LSR. The LSPID is useful in network management, in CR-LSP repair, and in using an already established CR-LSP as a hop in an ER-TLV.

The *explicit-route TLV (ER-TLV)*, shown in Figure 7.12, is used to specify the path to be taken by the LSP being established. It is composed of one or more *explicit route hop TLVs (ER-hop TLV)* which have the format shown in Figure 7.13. The type field indicates the type of the ER-hop contents, and it can take one of the following values: IPv4 prefix, IPv6 prefix, autonomous system number, LSPID. If an LSR receives a label

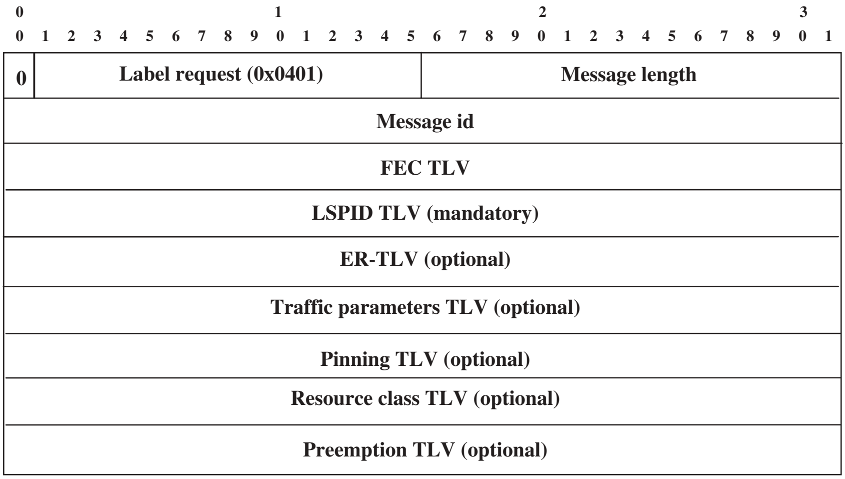


Figure 7.11 The CR-LDP label request message.

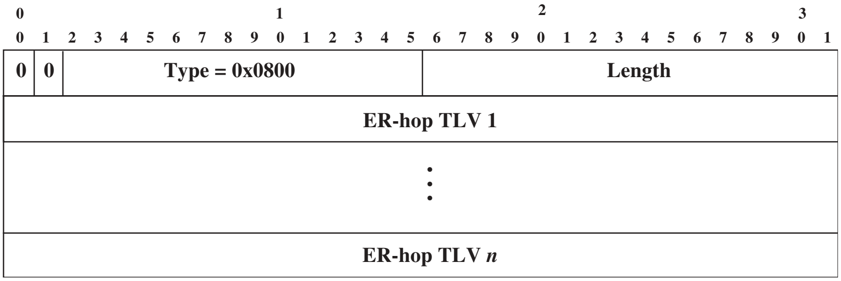


Figure 7.12 The ER-TLV.

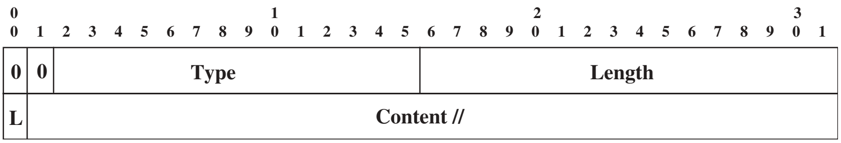


Figure 7.13 The ER-hop TLV.

request message containing an ER-hop TLV which it does not support, the LSR will not progress the label message to the next hop downstream LSR and it will send back a *no route* notification message. The L bit is used to indicate whether the ER-hop is loose or strict (see Section 6.2.3). The contents field contains a node or an abstract node representing a group of nodes.

Route pinning is applicable to segments of a CR-LSP that are loosely routed. It is signaled using the *route pinning TLV*. The resource classes that can be used to set up an CR-LSP are indicated in the *resource class TLV*.

The *preemption TLV* is used to assign a setup priority and a holding priority of the CR-LSP. These priorities are used to determine if the new CR-LSP can preempt an existing one. Assigning a higher holding priority means that the CR-LSP, once it has been set up, has a low chance of being preempted. Assigning a high setup priority means that, in the case that resources are unavailable, the CR-LSP has a high chance of preempting existing CR-LSPs.

The *traffic parameters TLV* is used to signal the values of the traffic parameters that characterize the CR-LSP that is being established. This TLV will be discussed in detail in Section 7.2.4.

7.2.3 The Label Mapping Message

The label mapping message is shown in Figure 7.14. The U bit is set to 0, and the message type set to label mapping (0x0400). The FEC and LSPID TLVs are the same as in the CR-LDP label request message. The label TLV is the same as in LDP (see Figure 7.8). The label request message id TLV is used as follows. If this label mapping message is a response to a label request message, then it must include the label request message id parameter. This parameter is carried in the label request message id TLV. The traffic parameters TLV is described in the next section.

7.2.4 The Traffic Parameters TLV

The traffic parameters TLV is used in the label request and label mapping messages. It is used to describe the traffic parameters of the CR-LSP that is being established. The traffic parameters TLV is shown in Figure 7.15. The traffic parameters TLV type is 0x0810, and the length of the value field is 24 bytes. The following fields have been defined: *flags, frequency, weight, peak data rate (PDR), peak burst size (PBS), committed data rate*

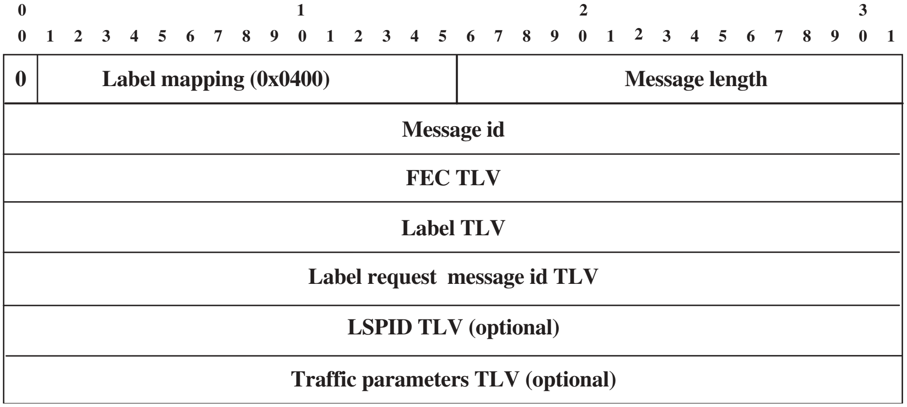


Figure 7.14 The label mapping message.

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td colspan="14">Type (0x0810)</td><td colspan="16">Length</td></tr>
<tr><td colspan="8">Flags</td><td colspan="8">Frequency</td><td colspan="8">Reserved</td><td colspan="8">Weight</td></tr>
<tr><td colspan="32">Peak data rate (PDR)</td></tr>
<tr><td colspan="32">Peak burst size (PBS)</td></tr>
<tr><td colspan="32">Committed data rate (CDR)</td></tr>
<tr><td colspan="32">Committed burst size (CBS)</td></tr>
<tr><td colspan="32">Excess burst size (EBS)</td></tr>
</table>

Figure 7.15 The traffic parameters TLV.

(CDR) , *committed burst size (CBS)* and *excess burst size (EBS)*. The PDR and the PBS parameters are used to define the traffic sent to the CR-LSP. The parameters CDR, CBS, and EBS, are used to specify how the network will police the traffic submitted to the CD-LSP. Finally, the flags, frequency, and weight fields are used to provide additional information as will seen below.

Peak data rate (PDR) and peak burst size (PBS)

The *peak rate* is the maximum rate at which traffic is sent to the CR-LDP, and is expressed in bytes/sec. The equivalent parameter in ATM is the *peak cell rate (PCR)*. Unlike the PCR, which is specified by a single value, the peak rate in CR-LDP is specified in terms of token bucket P. The maximum token bucket size of P is set equal to the *peak burst size (PBS)*, expressed in bytes, and the token bucket is replenished at the *peak data rate (PDR)*, expressed in bytes/sec. The PBS defines the maximum packet size that can be sent to the CR-LSP, and the PDR gives the maximum rate at which traffic is transmitted by the user. The peak rate, is the output from the token bucket P, which is sent to the CR-LSP.

The token bucket operates as follows:

- Initially, the token count (i.e., the number of tokens in the token bucket) is $T_P = PBS$.
- Let $PDR = N$ bytes/sec. Then, if $T_P \leq PBS$, the token count T_P is incremented every second by N (it should not exceed PBS).
- When a packet of size B bytes arrives, if $T_P - B \geq 0$, then the packet is not in excess of the peak rate, and $T_P = T_P - B$.
- Otherwise, it is in excess of the peak rate, and T_P is not decreased. The packet is a *violating packet* and it can be marked or dropped.

Note that a positive infinite value of either PBS or MBS implies that arriving packets are never in excess of the peak rate.

An example of the peak rate token bucket operation is given in Figure 7.16. The thick line at the top indicates the value of T_P, and the thin line below indicates the arrival of

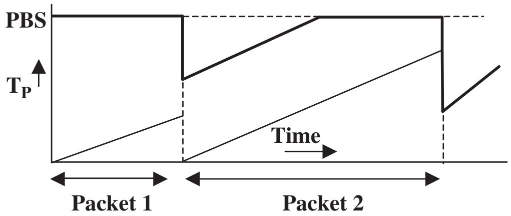


Figure 7.16 An example of the peak rate token buck operation.

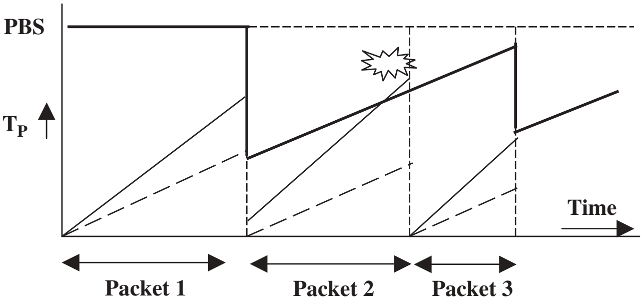


Figure 7.17 The transmission rate exceeds PDR.

a packet. The slope of all of the lines is equal to PDR. At time $t = 0$, the token bucket contains PBS tokens. Packet 1 begins to arrive at the rate of PDR bytes/sec. When it has completely arrived, the size of the packet expressed in bytes is deducted from the current token count T_P, the packet is allowed to go through, and the replenishment of the token bucket begins at the rate of PDR bytes/sec. Packet 2 begins to arrive immediately after packet 1. Since the size of packet 2 is less that the current token count, it is allowed to go through. In general, all of the packets will go through, as long as the user transmits at a rate which is less or equal to PDR.

Consider the case where the user's transmission rate temporarily exceeds PDR (see Figure 7.17). The dashed lines indicate the transmission rate of a packet if the source transmitted at the rate of PDR. The thin continuous lines (above the dashed lines) indicate the rate at which the packets are transmitted, and the thick line in the top of the diagram shows the current token count. Note that packet 1, although it arrives faster than it should, goes through. The token count is reduced appropriately, and the replenishment of the token bucket at the PDR rate begins. Packet 2 is not as lucky, and it will be either marked and let into the network or dropped. Packet 3 goes through.

Therefore, if the transmission rate temporarily exceeds PDR, it is possible that some of the packets will go through, as in the case of the GCRA scheme in ATM. On the other hand, if the user attempts to transmit a packet whose size is larger than MBS, this packet will immediately be classified as violating, since the token bucket can never contain more than PBS tokens.

Committed data rate (CDR), committed burst size (CBS), and excess burst size

The traffic that is sent to the network, which is the output of the token bucket P, is policed using the token bucket C, referred to as the *committed token bucket*. The maximum size of token bucket C is set equal to the *committed burst size (CBS)*, expressed in bytes, and the token bucket is replenished at the *committed data rate (CDR)*, expressed in bytes/sec. The output of this token bucket is referred to as the *committed rate* which is the amount of bandwidth the network should allocate for the CR-LSP.

In addition to C, it is possible to use a second policing token bucket E, referred to as the *excess token bucket*. The maximum size of this token bucket is equal to the *excess burst size (EBS)*, expressed in bytes, and the token bucket is replenished at the *committed data rate (CDR)*, expressed in bytes/sec. As will be seen below, this token bucket can be used to decide whether a violating packet should be marked and let into the network, or it should be dropped.

The operation of the committed and excess token buckets is as follows:

- Initially, the token count in the committed token bucket $T_C = CBS$, and the token count in the excess token bucket $T_E = EBS$.
- Thereafter, T_C and T_E are updated every second as follows:
  - If $T_C < CBS$, then T_C is incremented by M bytes (it should not exceed CBS).
  - If $T_E < EBS$, then T_E is incremented by M bytes (it should not exceed EBS) where $CDR = M$ bytes/sec.
- The following action is taken when a packet of size B arrives:
  - If $T_C - B \geq 0$, then there are enough tokens in the committed token bucket for the packet, and $T_C = T_C - B$.
  - If $T_C - B < 0$ and $T_E - B \geq 0$, then there not enough tokens in the committed token bucket, but there are enough tokens in the excess token bucket, and $T_E = T_E - B$.
  - If $T_C - B < 0$ and $T_E - B < 0$, then there not enough tokens in the committed token bucket or in the excess token bucket, and T_C or T_E are not decremented.

Note that if CDR is positive infinity, then an arriving packet will never be in excess of either token bucket counts.

The action taken when the size of a packet exceeds the token count in the token bucket (either committed or excess) is implementation dependent. For instance, if the packet size is larger than the token count in the committed token bucket, but less than the token count in the excess token bucket, then we might chose to mark it and let into the network. If the packet size is larger than both of the token counts, then we might choose to drop it.

An example of how these two policing schemes are used is shown in Figure 7.18. The top diagram refers to the excess token bucket, and the bottom one to the committed token bucket. The rules described in the above paragraph for marking and dropping apply. All four packets arrive at rates higher than CDR. As we can see, packets 1 and 3 go through. Packet 2 arrives at a time when the committed token bucket does not have enough tokens, but there are enough tokens in the excess token bucket. As a result, the token count T_C is left unchanged, the token count T_E is reduced by the size of packet 2, and packet 2 is marked and let into the network. Packets 4 and 5 are dropped, since they arrive at a time when neither token buckets have enough tokens. In both cases the token counts T_C and T_E are unchanged.

The five traffic parameters – PDR, PBS, CDR, CBS, and EBS – can be set to different values so as to create different classes of service, such as a delay sensitive service and a

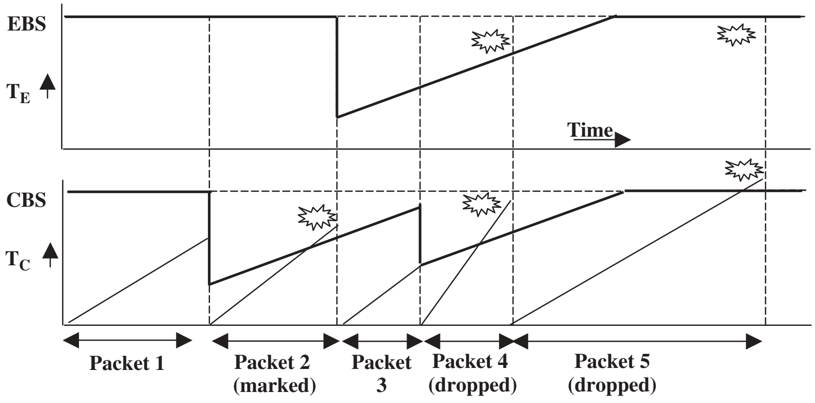


Figure 7.18 An example of the two policing schemes.

best effort service. They can also be set up to provide different ATM service categories. Examples of how these five parameters can be set so that to provide different classes of services are given below in Section 7.2.5.

As mentioned above, the output of the committed token bucket is the traffic that will enter the network. A bandwidth allocation algorithm can be used at each LSR to decide whether the new CR-LSP will be accepted or not. As in ATM, different schemes can be used to calculate how much bandwidth should be allocated to the CR-LSP. The simplest scheme is to allocate a bandwidth equal to CDR. This is equivalent to the peak rate allocation scheme in ATM.

The flags, frequency, and weight fields

The traffic parameters TLV can be included in the label mapping message. This permits an LSR to replace the proposed value for one or more traffic parameter by a lower value. The flags field defines which of the traffic parameters are negotiable; that is, they can be replaced by an LSR with a lower value. It consists of one 2-bit reserved subfield and six 1-bit flags. Five of these flags are associated with the five traffic parameters, and the sixth flag is associated with the weight field. Specifically, flag F1 corresponds to PDR, flag F2 corresponds to PBS, flag F3 corresponds to CDR, flag F4 corresponds to CBS, flag F5 corresponds to EBS, and flag F6 corresponds to the weight field. Each flag indicates whether its associated traffic parameter is negotiable or not. Flag F6 indicates whether the weight is negotiable or not. If a flag is set to 0, then the associated traffic parameter is not negotiable. Otherwise, it is negotiable.

As mentioned above, the CDR can be used to allocate bandwidth to a CR-LSP. The exact allocated bandwidth can vary over time, but the average bandwidth calculated during this time should be at least equal to CDR. The 8-bit frequency field is used to specify this period. The following frequency codes have been defined:

- *Unspecified (value 0).*
- *Frequent (value 1):* That is, the available rate should average at least the CDR when measured over any time interval equal to or longer than a small number of shortest packet times transmitted at the CDR.

- *VeryFrequent (value 2):* That is, the available rate should average at least the CDR when measured over any time interval equal to or longer than the shortest packet time transmitted at the CDR.
- *Reserved (values 3 to 255).*

Finally, the 8-bit weight field is used to indicate the CR-LSP's relative share of the excess bandwidth. Weight values range from 1 to 255. The value 0 means that the weight is not applicable.

7.2.5 Classes of Service

Class services can be constructed by appropriately manipulating the traffic parameters, and the rules regarding passing, marking, and dropping a packet. In Table 7.1, we give the traffic parameters and rules for marking and dropping packets for three classes of service: *delay sensitive (DS) service, throughput sensitive (TS) service*, and *best effort (BE) service*. In the delay sensitive service, the network commits with high probability to deliver packets at a rate of PDR with minimum delay. Packets in excess of PDR will be discarded. In the throughput sensitive service, the network commits to deliver with high

Table 7.1 Traffic parameters – DS, TS, and BE service classes.

Traffic Parameters	**Delay sensitive**	**Throughput sensitive**	**Best effort**
PDR	User-specific	User-specific	Infinite
PBS	User-specific	User-specific	Infinite
CDR	PDR	User-specific	Infinite
CBS	PBS	User-specific	Infinite
EBS	0	0	0
Frequency	Frequent	Unspecified	Unspecified
Dropping action	Drop > PDR	Drop > PDR, BPS, Mark > CDR, CBS	None

Table 7.2 Traffic parameters – ATM service categories.

Traffic parameters	**CBR**	**RT-VBR**	**NRT-VBR**	**UBR**
PDR	PCR	PCR	PCR	PCR
PBS	CDVT	CDVT	CDVT	CDVT
CDR	PCR	SCR	SCR	–
CBS	CDVT	MBS	MBS	–
EBS	0	0	0	0
Frequency	VeryFrequent	Frequent	Unspecified	Unspecified
Dropping action	Drop > PCR	Drop > PCR, Mark > SCR, MBS	Drop > PCR	Drop > PCR

probability packets at a rate of at least CDR. The user can transmit at a rate higher than CDR but packets in excess of CDR have a lower probability of being delivered. In the best effort service there are no service guarantees.

In Table 7.2, we give the traffic parameters and rules for marking and dropping packets for the ATM service categories: *constant bit rate (CBR), real-time variable bit rate (RT-VBR), non-real-time variable bit rate (NRT-VBR)*, and *unspecified bit rate (UBR)*.

7.3 THE RESOURCE RESERVATION PROTOCOL (RSVP)

An alternative signaling protocol to LDP and CR-LDP is the *resource reservation protocol – traffic engineering (RSVP-TE)*. RSVP-TE is an extension of the *resource reservation protocol (RSVP)* which was designed to support the *integrated services (intserv)* architecture. In order to understand RSVP-TE, we first have to understand how RSVP works. In view of this, in this section we describe the main features of RSVP and in the following section we describe RSVP-TE.

The intserv architecture was developed by IETF in the mid 1990s with a view to introducing QoS in the IP network. The following two service classes were defined in intserv:

1. *Guaranteed service*: This service provides firm bounds on the end-to-end queueing delay with no packet loss for all conforming packets.
2. *Controlled-load service*: This service provides the user with a QoS that closely approximates the QoS of the best effort service that the user would receive from an unloaded network. Specifically, a user might assume the following:
   a. A very high percentage of transmitted packets will be successfully delivered by the network to the receiver. The percentage of packets not successfully delivered must closely approximate the basic packet error rate of the transmission links.
   b. The end-to-end delay experienced by a very high percentage of the delivered packets will not greatly exceed the minimum end-to-end delay experienced by any successfully delivered packet.

In *intserv*, the sender specifies how much traffic it will transmit to its receiver(s), and a receiver specifies how much traffic it can receive and the required QoS, expressed in terms of packet loss and end-to-end delay. This information permits each IP router along the path followed by the sender's packets to perform the following functions:

1. *Policing*: This is used to verify that the traffic transmitted by the sender conforms to the *sender's Tspec*, a set of traffic descriptors that characterize the traffic transmitted by the sender.
2. *Admission control*: This is used to decide whether an IP router has adequate resources to meet the requested QoS.
3. *Classification*: This is used to decide which IP packets should be considered as part of the sender's traffic and be given the requested QoS.
4. *Queueing and scheduling*: In order for an IP router to provide different QoS to different receivers, it has to be able to queue packets into different queues and to transmit packets out of these queues according to a scheduler.

The intserv architecture requires a signaling protocol for the reliable establishment and maintenance of resource reservations. As in MPLS, intserv does not require the use of a specific signaling protocol, and it can accommodate a variety of signaling protocols, of which RSVP is the most popular one. RSVP was developed to support the intserv architecture, but it can be used to carry other types of control information. This is because RSVP is not aware of the content of the RSVP protocol fields that contain traffic and policy control information used by the routers to reserve resources. RSVP can be used to make resource reservations for both unicast and many-to-many multicast applications.

RSVP was designed with a view to supporting multiparty conferences, i.e., many-to-many, with heterogeneous receivers. In RSVP, the resource reservation is decided and initiated by a receiver, since only the receiver actually knows how much bandwidth it needs. This approach also permits a receiver to join or leave a multicast whenever it wants.

One problem with the receiver-initiated approach is that the receiver does not know the path from the sender to itself. Therefore, it cannot request resource allocation on each router along the path since it does not know which are these routers. This problem is solved using the *Path* message that originates from the sender and travels along the unicast or multicast route to the receiver. The main purpose of the Path message is to store the *path state* information in each node along the path and to carry information regarding the sender's traffic characteristics and the end-to-end path properties. The following is some of the information contained in the Path message:

- *Phop*: This is the address of the previous hop RSVP-capable router that forwards the message. This address is stored in the path state information at each node, and is used to send the reservation message upstream towards the sender.
- *Sender template*: This field carries the sender's IP address and optionally the UDP/TCP sender port.
- *Sender TSpec*: This defines the traffic characteristics of the data flow that the sender will generate. The sender Tspec format that is used for the intserv architecture will be described below.
- *Adspec*: This carries *one-pass with advertising (OPWA)* information. This is information (advertisements) gathered at each node along the path followed by the Path message. This information is delivered to the receiver, who can then use it to construct a new reservation request or to modify an existing reservation.

Upon receipt of the Path message, the receiver sends an *Resv* message towards the sender along the reverse path that the Path message followed (see Figure 7.19). The following is some of the information contained in the Resv message:

- *Flowspec*: Specifies the desired QoS. It consists of the *receiver TSpec*, the *RSpec*, and the service class. The receiver TSpec is a set of traffic descriptors that is used by the nodes along the path to reserve resources. The RSpec defines the desired bandwidth

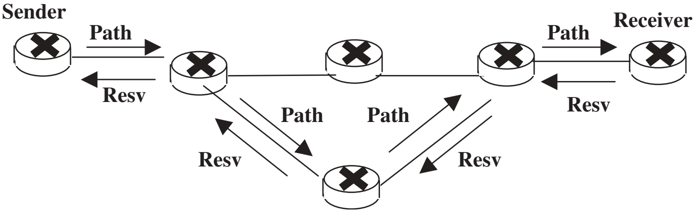


Figure 7.19 An example of the path and resv messages.

and delay guarantees. When RSVP is used in intserv, the service class could be either the guaranteed service or the controlled-load service. The format for the receiver TSpec and RSpec used in the intserv architecture is described below.
- *Filter spec*: Defines the packets that will receive the requested QoS that was defined in the flowspec. A simple filter spec could be just the sender's IP address and optionally its UDP or TCP port.

When a router receives the Resv message, it reserves resources per the receiver's instructions and then sends the Resv message to the previous hop router obtained from the path state information.

RSVP messages are sent in raw IP datagrams without a TCP or UDP encapsulation. (UDP encapsulation is permitted for routers that do not support raw IP datagrams).

RSVP makes use of the notions of *data flow* and *session*. A session is defined by the parameters: destination IP address, protocol id, and optionally destination port number. A data flow is simply the packets transmitted by a sender in a particular session.

RSVP is *simplex*; that is, it makes reservations for unidirectional data flows. Therefore, in order for two users A and B to communicate both ways, two separate sessions have to be established; one session from A to B, and another one from B to A.

7.3.1 Reservation Styles

Three different *reservation styles*, i.e., schemes, can be used with RSVP. In order to understand these schemes, let us consider the case where a number of senders transmit information to the same receiver. Each sender transmits its own data flow of packets in a session, which is defined by the receiver's IP address and protocol id. One reservation option is concerned with the resource reservation of these sessions. In particular, let us assume that several of these data flows pass through the same router. The router has the option to establish a separate reservation for each data flow or to make a single reservation for all of the data flows.

A second reservation option controls the selection of the senders. It can be *explicit* or *wildcard*. In the explicit sender selection, the receiver provides a list of senders from which it wishes to receive data. A sender cannot send packets to the receiver unless its IP address is in the explicit list. In the wildcard sender selection, any sender can transmit data to the receiver.

Based on these two options the following three different styles have been defined:

1. *Wildcard-filter (WF) style*: Any sender can transmit to the session. There is a single resource reservation shared by all of the data flows from all upstream senders. The resource reservation is the largest of all requested reservations.
2. *Fixed-filter (FF) style*: A separate reservation is made for each particular sender that is specified in the explicit list of senders. Other senders identified in the explicit list who transmit in the same session do not share this reservation.
3. *Shared explicit (SE) style*: A list of senders is explicitly stated, and there is a single shared reservation for all of their flows.

7.3.2 Soft State

RSVP takes a soft state approach to managing the reservation state in the routers and hosts. That is, the state information in each router and host has to be periodically refreshed by

Path and Resv messages. The state of a reservation is deleted if no matching refreshing messages arrive before a cleanup timeout interval. State can also be deleted by an explicit teardown message.

When a route changes, the next Path message will initialize the path state on the routers along the new route and the Resv message will establish a reservation on each of these routers. The state of the unused route will time out.

RSVP sends its messages as IP datagrams with no guarantee that they will get delivered. An RSVP message might never get delivered due to transmission errors or buffer overflows. This situation is taken care by the periodic refresh messages. Sending refresh messages increases the load on the network, but it eliminates the need to use a reliable protocol such as TCP which guarantees the reliable delivery of RSVP messages.

7.3.3 The RSVP Message Format

An RSVP message consists of a common header followed by a variable number of *objects*. Each object contains a group of related parameters and it has a variable length. The common header format is shown in Figure 7.20. The following fields have been defined:

- *Vers*: A 4-bit field used to indicate the protocol version number.
- *Flags*: A 4-bit field used for flags; no flags have been specified.
- *MsgType*: The message type is specified by a number carried in this 8-bit field. The following message types and numbers have been specified:
  - ○ 1 – *Path*
  - ○ 2 – *Resv*
  - ○ 3 – *PathErr*
  - ○ 4 – *ResvErr*
  - ○ 5 – *PathTear*
  - ○ 6 – *ResvTear*
  - ○ 7 – *ResvConf*
- *RSVP checksum*: A 16-bit checksum calculated on the entire message.
- *Send_TTL*: An 8-bit field that contains the IP time to live value.
- *RSVP length*: The total length in bytes is stored in this 8-bit field. The length includes the common header and all of the objects that follow.

The object format is shown in Figure 7.21. The following fields have been defined:

- *Length*: A 16-bit field used to indicate the total length in bytes of the object. It must be a multiple of four, and at least equal to four.
- *Class-num*: An 8-bit field used to identify the object class.
- *C-Type*: An 8-bit field used to define the object type.

4 Bits	4 Bits	8 Bits	16 Bits
Vers	**Flags**	**MsgType**	**RSVP checksum**
Send_TTL		**Reserved**	**RSVP length**

Figure 7.20 The common header format.

2 Bytes	1 Byte	1 Byte
Length (bytes)	**Class-num**	**C-Type**
Object contents		

Figure 7.21 The object format.

The following object classes have been defined:

- *NULL*: The contents of a NULL object are ignored by the receiver.
- *SESSION*: Contains the IP destination address, the IP protocol id, and optionally a destination port. This object is required in every RSVP message.
- *RSVP_HOP*: Carries the IP address of the RSVP-capable router that sent this message. For messages sent from the sender to the receiver, the RSVP_HOP object is referred to as the *previous hop (PHOP)* object; for messages sent from the receiver to the sender, it is referred to as the *next hop (NHOP)* object.
- *TIME_VALUES*: Contains the value for the refresh period used by the creator of the message. It is required in every Path and Resv message.
- *STYLE*: Defines the reservation style plus style-specific information. It is required in every Resv message.
- *FLOWSPEC*: Carries the necessary information in an Resv message to make a reservation in a router.
- *FILTER_SPEC*: Defines which data packets should receive the QoS specified in the FLOWSPEC object. It is required in a Resv message.
- *SENDER_TEMPLATE*: Defines the sender's IP address and perhaps some additional demultiplexing information, such as a port number. It is required in a Path message.
- *SENDER_TSPEC*: Contains the traffic characteristics of the sender's data flow. It is required in a Path message.
- *ADSPEC*: Carries the one-pass with advertising (OPWA) information. As discussed above, this information is gathered at each node along the path that is followed by the Path message. This information is delivered to the receiver, who can then use it to construct a reservation request or adjust an existing reservation appropriately.
- *ERROR_SPEC*: Specifies an error in a PathErr, ResvErr, or a confirmation in a ResvConf message.
- *POLICY_DATA*: Carries information that allows a router to decide whether a reservation is administratively permitted. It can appear in a Path, Resv, PathErr, or ResvErr message. One or more POLICY_DATA objects can be used.
- *INTEGRITY*: Carries cryptographic data to authenticate the originating node to verify the contents of this RSVP message.
- *SCOPE*: Carries an explicit list of sender hosts towards the information in the message is to be forwarded. It can appear in a Resv, ResvErr, or ResvTear message.
- *RESV_CONFIRM*: Carries the IP address of a receiver that requested a confirmation. It can appear in a Resv or ResvConf message.

Below, we describe the Path and Resv messages.

7.3.4 The Path Message

The Path message consists of the common header shown in Figure 7.20 followed by the objects:

- INTEGRITY (optional)
- SESSION
- RSVP_HOP
- TIME_VALUES
- POLICY_DATA objects (optional)
- A sender descriptor consisting of SENDER_TEMPLATE and the SENDER_TSPEC
- ADSPEC (optional)

Each sender host sends a Path message for each data flow it wishes to transmit. The Path message is forwarded from router to router using the next hop information in the routing table until it reaches the receiver. Each router along the path captures and processes the Path message. The router creates a path state for the pair {sender, receiver} defined in the SENDER_TEMPLATE and SESSION objects of the Path message. Any POLICY_DATA, SENDER_TSPEC, and ADSPEC objects are also saved in the path state. If an error is encountered a PathErr message is sent back to the originator of the Path message.

7.3.5 The Resv Message

When a receiver receives a Path message, it issues a Resv message which is sent back to the sender along the reverse path traveled by the Path message. Recall that the data packets follow the same path traveled by the Path message. The Resv message is a request to each node along the path to reserve resources for the data flow. It consists of the common header shown in Figure 7.20 followed by the objects:

- INTEGRITY (optional)
- SESSION
- RSVP_HOP
- TIME_VALUES
- RESV_CONFIRM (optional)
- SCOPE (optional)
- POLICY_DATA objects (optional)
- STYLE
- A flow descriptor list

The RSVP_HOP contains the NHOP (i.e., the IP address of the router that sent the Resv message). The presence of the RESV_CONFIRM object in the Resv message is a signal to the router to send a ResvConf message to the receiver to confirm the reservation. The RESV_CONFIRM carries the IP address of the receiver.

The flow descriptor list is style dependent. For the *wildcard-filter (WF)* style the flow descriptor list consists of the FLOWSPEC object. For the *fixed-filter (FF)* and *shared explicit (SE)* styles, it consists of the objects FLOWSPEC and FILTER_SPEC.

As mentioned above, RSVP is not aware of the content of the RSVP objects that contain the traffic information used by the routers to reserve resources. This was done purposefully so that the applicability of RSVP is not restricted to the intserv architecture. Below, we describe the contents of the SENDER_SPEC and FLOWSPEC objects as they have been defined for the intserv architecture.

SENDER_TSPEC and FLOWSPEC contents for intserv

The SENDER_TSPEC contains the traffic parameters: token bucket rate, token bucket size, peak data rate, minimum policed unit (i.e., size of smallest allowable packet), and maximum policed unit (i.e., size of maximum allowable packet).

The contents of the FLOWSPEC depend on whether the controlled-load service or the guaranteed service is requested. When requesting the controlled-load service, the FLOWSPEC consists of the receiver TSPec which contains values for the parameters: token bucket rate, token bucket size, peak data rate minimum policed unit, and maximum policed unit. These parameters are used to calculate the resource reservation in a router.

When requesting the guaranteed service, the FLOWSPEC consists of the receiver TSpec and the RSpec which carries the parameters rate and slack time. These two parameters are used to define the desired bandwidth and delay guarantee.

7.4 THE RESOURCE RESERVATION PROTOCOL – TRAFFIC ENGINEERING (RSVP–TE)

The *resource reservation protocol – traffic engineering (RSVP–TE)* is an extension of the *resource reservation protocol (RSVP)*, described above. RSVP-TE can be used in MPLS to set up LSPs using either the next hop information in the routing table or an explicit route.

In keeping with the terminology used in RSVP-TE (which in fact is the same as in RSVP) we will use the terms *node, sender*, and *receiver* to indicate an LSR, an ingress LSR, and an egress LSR, respectively. Recall that, in RSVP, a session is a data flow with a particular IP destination address and protocol id. In RSVP-TE, a session is an LSP.

RSVP-TE uses downstream-on-demand label allocation to set up an LSP. This is implemented using the Path and Resv messages which have been augmented with new objects. An LSP can be set up using the next hop information in the routing table. Also, RSVP-TE can set up explicitly routed LSPs. This is done by using a new object – the EXPLICIT_ROUTE – to encapsulate the hops that make up the explicit path. Each hop in the path can be an individual node or an abstract node. An abstract node is a group of nodes whose internal topology is opaque to the sender. Strict or loose routing through an abstract node is permitted.

To set up an LSP, the ingress node sends a Path message with a LABEL REQUEST object. This is a new object and it indicates that a label binding for the path is requested. If an explicit route is requested, then the EXPLICIT_ROUTE object will be inserted in the Path message. If the LSP is set up using the next hop information in the routing table, then the EXPLICIT_ROUTE object is not used.

The Path message is forwarded to the next hop indicated in the routing table for the particular IP destination address of the receiver, or the next hop indicated in the EXPLICIT_ROUTE object. A node incapable of accepting the new requested LSP sends back a PathErr message.

The receiver, i.e., the egress LSR of the LSP, responds with an Resv message. A new object called the LABEL object, is inserted in the message which is sent back upstream towards the sender, i.e., the ingress node, following the inverse path traversed by the Path message. Each node that receives the Resv message with a LABEL object uses that label for outgoing traffic associated with the LSP. It allocates a new label, places it in the LABEL object of the Resv message and sends it upstream to the previous hop node. The LSP is established when the sender receives the Resv message. As we can see, in RSVP-TE an LSP is established using the ordered LSP control scheme.

RSVP-TE enables the reservation of resources along the LSP. For example, bandwidth can be allocated to an LSP using standard RSVP reservations together with intserv service classes. Resource reservation is optional, and LSPs can be set up without reserving any resources. Such LSPs can be used, for instance, to carry best effort traffic and implement backup paths.

Other features of RSVP-TE include setup and hold priorities for an LSP, and dynamic reroute of an established LSP. Also, by adding a RECORD_ROUTE object to the Path message, the sender can receive information about the actual route that the LSP traverses.

7.4.1 Service Classes and Reservation Styles

There is no restriction in RSVP-TE as to which intserv service classes should be supported. RSVP-TE, however, should support the controlled-load service and the *null service*. The controlled-load service class was described in the previous section. The null service class is a newer service class that was introduced in RSVP in order to support RSVP signaling in the *differentiated service (diffserv)* architecture. In the null service, an application does not request a resource reservation on each node along the path from the sender to the receiver. Instead, a QoS policy agent in each node provides the appropriate QoS parameters to the application, as determined by a network administrator.

In the previous section, we defined the three RSVP reservation styles: *wildcard-filter (WF), fixed-filter (FF)*, and *shared explicit (SE)*. Of these three reservation styles, the wildcard-filter is not used in RSVP-TE. The receiver node can use the FF or SE style for an LSP, and it can chose different styles for different LSPs.

When using the FF style, each LSP has its own reservation on each node along the path, and each node allocates a unique label for each sender. For instance, if an explicit route is set up using the FF style, then each node will reserve resources to the LSP and a unique label that the previous hop node has to use when transmitting packets in this specific LSP. Consider the case where an LSP is set up using the next hop information from the routing table. In this case, if there are multiple senders, a multipoint-to-point inverse tree will be formed. Each sender has its own path to the receiver, which is independent of the paths from the other receivers. A node in this inverse tree that handles several such paths reserves separate resources to each path and allocates a different label for each path to be used by the previous hop node. As a result, this inverse tree consists of multiple point-to-point independent LSPs. This also means that the same previous hop node can use different labels to transmit traffic to the same receiver from different senders.

The SE style allows a receiver to explicitly specify the senders to be included in a reservation. There is a single reservation on each node for all senders whose path to the receiver pass through that node. Different labels are assigned to different senders, thereby creating separate LSPs.

7.4.2 The RSVP-TE New Objects

The following five new objects have been introduced to support the functionality of RSVP-TE:

- LABEL
- LABEL_REQUEST
- EXPLICIT_ROUTE
- RECORD_ROUTE
- SESSION_ATTRIBUTE

Also, new C-Types have also been defined for the SESSION, SENDER_TEMPLATE, and FILTER_SPEC objects. We now proceed to examine the format of these new five objects.

The LABEL object

The LABEL object is used in the Resv message to advertise a label. For the FF and SE styles, a node allocates a separate label for each sender to be used by the previous hop node. The format of the LABEL object is shown in Figure 7.22. The LABEL object class (given in the Class-num field) is 16, the object type (given in the C-Type field) is C-Type 1, and the object contents is populated with a single label encoded in 4 bytes. A generic MPLS label and a frame relay label is encoded in four bytes. An ATM label is encoded with the VPI field in the bits 0 to 15 and the VCI field in the bits 16 to 31.

A node can support multiple label spaces. For instance, it can associate a unique label space for each incoming interface. Labels received in Resv messages on different interfaces are always considered as being different even if the label value is the same.

The LABEL_REQUEST object

The LABEL_REQUEST object class (Class-num field) is 19, and there are three different object types (C-Type field): C-Type 1, C-Type 2 and C-Type 3. C-Type 1 is a label request for generic labels, C-Type 2 is a label request for ATM labels, and C-Type 3 is a label request for frame relay labels. These three formats are shown in Figure 7.23.

The object contents of the C-Type 1 consist of a 16-bit reserved field, and the 16-bit L3PID field which is populated with an identifier of the Layer 3 protocol using this path.

The object contents of the C-type 2 consist of the following fields (reserved fields are not listed):

- *L3PID*: A 16-bit field that carries the identifier of the Layer 3 protocol that is using this path.

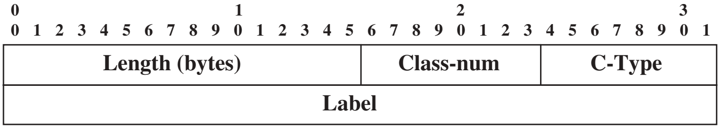


Figure 7.22 The LABEL object format.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td>Length (bytes)</td><td>Class-num</td><td>C-Type</td></tr>
<tr><td>Reserved</td><td colspan="2">L3PID</td></tr>
</table>

C_Type = 1

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td colspan="3">Length (bytes)</td><td>Class-num</td><td>C-Type</td></tr>
<tr><td colspan="3">Reserved</td><td colspan="2">L3PID</td></tr>
<tr><td>M</td><td>Res</td><td>Minimum VPI</td><td colspan="2">Minimum VCI</td></tr>
<tr><td colspan="2">Res</td><td>Maximum VPI</td><td colspan="2">Maximum VCI</td></tr>
</table>

C_Type = 2

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td colspan="3">Length (bytes)</td><td>Class-num</td><td>C-Type</td></tr>
<tr><td colspan="3">Reserved</td><td colspan="2">L3PID</td></tr>
<tr><td>Reserved</td><td>DLI</td><td colspan="3">Minimum DLCI</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="3">Maximum DLCI</td></tr>
</table>

C_Type = 3

Figure 7.23 The LABEL_REQUEST object formats.

- *M*: A 1-bit field that is used to indicate whether the node is capable of merging in the data plane.
- *Minimum VPI*: A 12-bit field that gives a lower bound on the block of the VPI supported values.
- *Minimum VCI*: A 16-bit field that gives a lower bound on the block of the VCI supported values.
- *Maximum VPI*: A 12-bit field that gives an upper bound on the block of the VPI supported values.
- *Maximum VCI*: A 16-bit field that gives an upper bound on the block of the VCI supported values.

The object contents of the C-type 3 consist of the following fields (reserved fields are not listed):

- *L3PID*: A 16-bit field that carries the identifier of the Layer 3 protocol that is using this path.
- *DLCI length indicator (DLI)*: A 2-bit field that indicates the length of the DLCI value. The following values are supported: 0 and 2. When DLI = 0, then the length of the DLCI is 10 bits. When it is set to 2, the length of the DLCI value is 23 bits.
- *Minimum DLCI*: This 23-bit field gives the lower bound on the block of the supported DLCI values.

- *Maximum DLCI*: This 23-bit field give the upper bound on the block of the supported DLCI values.

In order to establish an LSP, the sender creates a Path message with a LABEL_REQUEST object. This object indicates that a label binding for this path is requested and it provides an indication of the network protocol layer that is to be carried over the path. This permits packets form non-IP network layer protocols to be sent down an LSP. This information is also useful in label allocation, because some reserved labels are protocol specific. A receiver that cannot support the protocol indicated in the L3PID field, sends a PathErr message back to the sender.

The EXPLICIT_ROUTE object (ERO)

This object is used to specify the hops in the requested explicit route. Each hop could be a single node or a group of nodes, referred to as an abstract node. For simplicity, RSVP-TE refers to all of the hops as *abstract nodes*, with the understanding that an abstract node could consist of a single node.

The EXPLICIT_ROUTE object class is 20, and only one object type (C-Type 1) has been defined. The object contents consists of a series of variable-length sub-objects, each of which contains an abstract node. The format of the sub-object is shown in Figure 7.24. The following fields have been defined:

- *L*: A 1-bit field used to indicate whether the route through an abstract node is loose or strict.
- *Type*: This 7-bit field is populated with a value that indicates the type of contents of the sub-object. The following values have been defined: 0 if the sub-object contains an IPv4 prefix, 1 if it contains an IPv6 prefix, and 32 if it contains an autonomous system number.
- *Length*: This 8-bit field is populated with the length (in bytes) of the sub-object including the L, type, and length fields.

The format of the sub-objects for the IPv4 and IPv6 is shown in Figure 7.25. The field *IPv4 address* (respectively *IPv6 address*) in the IPv4 (IPv6) sub-object contains an IPv4 (IPv6) prefix whose length is given in the prefix length field. The abstract node represented by this sub-object is the set of all nodes whose IPv4 (IPv6) address has the prefix given in the IPv4 (IPv6) address field. Note that a prefix length of 128 indicates a single node.

The sub-object format for the autonomous system is the same as the one shown in Figure 7.24, with the sub-object contents consisting of a two-byte field populated with the autonomous system number. The abstract node represented by this sub-object is the set of all nodes belonging to the autonomous system.

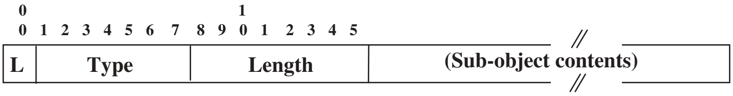


Figure 7.24 The format of a sub-object.

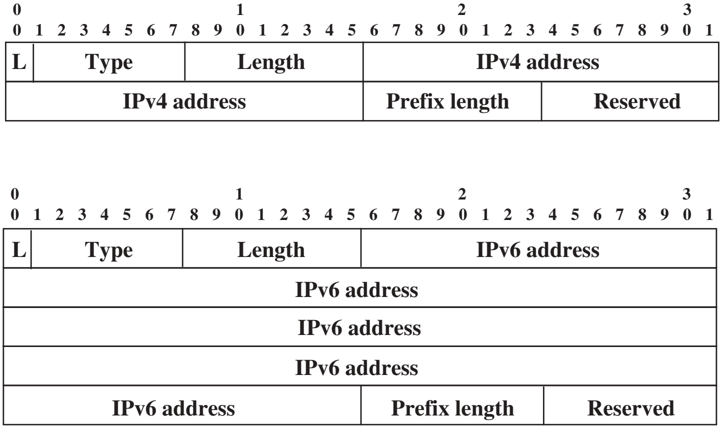


Figure 7.25 The format of the sub-object for IPv4 and IPv6 prefixes.

The RECORD_ROUTE object (RRO)

The existence of loose routes through an abstract node means that it is possible that loops can be formed particularly during periods when the underlying routing protocol is in a transient state. Loops can be detected through the RECORD_ROUTE object. In this object the IP address of each node along the path can be recorded. Also, the labels used along the path can be recorded. The RECORD_ROUTE object can be present in both Path and Rev messages.

The RECORD_ROUTE object class is 21 and there is one object type, C-Type 1. The object contents consists of a series of variable-length sub-objects organized in a last-in-first-out stack. Three different sub-objects have been defined: the IPv4 sub-object, the IPv6 sub-object, and the label sub-object. The first two sub-objects are the same as the IPv4 and the IPv6 sub-objects defined above in the EXPLICIT_ROUTE object and shown in Figure 7.22 (with the exception that the reserved field has been replaced by a flags field.) The label sub-object has the structure shown in Figure 7.24, and it contains the entire contents of the LABEL object.

The SESSION_ATTRIBUTE object

This object contains setup holding priorities for an LSP, plus various flags. The *setup priority* is the priority used for allocating resources. The holding priority is the priority used for holding onto resources.

7.4.3 The RSVP-TE Path and Resv Messages

The RSVP-TE Path and Resv message are similar to those in RSVP. The RSVP-TE Path message consists of the common header shown in Figure 7.20 followed by the objects:

- INTEGRITY (optional)
- SESSION

- RSVP_HOP
- TIME_VALUES
- EXPLICIT_ROUTE (optional)
- LABEL_REQUEST
- SESSION_ATTRIBUTE (optional)
- POLICY_DATA objects (optional)
- A sender descriptor consisting of the SENDER_TEMPLATE and the SENDER_TSPEC
- ADSPEC (optional)
- RECORD_ROUTE (optional)

The RSVP-TE Resv message consists of the common header shown in Figure 7.20 followed by the objects:

- INTEGRITY (optional)
- SESSION
- RSVP_HOP
- TIME_VALUES
- RESV_CONFIRM (optional)
- SCOPE (optional)
- POLICY_DATA objects (optional)
- STYLE
- A style-dependent flow descriptor list. For the *fixed-filter (FF)* style, it consists of the objects: FLOWSPEC, FILTER_SPEC, LABEL, RECORD_ROUTE (optional). For the *shared explicit (SE)* style, it consists of the objects: FILTER_SPEC, LABEL, RECORD_ROUTE (optional).

7.4.4 RSVP-TE Extensions

RSVP was designed to support resource reservations for data flows defined between a sender and a receiver. As the number of data flows increases, the RSVP overhead on the network increases as well due to the continuous refreshing messages that have to be exchanged. Also, the memory required to store the path state information in each router and the amount of processing increases as well. In view of this, RSVP is not considered a protocol that scales up well. Similar problems arise in RSVP-TE, since it is based on RSVP.

Several solutions have been proposed to alleviate these problems. For instance, a mechanism for reliable delivery has been proposed that reduces the need for refresh messages. This mechanism makes use of two new objects, MESSAGE_ID and MESSAGE_ID_ACK. Also, the amount of data transmitted due to refresh messages can be reduced by using the *Srefresh* message, a new summary refresh message.

PROBLEMS

1. In LDP the hallo adjacency as well as the session have to be continuously refreshed. Since the hallo adjacencies in a session are continuously refreshed, why is there a need to also refresh the session?
2. Explain the need for loosely explicit routes in CR-LDP. Give an example of an application that requires pinning.

3. Could CR-LDP work using unsolicited downstream label allocation and independent order? Why?
4. Consider the traffic parameters for the delay sensitive service class given in Table 7.1. Do these parameters suffice to provide this service? What additional mechanism(s) is (are) required?
5. Explain why in RSVP the Path message contains the RSVP_HOP object.
6. Explain the difference between the fixed-filter style and the shared explicit style.
7. Is it possible for RSVP-TE to set up a CR-LSP based on the next hop routing information? How?
8. Compare CR-LDP with RSVP-TE. What are the common features in these two protocols? Identify some of the main differences in these two protocols.

8

Optical Fibers and Components

This chapter deals with the physical layer of *wavelength division multiplexing (WDM)* optical networks. We first give a general overview of WDM optical networks. We then proceed to describe how light is transmitted through an optical fiber. Specifically, we discuss the *index of refraction, step-index* and *graded-index* optical fibers, *multi-mode* and *single mode* optical fibers, and various optical effects that occur when light is transmitted through an optical fiber, known as *impairments*. Finally, we conclude this chapter by describing some of the components used in WDM optical networks, such as lasers, optical amplifiers, 2 × 2 couplers and star couplers, and *optical cross-connects (OXCs)*.

This chapter, somewhat stretches the intended scope of this book, which focuses on layers higher than the physical layer. However, due to the novelty of optical networks, it is important to have some knowledge of the underlying WDM technology. It is not necessary to read this chapter in detail in order to understand the subsequent chapters on optical networks. The key sections to study are the introductory section, (Section 8.1) and the section on components (Section 8.3).

8.1 WDM OPTICAL NETWORKS

WDM refers to the technology of combining multiple wavelengths onto the same optical fiber. Each wavelength is a different channel. Conceptually, WDM is the same as *frequency division multiplexing (FDM)*, which is used in microwave radio and satellite systems.

A typical point-to-point connection is shown in Figure 8.1. At the transmitting end, there are W independent transmitters. Each transmitter *Tx* is a light source, such as a laser, and is independently modulated with a data stream. The output of each transmitter is an optical signal on a unique wavelength $\lambda_i,\ \ i = 1, 2, \ldots, W$. The optical signals from the W transmitters are combined into a single optical signal at the wavelength multiplexer and transmitted out onto a single optical fiber. At the other end, the combined optical signal is demultiplexed into the W individual signals, and each one is directed to the appropriate receiver *(Rx)*, where it is terminated and converted to the electric domain. Amplification is used immediately after the wavelength multiplexer and before the wavelength demultiplexer. Also, if the fiber is very long, the signal is further amplified using in-line amplifiers.

As can be seen, this point-to-point system provides W independent channels, all on the same fiber. As the WDM technology improves, the number of wavelengths that can

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

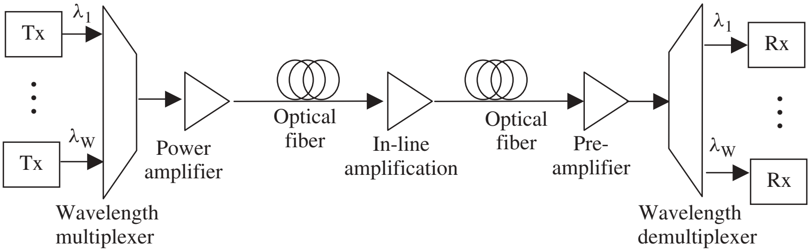


Figure 8.1 A WDM point-to-point link.

be transmitted onto the same fiber increases as well. Thus, the capacity of a link can be increased by utilizing the WDM technology rather than adding new fibers. The latter solution is significantly more expensive than the upgrading of components necessary for the introduction of WDM.

More complex WDM optical networks can be built using *optical cross-connects (OXC)*. An OXC is an $N \times N$ optical switch, with N input fibers and N output fibers. The OXC can switch optically all of the incoming wavelengths of the input fibers to the outgoing wavelengths of the output fibers, assuming no external conflicts at the output fibers. For instance, it can switch the optical signal on incoming wavelength λ_i of input port k to the outgoing wavelength λ_i of output port m. If it is equipped with converters, it can also switch the optical signal of the incoming wavelength λ_i to another outgoing wavelength λ_j.

An OXC can also be used as an *optical add/drop multiplexer (OADM)*. That is, it can terminate the optical signal of a number of incoming wavelengths and insert new optical signals on the same wavelengths in an output port. The remaining wavelengths are switched as described above.

An OXC can switch wavelengths in a *static* or *dynamic* manner. In the static case, the OXC is configured to switch permanently the incoming wavelengths to the outgoing wavelength. In the dynamic case, the OXC will switch a particular incoming wavelength to an outgoing wavelength on demand. An OADM can also add/drop wavelengths either in a static manner or dynamically (i.e., on demand).

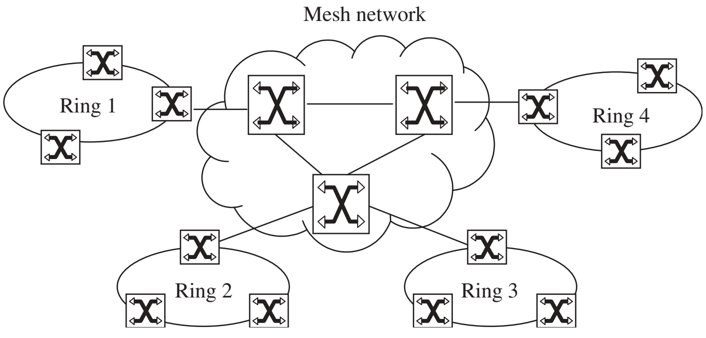


Figure 8.2 An example of an optical network.

A typical WDM optical network, as operated by a telecommunication company, consists of WDM metro (i.e., metropolitan) rings, interconnected by a mesh WDM optical network, i.e. a network of OXCs arbitrarily interconnected. An example of such a network is shown in Figure 8.2.

There are many different types of optical components used in a WDM optical network, and some of these components are described in Section 8.3. We now proceed to examine some of the basic principles of light transmission through an optical fiber.

8.2 HOW LIGHT IS TRANSMITTED THROUGH AN OPTICAL FIBER

Light radiated by a source can be seen as consisting of a series of propagating electromagnetic spherical waves (see Figure 8.3). Along each wave, one can measure the electric field, indicated in Figure 8.3 by a dotted line, which is vertical to the direction of the light. The magnetic field (not shown in Figure 8.3) is perpendicular to the electric field.

The intensity of the electrical field oscillates following a sinusoidal function. Let us mark a particular point, say the peak, on this sinusoidal function. The number of times that this particular point occurs per unit of time is called the *frequency*. The frequency is measured in *Hertz*. For example, if this point occurs 100 times, then the frequency is 100 Hertz. An electromagnetic wave has a frequency f, a speed v, and a wavelength λ. In vacuum or in air, the speed v is approximately the speed of light which is 3×10^8 meters/sec. The frequency is related to the wavelength through the expression: $v = f\lambda$.

An optical fiber consists of a transparent cylindrical inner core which is surrounded by a transparent cladding (see Figure 8.4). The fiber is covered with a plastic protective cover. Both the core and the cladding are typically made of silica (SiO_2), but they are made so that to have different *index of refraction*. Silica occurs naturally in impure forms, such as quartz and sand.

The index of refraction, known as the *refractive index*, of a transparent medium is the ratio of the velocity of light in a vacuum c to the velocity of light in that medium v, that

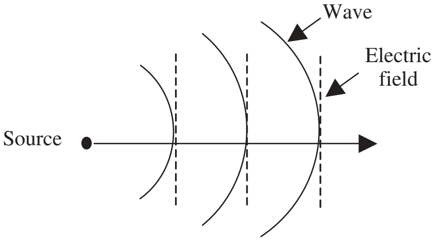


Figure 8.3 Waves and electrical fields.

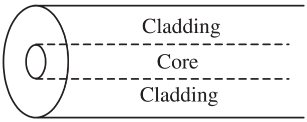


Figure 8.4 An optical fiber.

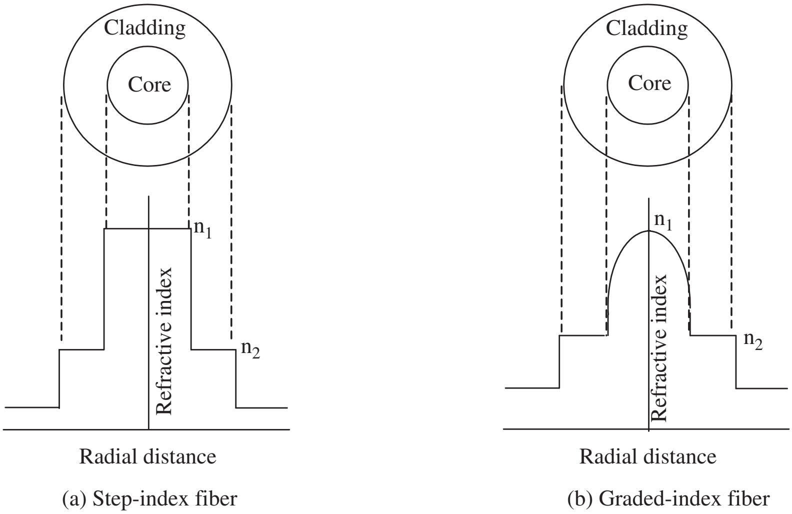


Figure 8.5 Step-index and graded-index fibers.

is $n = c/v$. The value of the refractive index of the cladding is always less than that of the core.

There are two basic refractive index profiles for optical fibers: the *step-index* and the *graded-index*. In the step-index fiber, the refractive index of the core is constant across the diameter of the core. In Figure 8.5(a), we show the cross-section of an optical fiber and below the refractive index of the core and the cladding has been plotted. (For presentation purposes, the diameter of the core in Figure 1, Figure 5, and some of the subsequent figures is shown as much bigger than that of the cladding.) In the step-index fiber, the refractive index for the core (n_1) remains constant from the center of the core to the interface between the core and the cladding. It then drops to n_2, inside the cladding. In view of this step-wise change in the refractive index, this profile is referred to as step-index. In the graded-index fiber, the refractive index varies with the radius of the core (see Figure 8.5(b)). In the center of the core it is n_1, but it then drops off to n_2 following a parabolic function as we move away from the center towards the interface between the core and the cladding. The refractive index is n_2 inside the cladding.

Let us investigate how light propagates through an optical fiber. In Figure 8.6, we see a light ray is incident at an angle θ_i at the interface between two media with refractive indices n_1 and n_2, where $n_1 > n_2$. Part of the ray is *refracted* – that is, transmitted through the second medium – and part of it is *reflected* back into the first medium. Let θ_i be the angle between the incident ray and the dotted line, an imaginary vertical line to the interface between the two media. This angle is known as the *incidence angle*. The *refracted angle* θ_f is the angle between the refracted ray and the vertical dotted line. We have that $\theta_i < \theta_f$. Finally, the *reflected angle* θ_r is the angle between the reflected ray and the vertical dotted line. We have that $\theta_r = \theta_f$.

Interestingly, an angle θ_c, known as the *critical angle*, exists, past which the incident light will be reflected entirely. That is, if $\theta_i > \theta_c$, then the entire incident ray will be

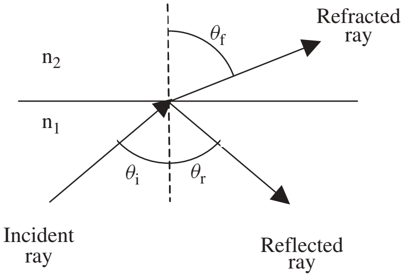
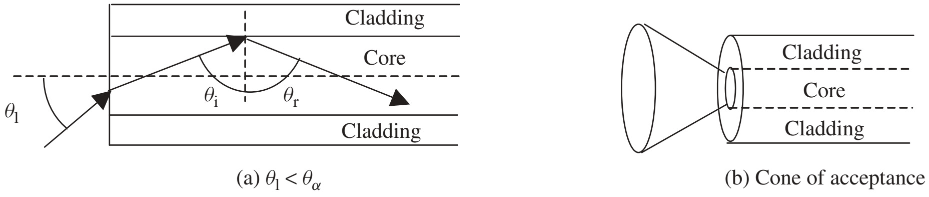


Figure 8.6 Refraction and reflection of a light ray.

reflected. For a light ray to be transmitted through an optical fiber, it has to hit the interface between the core and the cladding at an angle θ_i which is greater than the critical angle θ_c. In order for this to happen, the light ray must be launched at the end of the fiber at an angle θ_l, which is less than a critical angle θ_α (see Figure 8.7(a)). The angle θ_l is referred to as the *launch angle*. This results into a *cone of acceptance* within which a light ray must be launched (see Figure 8.7(b)). Typically, a lens is used to focus the launched light onto a small area of the core (see Figure 8.8).

In Figure 8.7(a), we see that the light ray travels through the core in a straight line until it is reflected at the interface between the core and the cladding. The reflected ray also continues on in a straight line. This is because we assumed a step-index optical fiber, and as mentioned above, the refractive index for the core remains constant from the center of the core to the cladding. In the case of a graded-index fiber, however, the refractive index changes with the distance from the center of the core following a parabolic function. In this case, the path of a light ray will be a curve (see Figure 8.9).



(a) $\theta_l < \theta_\alpha$

(b) Cone of acceptance

Figure 8.7 Angle of launching a ray into the fiber.

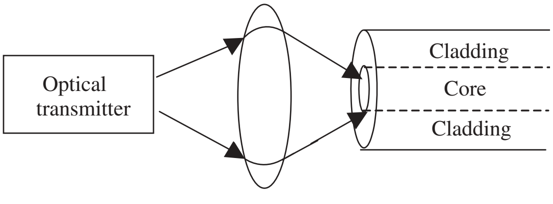


Figure 8.8 A lens is used to focus the launched light.

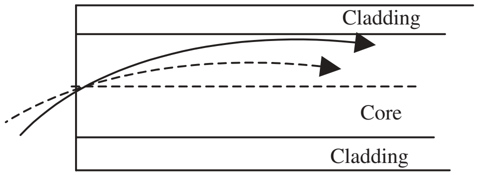


Figure 8.9 Path of a light ray in a graded-index fiber.

8.2.1 Multi-mode and Single-mode Optical Fibers

Both *multi-mode fiber* and *single-mode fiber* are used in communication systems. Single-mode fiber is used in long-distance telephony, CATV, and packet-switching networks. Multi-mode fiber is often cheaper than single-mode fiber, and is used in short distance networks, such as LANs. Both fiber types have the same diameter (125 μm), but they have different core diameters. Specifically, the single-mode fiber has a very small core diameter, whereas the multi-mode fiber has a large core diameter. Core/cladding diameters are given in Table 8.1.

In order to understand the difference between multi-mode and single-mode fibers, we have to first introduce the concept of *a fiber mode*. Let us consider two incident rays, Rays 1 and 2, which are launched into the fiber at the same launch angle θ_l (see Figure 8.10). Ray 1 is reflected for the first time at Point A, and Ray 2 is reflected at Point B. Recall that a ray, whether an incident ray launched into the fiber or a reflected ray, has an electric field which is vertical to the direction of its path. This electric field is depicted in Figure 8.10 by the sinusoidal curve along the ray's path. For presentation purposes, we assume a step-index optical fiber.

The electric field of the reflected Ray 1 suffers a phase-shift at the interface between the core and the cladding. This phase-shift depends on a number of factors, such as the ratio of the refractive index of the core and the cladding and the angle of incidence θ_i. A similar phase-shift applies to the electric field of the reflected Ray 2. However, the electric field of the incident Ray 1 traveling upwards is in phase with the electric field of the reflected Ray 2 which is also traveling upwards. Likewise, the electric field of the incident Ray 2 traveling downwards is in phase with the electric field of the reflected Ray 1 which is also traveling downwards.

The electric fields of the incident rays and the reflected rays interfere with each other, and depending upon the case, they either reinforce each other or they extinguish each other. The electric fields of the two upwards (or the two downwards) traveling rays are in-phase (see Figure 8.10). As a result, they reinforce each other; the fiber is excited; and a light beam is formed, which is guided through the core of the fiber. On the other hand,

Table 8.1 Core/cladding diameters.

Fiber type	Core/cladding diameters
Multi-mode fiber	50/125 μm, 62.5/125 μm, 100/140 μm
Single-mode fiber	9 or 10/125 μm

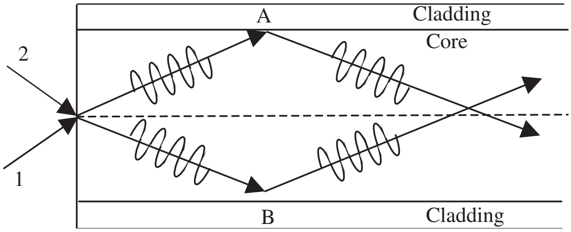


Figure 8.10 Electric fields.

the electric fields of a downward and an upward traveling ray are in-phase, and so they interfere with each other and no light beam is formed.

Due to these interactions, the electrical fields cancel each other out at the interface of the core and the cladding, but they reinforce each other in the center of the core. The resulting light beam has an electric field whose amplitude varies as shown in Figure 8.11, case $m = 0$. The pattern of the electric field is called a *fiber mode*.

Different fiber modes can be created by launching the two rays at a different launch angle θ_l (see Figure 8.10). The amplitude of the electric field for each mode is 0 at the interface between the core and the cladding, as in the case of $m = 0$, discussed above. However, zero amplitude points (known as *null points*) can be also created inside the core. For instance, in Figure 8.11, we show examples of the amplitude of the electric field of a formed beam where it has a null point in the center of the core ($m = 1$), and where it has two null points inside the core ($m = 2$). In general, the different modes in a fiber are numbered 0, 1, 2, and so on. Also, the number of a mode is equal to the number of null points inside the core of the fiber associated with the mode.

As shown in Figure 8.8, a lens is used to focus the launched light onto a small area of the core. As a result, there are many different rays that enter the fiber at different launch angles, and thus many different fiber modes are created and propagated down the core of the fiber. In other words, the propagation of the light through the fiber core, is done in terms of different modes all propagating down the fiber core. In Figure 8.12, various modes for a step-index and a graded-index fiber are shown. As mentioned above, the path of a ray through a step-index fiber is a straight line until it gets reflected into another straight line. On the other hand, the path of a ray through a graded-index fiber is a curve.

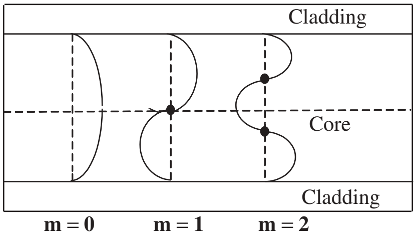
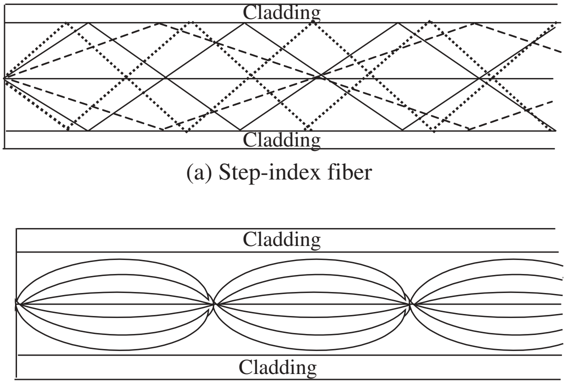


Figure 8.11 Electric field amplitudes for various fiber modes.



(a) Step-index fiber

(b) Graded-index fiber

Figure 8.12 Propagation of modes.

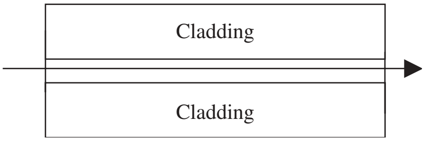


Figure 8.13 Single-mode fiber.

The number of modes in a fiber is related to the diameter of the core. As shown in Table 8.1, multi-mode fibers have a large core diameter, and the light propagates through the core in different modes, as explained above. Single-mode fibers (see Figure 8.13) have a core with a very narrow diameter that only permits the propagation of the single ray with mode 0 (i.e., $m = 0$).

8.2.2 Impairments

The transmission of light through an optical fiber is subjected to optical effects, known as *impairments*. There are *linear* and *non-linear* impairments.

Linear impairments are due to *attenuation* and *dispersion*. Attenuation is the decrease of the optical power along the length of the fiber. Dispersion is the distortion of the shape of a pulse. These impairments are called linear because their effect is proportional to the length of the fiber.

Non-linear impairments can be due to the dependency of the refractive index on the intensity of the applied electrical field. The most important non-linear effects in this category are: *self-phase modulation* and *four-wave mixing*. Another category of non-linear impairments includes the *stimulated Raman scattering* and *stimulated Brillouin scattering*. These two impairments are due to the scattering effects in the fiber medium because of the interaction of light waves with phonons (molecular vibrations) in the silica medium. All of these impairments are called non-linear because when they occur the response of a medium such as silica, is a non-linear function of the applied electric and magnetic field amplitude.

We now proceed to examine the two linear impairments: *attenuation* and *dispersion*.

Attenuation

Attenuation varies with wavelength and is defined in *decibel* per kilometer. A *decibel (dB)* is a unit used to express the relative difference in signal strength; it is calculated as follows:

$$\text{Number of decibels} = 10 \log_{10}\left(\frac{P_i}{P_0}\right),$$

where P_i is the optical power input to the fiber and P_0 is the optical power output from the fiber. The attenuation for any length of fiber is $\alpha_{\mathrm{dB}}L$, where α_{dB} is the attenuation expressed in decibels per kilometer, and L is the length of the fiber.

Attenuation is due to a number of reasons, such as: absorption, Rayleigh scattering, and reflections due to splices and connectors.

- *Absorption*: The light is absorbed as it passes through the optical fiber.
- *Rayleigh scattering*: The density of the particles of the core is not always the same due to imperfections of the fiber. This causes the light to be scattered. Long wavelengths have less scattering.
- *Reflection by splices and connections*: A long fiber link typically consists of several segments connected through splices. There might also be connectors for connecting users. Both splices and connectors reflect back in the opposite direction of the fiber some of the light, thus reducing the power of the light moving forward.

As mentioned above, attenuation varies with the wavelength. Figure 8.14 gives the attenuation for a single-mode fiber as a function of the wavelength. The attenuation for short wavelengths (round 850 nm) is more than three times the attenuation for the long wavelengths (1300 nm and 1550 nm). Typically, the windows near 850 nm, 1300 nm, and 1550 nm are used.

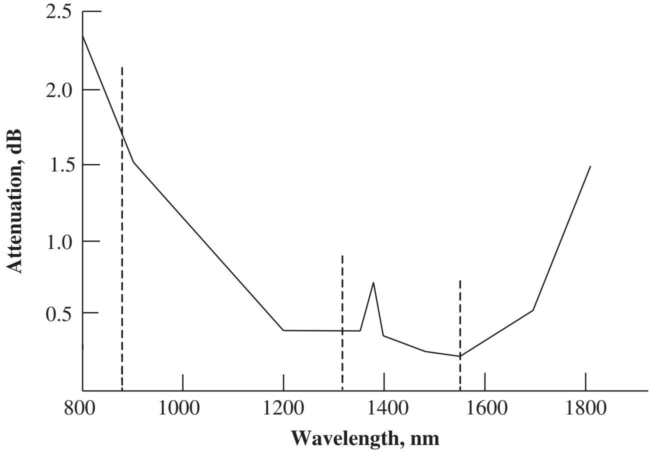


Figure 8.14 Attenuation as a function of wavelength.

The lost power of an optical signal can be restored using an optical amplifier (see Section 8.3.3).

Dispersion

Dispersion is due to a number of reasons, such as *modal dispersion, chromatic dispersion*, and *polarization mode dispersion.*

Modal dispersion is associated with multi-mode fibers. As discussed in the previous section, when light is launched at the end of the fiber many fiber modes are created and propagated down the core of the fiber. Now, as shown in Figure 8.12(a), due to the different angles at which rays enter the core of a multi-mode fiber, some modes travel a longer distance to get to the end of the fiber than others. In view of this, the modes have different delays, which causes a spreading of the output pulse (see Figure 8.15(b)). Pulse spreading increases with the length of the fiber. In the case of graded-index multi-mode fiber (see Figure 8.15(c)), pulse spreading is minimum. This is because in graded-index fibers, the rays travel closer to the center of the core due to the parabolic refractive index. Consequently, the modes do not have a significant delay difference.

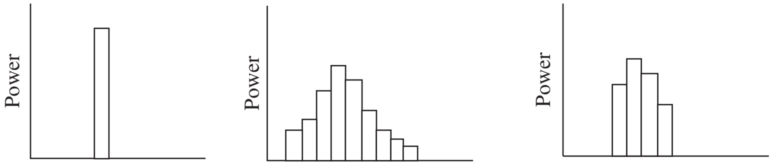
At high speeds, pulse spreading can cause pulses to run into one another, to the point where the data stream cannot be recovered. In the case of a single-mode fiber, pulse spreading is almost non-existent. This is because the core is small and only one ray is transmitted through.

Chromatic dispersion is due to the fact that the refractive index of silica, the material used to make the core of the fiber, is frequency dependent. In view of this, different frequencies travel at different speeds, and as a result they experience different delays, as in the case of the modes in a multi-mode fiber. These delays cause spreading in the duration of the output pulse. Chromatic dispersion is measured in $ps/nm - km$, where ps refers to the time spread of the pulse, nm is the spectral width of the pulse, and km corresponds to the length of the fiber. Chromatic dispersion is usually overshadowed by model dispersion in multi-mode fibers. This type of dispersion is called *material dispersion*.

Chromatic dispersion can be corrected using a *dispersion compensating fiber*. The length of this fiber is proportional to the dispersion of the transmission fiber. Approximately, a spool of 15 km of dispersion compensating fiber is placed for every 80 km of transmission fiber. Dispersion compensating fiber introduces attenuation of about 0.5 dB/*km*.

Another type of dispersion is the *waveguide* dispersion, which is important only in single-mode fibers. Single-mode fibers are designed so that material dispersion and waveguide dispersion cancel each other out.

The *polarization mode dispersion (PMD)* is due to the fact that the core of the fiber is not perfectly round. When light travels down a single-mode fiber it gets polarized and it



(a) Input pulse (b) Output pulse (step-index) (c) Output pulse (graded-index)

Figure 8.15 Pulse spreading.

travels along two polarization planes which are vertical to each other. In an ideal circularly symmetric fiber the light traveling on each polarized plane has the same speed with the light traveling on the other plane. However, when the core of the fiber is not round, the light traveling along one plane might travel either slower or faster than the light polarized along the other plane. This difference in speed will cause the pulse to break.

8.2.3 Types of Fibers

The multi-mode fiber has been used extensively in LANs and, more recently, in 1-Gigabit and 10-Gigabit Ethernet. A vast majority of the installed multi-mode fiber has a core diameter of 62.5 μm and operates in the region of 850 nm and 1300 nm. It provides speeds up to 100 Mpbs. A small percentage of multi-mode fiber adheres to an earlier standard which has a core diameter of 50 μm and operates in both regions of 850 nm and 1300 nm.

Single-mode fiber is used for long-distance telephony, CATV, and packet-switching networks. The following are various different types of single-mode fiber, classified according to their dispersion loss.

1. *Standard single-mode fiber (SSMF)*: Most of the installed fiber falls in this category. It was designed to support early long-haul transmission systems, and it has zero dispersion at 1310 nm.
2. *Non-zero dispersion fiber (NZDF)*: This fiber has zero dispersion near 1450 nm.
3. *Negative dispersion fiber (NDF)*: This type of fiber has a negative dispersion in the region 1300 to 1600 nm.
4. *Low water peak fiber (LWPF)*: As shown in Figure 8.14, there is a peak in the attenuation curve at 1385 nm, known as the *water peak*. With this new type of fiber this peak is eliminated, which allows the use of this region.

Single-mode and multi-mode fibers are costly and require a skilled technician to install them. *Plastic optical fibers (POF)*, on the other hand, are inexpensive and can be easily installed by an untrained person. First introduced in 1960s, POFs perform well over distances of less than 30 meters. The core of a plastic optical fiber is made of a general-purpose resin called PMMA; the cladding is made of fluorinated polymers. The core has a very large diameter – about 96% of the cladding's diameter. Plastic optic fibers are used in digital home appliance interfaces; home networks; and mobile environments, such as automobiles.

8.3 COMPONENTS

In the rest of this chapter, we describe some of the components used in WDM optical networks. In the previous section, we talked about launching a light into an optical fiber. In Section 8.3.1, we will see how this light is generated using a laser, and how a data stream is modulated onto the light stream. Curiously, *laser* is not a word! Rather, it is an acronym derived from the name of the underlying technique: *light amplification by stimulated emission of radiation*. In the same section, we will also discuss the concept of dense WDM and the wavelength grid proposed in the ITU-T G.692 standard. In Section 8.3.2, we briefly discuss photo-detectors and optical receivers. In Section 8.3.3, we discuss optical amplifiers and in particular we describe the *Erbium-doped fiber amplifier (EDFA)*, a key technology that enabled the deployment of WDM systems. In Section 8.3.4, we will describe the 2 × 2 *coupler* and the *star-coupler*, and last but not least, in Section 8.3.5, we will describe various technologies used for *optical cross-connects (OXC)*.

8.3.1 Lasers

Let us consider the transmitting side of the WDM link with W wavelengths shown in Figure 8.1, and reproduced again here in Figure 8.16. There are W different transmitters, each transmitting at a different wavelength λ_i, $i = 1, 2, \ldots, W$. The output of a transmitter is modulated by a data stream, and the W modulated outputs are all multiplexed onto the same fiber using an N-to-1 combiner (see Section 8.3.4).

A transmitter is typically a laser, although a *light-emitting diode (LED)* can also be used. There are different types of lasers, of which the semiconductor laser is the most commonly used laser in optical communication systems. Semiconductor lasers are very compact and can be fabricated in large quantities.

A *laser* is a device that produces a very strong and concentrated beam. It consists of an energy source which is applied to a *lasing* material, a substance that emits light in all directions and it can be of gas, solid, or semiconducting material. The light produced by the lasing material is enhanced using a device such as the *Fabry-Perot resonator cavity*. This cavity consists of two partially reflecting parallel flat mirrors, known as *facets*. These mirrors are used to create an optical feedback which causes the cavity to oscillate with a positive gain that compensates for any optical losses. Light hits the right facet and part of it leaves the cavity through the right facet and part of it is reflected (see Figure 8.17). Part of the reflected light is reflected back by the left facet towards the right facet, and again part of it exits through the right-facet and so on.

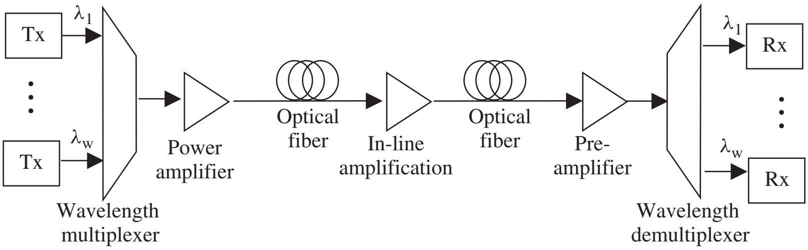


Figure 8.16 A WDM point-to-point link.

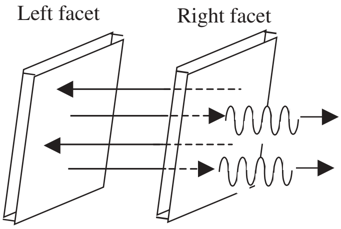


Figure 8.17 The Fabry-Perot resonator cavity.

Consider a wavelength for which the cavity length (i.e., the distance between the two mirrors) is an integral multiple of half the wavelength. That is, the round trip through the cavity is an integral multiple of the wavelength. For such a wavelength, all of the light waves transmitted through the right facet are in phase; therefore, they reinforce each other. Such a wavelength is called a *resonant wavelength* of the cavity.

Since there are many resonant wavelengths, the resulting output consists of many wavelengths spread over a few nm, with a gap between two adjacent wavelengths of 100 GHz to 200 GHz. However, it is desirable that only a single wavelength comes out from the laser. This can be done by using a filtering mechanism that selects the desired wavelength and provides loss to the other wavelengths. Specifically, another cavity can be used after the primary cavity where gain occurs. Using reflective facets in the second cavity, the laser can oscillate only at those wavelength resonant for both cavities.

If we are to make the length of the cavity very small, then only one resonant wavelength occurs. It turns out that such a cavity can be done on a semiconductor substrate. In this case the cavity is vertical with one mirror on the top surface and the other in the bottom surface. This type of laser is called a *vertical cavity surface emitting laser (VCEL)*. Many VCELs can be fabricated in a two-dimensional array.

Tunable lasers

Tunable lasers are important to optical networks, as will be seen in the two subsequent chapters. Also, it is more convenient to manufacture and stock tunable lasers, than make different lasers for specific wavelengths. Several different types of tunable lasers exist, varying from slow tunability to fast tunability.

Modulation

Modulation is the addition of information on a light stream. This can be realized using the *on-off keying (OOK)* scheme. In this scheme, the light stream is turned on or off depending whether we want to modulate a 1 or a 0. OOK can be done using *direct* or *external* modulation. In direct modulation, the light drive current into the semiconductor laser is set above threshold for 1 and below it for a 0. As a result, the presence of high or low power is interpreted as a 1 or 0, respectively. Direct modulation is easy and inexpensive to implement. However, the resulting pulses become *chirped*; that is, the carrier frequency of the transmitted pulse varies in time.

External modulation is achieved by placing an external modulator in front of a laser. The laser continuously transmits but the modulator either lets the light through or stops it accordingly if it is a 1 or a 0. Thus, the presence of light is interpreted as a 1 and no light is interpreted as a 0. An external modulator minimizes chirp.

Dense WDM (DWDM)

In the literature, the term *dense WDM (DWDM)* is often used. This term does not imply a different technology to that used for WDM. In fact, the two terms are used interchangeably. Strictly speaking, *DWDM* refers to the wavelength spacing proposed in the ITU-T G.692 standard. Originally, the wavelengths were separated by wide bands which were several tens or hundreds of nanometers. These band became very narrow as technology improved.

Table 8.2 ITU-T DWDM grid.

Channel code	λ (nm)	Channel code	λ (nm)	Channel code	λ (nm)	Channel code	λ (nm)
18	1563.05	30	1553.33	42	1543.73	54	1534.25
19	1562.23	31	1552.53	43	1542.94	55	1533.47
20	1561.42	32	1551.72	44	1542.14	56	1532.68
21	1560.61	33	1590.12	45	1541.35	57	1531.90
22	1559.80	34	1550.12	46	1540.56	58	1531.12
23	1558.98	35	1549.32	47	1539.77	59	1530.33
24	1558.17	36	1548.52	48	1538.98	60	1529.55
25	1557.36	37	1547.72	49	1538.19	61	1528.77
26	1556.56	38	1546.92	50	1537.40	62	1527.99
27	1555.75	39	1546.12	51	1536.61	–	–
28	1554.94	40	1545.32	52	1535.82	–	–
29	1554.13	41	1544.53	53	1535.04	–	–

ITU-T proposed a set of closely spaced wavelengths in the 1550 nm window. The reason that the 1550 nm window was chosen is due to the fact that it has the smallest amount of attenuation and it also lies in the band where the Erbium-doped fiber amplifier (see Section 8.3.3) operates. The ITU-T proposed guard is 0.8 nm or 100 GHz and the grid is centered in 1552.52 nm or 193.1 THz. The ITU-T grid is given in Table 8.2. The ITU-T grid is not always followed, since there are many proprietary solutions.

Note that this book uses the acronym *WDM*.

8.3.2 Photo-detectors and Optical Receivers

Let us consider the receiving side of the WDM link (see Figure 8.16). The WDM optical signal is demultiplexed into the *W* different wavelengths, and each wavelength is directed to a receiver *Rx*. The demultiplexer is a 1-to-*N* splitter (see Section 8.3.3). Each receiver consists of a photodetector, an amplifier, and a signal-processing circuit. The photodetector senses an optical signal and produces an electrical signal that contains the same information as in the optical signal. The electrical signal is subsequently amplified so that it can be processed electronically by the signal-processing circuit.

8.3.3 Optical Amplifiers

The optical signal looses its power as it propagates through an optical fiber, and after some distance it becomes too weak to be detected (Section 8.2.2). Optical amplification is used to restore the strength of the signal. Optical amplification can be used as power amplifiers, in-line amplifiers, and preamplifiers, in a WDM link (Figure 8.16). The optical signal can be boosted after it leaves the multiplexer, and before it enters the receiver. In-line amplifiers are used for very long transmission WDM links, and they typically boost the power of the signal to compensate for what was lost prior to entering the optical amplifier.

Prior to optical amplifiers, the optical signal was regenerated by first converting it into an electrical signal, then apply *1R* or *2R* or *3R* regeneration, and then converting the regenerated signal back into the optical domain. In 1R, the electrical signal is simply re-amplified, in 2R, the signal is re-amplified and re-shaped, and in 3R, the signal is re-amplified, re-shaped, and re-timed. In order to re-amplify or re-shape an electrical signal, we do not need to have knowledge of its bit-rate and frame format. However, for re-timing knowledge of both the bit-rate and frame format is necessary.

Optical amplification, as in the 1R scheme, can be done without knowledge of the bit rate and the framing format, and it can be applied simultaneously to the combined signal of all of the wavelengths in a WDM link. Currently, re-shaping and re-timing cannot be done in the optical domain. There are several different types of optical amplifiers, such as the *Erbium-doped fiber amplifier (EDFA)*, the *semiconductor optical amplifier (SOA)* and the *Raman* amplifier. Below, we describe the Erbium-doped fiber amplifier, a key technology that enabled the deployment of WDM systems. The SOA is mostly used in *optical cross-connects (OXCs)* and is described in Section 8.3.5.

The Erbium-doped fiber amplifier (EDFA)

The EDFA consists of a length of silica fiber whose core is doped with Erbium, a rare earth element. As shown in Figure 8.18, a laser is emitted into the fiber and is combined through a coupler (see Section 8.3.4) with the signal that needs to be amplified. This laser operates at 980 nm or 1480 nm, and the signal to be amplified is in the 1550 nm window. The signal from the laser *pumps* the doped fiber and induces a stimulated emission of the electrons in the fiber. That is, electrons are induced to transmit from a higher energy level to a lower energy level, which causes the emission of photons, and which in turn amplifies the incoming signal. An isolator is used at the input and/or output to prevent reflections into the amplifier.

In practice, EDFAs are more complex than the one shown in Figure 8.18; the two-stage EDFA shown in Figure 8.19 is much more common. In the first stage, a *co-directional*

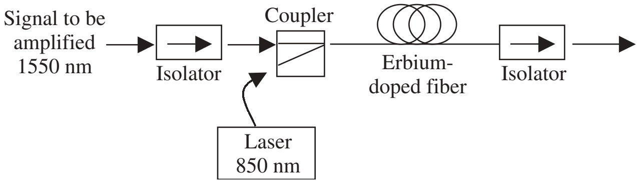


Figure 8.18 The Erbium-doped fiber amplifier.

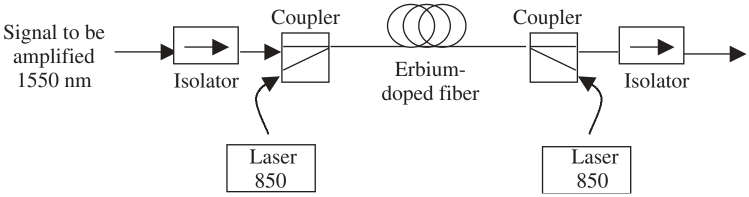


Figure 8.19 A two-stage EDFA.

laser pumps into the coupler in the same direction as the signal to be amplified, and in the second stage, a *counter-directional* laser pumps into the coupler in the opposite direction of the signal to be amplified. Counter-directional pumping gives higher gain, but co-directional pumping gives better noise performance.

8.3.4 The 2 x 2 Coupler

The 2 × 2 coupler is a basic device in optical networks, and it can be constructed in variety of different ways. A common construction is the *fused-fiber* coupler. This is fabricated by twisting together, melting, and pulling two single-mode fibers so that they get fused together over a uniform section of length. Each input and output fiber has a long tapered section (see Figure 8.20).

Let us assume that an input light is applied to input 1 of fiber 1. As the input light propagates through the fiber 1 tapered region into the coupling region, an increasing portion of the input electric field propagates outside of the fiber 1 core and is coupled into fiber 2. A negligible amount of the incoming optical power is reflected back into the fibers. In view of this, this type of coupler is known as a *directional coupler*. The optical power coupled from one fiber to the other can be varied by varying the length of the coupling region, the size of the reduced radius of the core in the coupling region, and the difference in the radii of the two fibers in the coupling region. There is always some power loss when the light goes through the coupler.

A more versatile 2 × 2 coupler is the *waveguide coupler* (see Figure 8.21). A *waveguide* is a medium that confines and guides a propagating electromagnetic wave. A

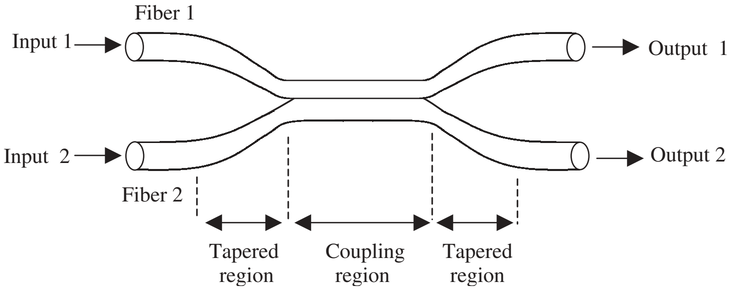


Figure 8.20 A fused-fiber 2 × 2 coupler.

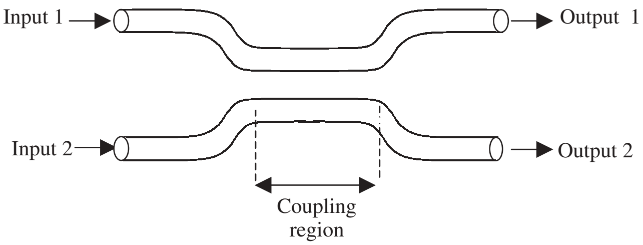


Figure 8.21 A 2 × 2 waveguide coupler.

waveguide coupler has two identical parallel guides in the coupling region. (Alternatively, one guide might be wider than the other.) As in the fused-fiber coupler, part of the light going down one guide is coupled onto the other guide. The degree of interaction between the two guides can be varied through the width of the guide, the gap between the two guides, and the refractive index between the two guides.

Couplers are reciprocal devices. That is, they work exactly in the same way if their inputs and outputs are reversed.

A 2 × 2 coupler is called a *3-dB coupler* when the optical power of an input light applied to, say input 1 of fiber 1, is evenly divided between output 1 and output 2. If we only launch a light to the one of the two inputs of a 3-dB coupler, say input 1, then the coupler acts as a *splitter*. If we launch a light to input 1 and a light to input 2 of a 3-dB coupler, then the two lights will be coupled together and the resulting light will be evenly divided between outputs 1 and 2. In this case, if we ignore output 2, the 3-dB coupler acts as a *combiner*.

A generalization of the 2 × 2 coupler is the *star coupler*. This device combines the power from N inputs and then divides it equally on all of the outputs. One popular method of constructing a star coupler is to use the fiber-fused technology. This involves twisting, heating and stretching N fibers together. The resulting device can be used to split evenly an incoming light to N outputs, or it can combine N incoming lights to a single output, or it can combine N incoming lights and distribute evenly to N outputs. However, due to difficulties in controlling the heating and pulling process, fiber-fused technology is limited to a small number of fibers.

An alternative way to construct a star coupler is to combine 3-dB couplers in a Banyan network (see Figure 8.22). Each box in Figure 8.22 represents a 3-dB coupler. The number of 3-dB couplers required for an $N \times N$ Banyan network is $(N/2)\log_2 N$. Each input is associated with a different wavelength (see Figure 8.22). The star coupler combines all of the wavelengths together and then evenly distributes them on all of the output ports. It can also be used as 1-to-N splitter (i.e., a demultiplexer) or an N-to-1 combiner (i.e., a multiplexer).

8.3.5 Optical Cross-connects (OXCs)

An *optical cross-connect (OXC)* is an $N \times N$ optical switch, with N input fibers and N output fibers. The OXC can switch optically all of the incoming wavelengths of the input fibers to the outgoing wavelengths of the output fibers. For instance, it can switch the

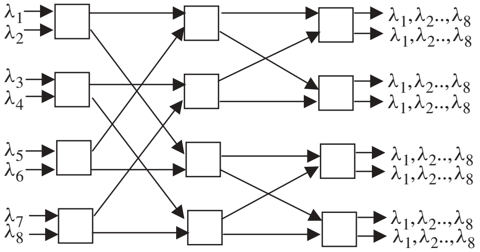

λ_1
λ_2
λ_3
λ_4
λ_5
λ_6
λ_7
λ_8
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$
$\lambda_1,\lambda_2..,\lambda_8$

Figure 8.22 A banyan network of 3-dB couplers.

optical signal on incoming wavelength λ_i of input fiber k to the outgoing wavelength λ_i of output fiber m. If it is equipped with converters, it can also switch the optical signal of the incoming wavelength λ_i of input fiber k to another outgoing wavelength λ_j of the output fiber m. This happens when the wavelength λ_i of the output fiber m is in use. Finally, an OXC can also be used as an *optical add/drop multiplexer (OADM)*. That is, it can terminate the optical signal of a number of incoming wavelengths and insert new optical signals on the same wavelengths in an output port. The remaining incoming wavelengths are switched through as described above.

An OXC consists of amplifiers, multiplexers/demultiplexers, a switch fabric, and a CPU (see Figure 8.23). The CPU is used to control the switch fabric and to run communications-related software, such as routing, signaling, and network management. There are N input and N output optical fibers; each fiber carries W wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_W$. The optical signal from each input fiber is pre-amplified and then it is demultiplexed into the W wavelengths. Each wavelength enters the switch fabric through an input port; the switch fabric then directs each wavelength to an output fiber. The W wavelengths switched to the same output fiber are multiplexed onto the same output fiber, and the multiplexed signal is amplified before it is propagated out onto the link. The switch fabric has *NW* input ports (one per incoming wavelength) and *NW* output ports (one per outgoing wavelength). Figure 8.23 gives an unfolded view of the OXC, with traffic flowing from left to right. Each input fiber i and its corresponding output fiber i (where $i = 1, 2, \ldots, N$) are associated with the same user. That is, the user transmits to the OXC on input fiber i, and receives information from the OXC on output fiber i.

The incoming wavelengths are switched to the output fibers optically, without having to convert them to the electrical domain. In view of this, such an OXC is often referred to as a *transparent switch*. This is in contrast to an *opaque* switch, where switching takes place in the electrical domain. That is, the input optical signals are converted to electrical signals, from where the packets are extracted. Packets are switched using a packet switch, and then they are transmitted out of the switch in the optical domain.

In the example given in Figure 8.23, we see that wavelengths λ_1 and λ_W of input fiber 1 are directed to output fiber N. Likewise, wavelengths λ_1 and λ_W of input fiber N are directed to output fiber 1. Let us assume that incoming wavelength λ_W of input fiber k has

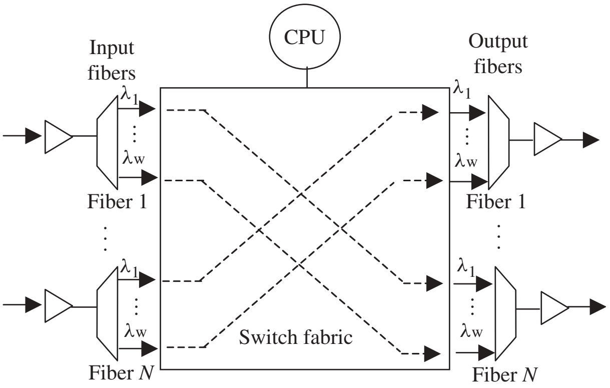


Figure 8.23 A logical diagram of an OXC.

to be directed to output fiber N. As can be seen, this cannot happen since λ_W of output fiber N is in use. (This is known as *external conflict*.) However, if the OXC is equipped with a *wavelength converter*, then the incoming wavelength λ_W can be converted to any other wavelength which happens to be free on output fiber N, so that the optical signal of λ_W can be directed through to output fiber N. Wavelength converters, which can be made using different types of technologies, are very important in optical networks.

OXCs are expected to handle a large number of ports, with a large number of wavelengths per fiber. They are also expected to have a very low switching time. This is the time required to set up the switch fabric so that an incoming wavelength can be directed to an output fiber. The switching time is not critical for permanent connections, but it is critical for dynamically established connections; it is also critical in OBS (see Chapter 10).

An OXC should have a low *insertion loss* and low *crosstalk*. Insertion loss is the power lost because of the presence of the switch in the optical network. Crosstalk occurs within the switch fabric, when power leaks from one output to the other outputs. Crosstalk is defined as the ratio of the power at an output from an input to the power from all other inputs. Finally, we note that an OXC should have low polarization-dependent loss.

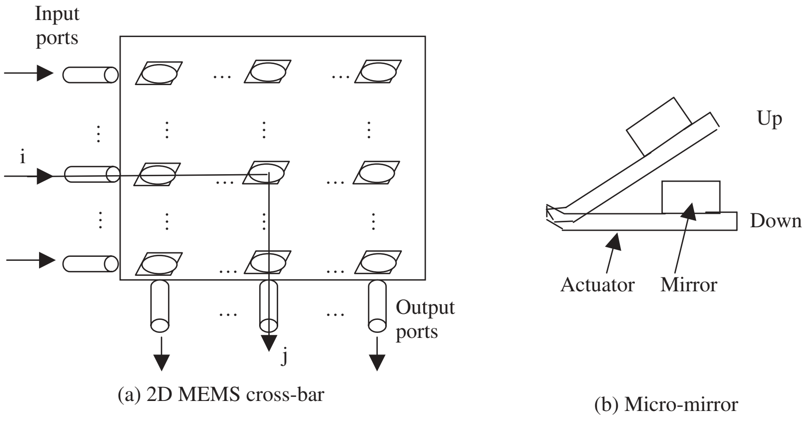
There are several different technologies for building a switch fabric of an OXC, such as multi-stage interconnection networks of directional couplers, digital *micro electronic mechanical systems (MEMS),* and *semiconductor optical amplifier*s *(SOA)*. Other technologies used are micro-bubbles, and holograms. Large OXC switch fabrics can be constructed using 2 × 2 switches arranged in a multi-stage interconnection network, such as a Banyan network and a Clos network. A 2 × 2 switch is a 2 × 2 directional coupler which can direct the optical signal on any input to any output j. There are various types of 2 × 2 switches, such as the *electro-optic switch*, the *thermo-optic switch*, and the *Mach-Zehnder interferometer*. MEMS and SOA are promising technologies for constructing all optical switches, and are described below.

MEMS optical switch fabrics

Micro electronic mechanical systems (MEMS) are miniature electro-mechanical devices that range in dimension from a few hundred microns to millimeters. They are fabricated on silicon substrates using standard semiconductor processing techniques. Starting with a silicon wafer, one deposits and patterns materials in a sequence of steps in order to produce a three-dimensional electro-mechanical structure. MEMS are complex devices, but they are robust, long-lived, and inexpensive to produce. Optical MEMS is a promising technology for constructing all optical switches. Below, we describe a *2D MEMS, 3D MEMS*, and *1D*.

The 2D MEMS optical switch fabric consists of a square array of $N \times N$ micro-mirrors arranged in a crossbar (see Figure 8.24(a)). Each row of micro-mirrors corresponds to an input port, and each column of micro-mirrors corresponds to an output port. Also, each input and output port of the crossbar is associated with a single wavelength. A micro-mirror is indicated by its row number and column number.

A micro-mirror (see Figure 8.24(b)) consists of an actuator and a mirror, and it can be either in the down or up position. For an incoming wavelength on input port i to be switched to output port j, all of the micro-mirrors along the *ith* row, from column 1 to port $j - 1$ have to be in the down position, the micro-mirror in the (i, j) position has to be up, and the micro-mirrors on the *jth* column from rows $i + 1$ to N have to be in the



(a) 2D MEMS cross-bar

(b) Micro-mirror

Figure 8.24 2D MEMS switching fabric.

down position. In this way, the incoming light will be reflected on the *(i, j)th* micro-mirror and redirected to the *jth* output port. The micro-mirrors are positioned so that they are at 45° angle to the path of the incoming wavelengths. The incoming wavelengths have to be *collimated* (i.e., they travel exactly in the same direction).

Micro-mirror control is straightforward, since it is either up or down. The number of micro-mirrors increases with the square of the number of the input and output ports. Therefore, 2D architectures are limited to 32 × 32 ports or 1024 micro-mirrors. The main limiting factors being the chip size and the power loss due to the distance that the light has to travel through the switch.

The 2D MEMS architecture can be used to construct an *optical add/drop multiplexer (OADM)*. This device is connected to a WDM optical link and it can drop (i.e., terminate) a number of incoming wavelengths and insert new optical signals on these wavelengths. The remaining wavelengths of the WDM link are allowed to pass through. The specific wavelengths that it adds/drops can be either statically or dynamically configured. An OADM can also add/drop wavelengths from a number of WDM links.

A logical diagram of an OADM is shown in Figure 8.25. The optical signal on the WDM link is demultiplexed, and each wavelength is directed to the upper input port of a 2 × 2 optical switch. The wavelength is switched to the lower output port of the 2 × 2

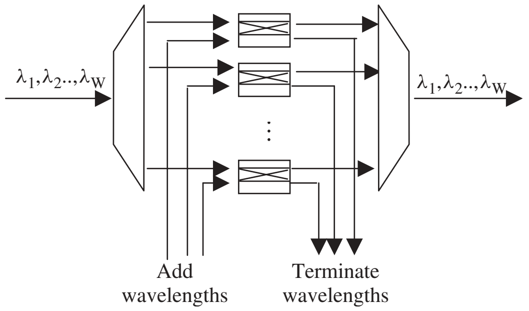


Figure 8.25 A logical design of an OADM.

switch if it is to be dropped, or to the upper output port if it is allowed to pass through. The lower input port of the 2 × 2 optical switch is used for the wavelength to be added in; it is always switched to the upper output port of the switch. There is one 2 × 2 optical switch for each wavelength. For example, assume that wavelength λ_i is to be dropped. Then its 2 × 2 optical switch is instructed to direct the wavelength coming into its upper input port to the lower output port. At the same time, a new data stream is modulated onto the same wavelength λ_i, which is directed to the lower input port of the same 2 × 2 optical switch. This new added wavelength is switched to the upper output port. All of the wavelengths that exit from the upper output ports of the 2 × 2 optical switches are multiplexed and propagated out onto the link.

This OADM can be easily implemented using the 2D MEMS architecture. As shown in Figure 8.26, each micro-mirror is a pair of micro-mirrors, which operate simultaneously. That is, they both go up or down at the same time. Assume that the *ith* incoming wavelength has to be dropped. Then the *(i, i)th* micro-mirror pair is activated; that is, it goes up. All of the other micro-mirrors on the *ith* row and *ith* column are down. The incoming light from the demultiplexer is reflected on one of the two micro-mirrors and is directed into the ith drop port. At the same time, the new wavelength from the *ith* add port is reflected off of the other micro-mirror and is directed to the output port. If the *ith* wavelength is not to be dropped, then all of the micro-mirror pairs on the *ith* row are not activated. That is, they are all down, and the *ith* wavelength simply passes through the OADM uninterrupted.

In the 2D MEMS architecture, all of the light beams reside on the same plane – thus the name of the architecture. In the 3D MEMS architecture, the light beams travel in a three-dimensional space, which allows scaling far beyond 32 ports. In the 3D MEMS architecture, each micro-mirror is gimbaled in two dimensions (see Figure 8.27). The micro-mirror is attached to an inside ring so that it can rotate over the x axis. The inside ring is attached to an outside ring so that it can rotate on the y axis. Using this gimbaled mechanism, the micro-mirror can rotate freely about two axis.

The 3D MEMS architecture is shown in Figure 8.28. There is a MEMS array of micro-mirrors associated with the input fibers, and each micro-mirror is dedicated to an input fiber. Likewise, there is another MEMS array of micro-mirrors associated with the output

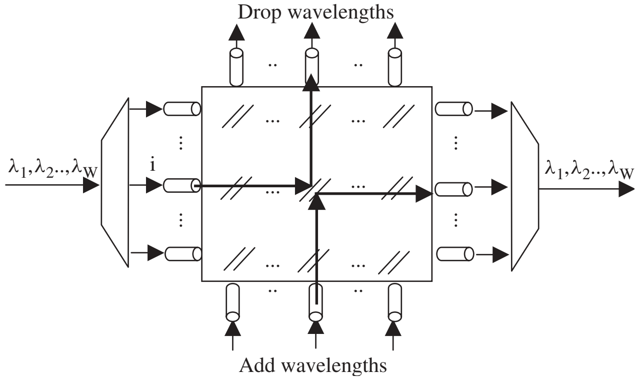

Drop wavelengths
$\lambda_1,\lambda_2..,\lambda_W$
i
$\lambda_1,\lambda_2..,\lambda_W$
Add wavelengths

Figure 8.26 A 2D MEMS OADM.

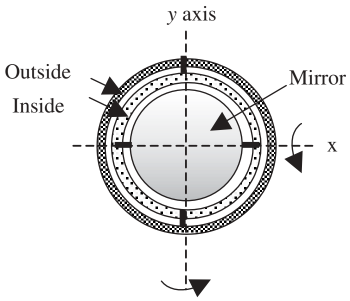


Figure 8.27 The gimbaled mirror.

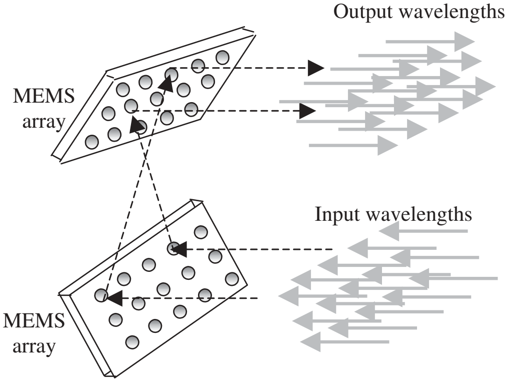


Figure 8.28 The 3D MEMS architecture.

fibers, and each micro-mirror is dedicated to an output fiber. Switching of the incoming light on an input fiber i to an output fiber j is achieved by appropriately tilting their dedicated micro-mirrors so that the incoming light is reflected to the output fiber.

The operation of this MEMS architecture requires a sophisticated control mechanism that it can tilt the angle of the micro-mirrors and maintain them in position for the duration of the connection. The number of micro-mirrors increases with the square root of the number of input and output ports. This permits scaling of the optical switching fabric to thousands of ports with a low insertion loss of about 3 dB.

Finally, the 1D MEMS optical switch fabric uses a single MEMS array. In this architecture, a dispersive optics element is used to separate the input DWDM optical signal into the individual wavelengths. Each wavelength strikes an individual micro-mirror which directs it to the desired output where it is combined with the other wavelengths via a dispersive element. There is one micro-mirror per wavelength, which makes the architecture scale linearly with the number of wavelengths.

Semiconductor optical amplifier (SOA)

A *semiconductor optical amplifier (SOA)* is a *pn-junction* that acts as an amplifier and also as an on-off switch. A pn-junction, as shown in Figure 8.29, consists of a *p-type*

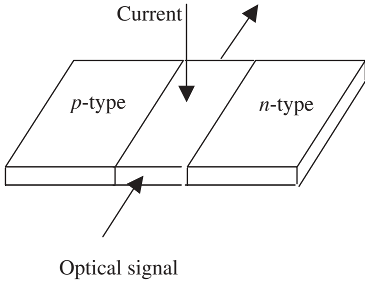


Figure 8.29 A pn-junction.

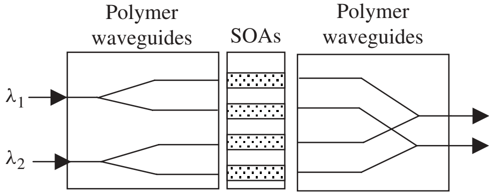


Figure 8.30 A 2 × 2 SOA switch.

and an *n-type* semiconductor material. A p-type semiconductor is doped with impurity atoms so that it has an excessive concentration of mobile electron vacancies, known as *holes*. The n-type semiconductor has an excess concentration of electrons. When p-type and n-type semiconductors are placed in contact with each other, the holes move from the p-type semiconductor to the n-type semiconductor and the electrons move from the n-type semiconductor to the p-type semiconductor. This creates a region with negative charge in the p-type semiconductor, and a region with positive charge in the n-type semiconductor. These two regions are known as the *depletion area*. When charge is applied, the width of the depletion region is reduced and current flows from the p-type semiconductor to the n-type semiconductor causing the depletion area to become an active region. An optical signal is amplified if it passes through the depletion area when it is active. When no charge is applied to the pn-junction, the optical light that passes through the depletion area is absorbed.

SOAs can be used to build optical switches. A 2 × 2 switch is shown in Figure 8.30. The waveguides act as splitters or couplers, and are made of polymeric material. The incoming wavelength λ_1 is split into two optical signals, and each signal is directed to a different SOA (see Figure 8.30). One SOA amplifies the optical signal and permits it to go through, and the other SOA stops it. Accordingly, wavelength λ_1 leaves the 2 × 2 switch from either the upper or the lower output port. The switching time of an SOA switch is currently about 100 psec.

PROBLEMS

1. Let us assume that in a WDM point-to-point link each wavelength is used to transmit SONET/SDH frames at the rate of OC-48/STM-16 (i.e., 2488 Mbps). Calculate the total capacity of the link for $W = 1$, 16, 32, 128, 512, 1024. Repeat these calculations assuming that the rate

of transmission over a single wavelength is: OC-192/STM-64 (9953 Mbps), OC-768/STM-256 (39,813 Mbps).

2. Browse the Internet to find out the maximum number of wavelengths which is currently commercially available in a WDM point-to-point link.
3. What are the differences between a single-mode and a multi-mode fiber? The standard for the 1-Gbps Ethernet makes use of both single-mode and multi-mode fibers. Browse the Internet to find out how each of these fiber modes are used.
4. Explain what is attenuation and dispersion.
5. Explain the terms transparent switch and opaque switch.
6. Draw a three-stage Clos network. What are the main differences between a Banyan network and a Clos network? (Hint: for information, check the literature on ATM switch architectures.)
7. Use the 2D MEMS OADM shown in Figure 8.26 to design an OADM that serves a single fiber with 64 wavelengths. Each 2D MEMS is assumed to have 32×32 ports.

9

Wavelength Routing Optical Networks

Wavelength routing optical networks have been successfully commercialized and standards bodies, such as the IETF, OIF, and ITU-T, are currently active in the development of the standards. A *wavelength routing optical network* consists of *optical cross-connects (OXCs)* interconnected with WDM fibers. Transmission of data over this optical network is done using optical circuit-switching connections, known as *lightpaths*.

In this chapter, we explore different aspects of the wavelength routing optical networks. We first start with a description of the main features of a wavelength routing network and introduce the ever important concept of a lightpath and the concept of *traffic grooming*, which permits multiple users to share the same lightpath. We also present protection and restoration schemes used to provide carrier grade reliability.

Information on a lightpath is typically transmitted using SONET/SDH framing. Ethernet frames can also be transmitted over an optical network. In the future, it is expected that information will be transmitted over the optical network using the new ITU-T G.709 standard, part of which is described in this chapter. G. 709, also known as the *digital wrapper*, permits the transmission of IP packets, Ethernet frames, ATM cells, and SONET/SDH data over a synchronous frame structure.

The rest of the chapter is dedicated to the control plane for wavelength routing networks. We present different types of control plane architectures, and then describe the *generalized MPLS (GMPLS)* architecture and the OIF *user network interface (UNI)*. GMPLS is an extension of MPLS, and was designed to apply MPLS label-switching techniques to *time-division multiplexing (TDM)* networks and wavelength routing networks, in addition to packet-switching networks. The OIF UNI specifies signaling procedures for clients to automatically create and delete a connection over a wavelength routing network. The UNI signaling has been implemented by extending the label distribution protocols, LDP and RSVP.

9.1 WAVELENGTH ROUTING NETWORKS

A wavelength routing (or routed) network consists of OXCs interconnected by WDM fibers. An OXC is an $N \times N$ optical switch, with N input fibers and N output fibers (see Section 8.3.5). Each fiber carries W wavelengths. The OXC can optically switch all of the incoming wavelengths of its input fibers to the outgoing wavelengths of its output

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

fibers. For instance, it can switch the optical signal on incoming wavelength λ_i of input fiber k to the outgoing wavelength λ_i of output fiber m. If output fiber m's wavelength λ_i is in use, and if the OXC is equipped with converters, then the OXC can also switch the optical signal of input fiber k's incoming wavelength λ_i to another one of output fiber m's outgoing wavelength λ_j.

In addition to switching individual wavelengths, an OXC can switch a set of contiguous wavelengths (known as a *waveband*) as a single unit. That is, it can switch a set of contiguous wavelengths of an input fiber to a set of contiguous wavelengths of an output fiber. This can be a desirable OXC feature, because it can reduce the distortion of the individual wavelengths. In addition, an OXC might not have the capability to separate incoming wavelengths that are tightly spaced. In this case, it can still switch them using waveband switching. Finally, an OXC can also switch an entire fiber. That is, it can switch all of the W wavelengths of an input fiber to an output fiber.

There are several technologies available for building an OXC, such as multistage interconnection networks of 3-dB couplers, MEMS, SOA, micro-bubbles and holograms (see Section 8.3.5). New technologies are expected to emerge in the future.

An OXC can be used as an *optical add/drop multiplexer (OADM)*. That is, it can terminate the signal on a number of wavelengths and insert new signals into these wavelengths. The remaining wavelengths are switched through the OXC transparently. An example of an OADM is shown in Figure 9.1. One of its output fibers is fed into a SONET/SDH DCS. Typically, a SONET/SDH frame is used to transmit data on each wavelength. The DCS converts the incoming W optical signals into the electrical domain, and extracts the SONET/SDH frame from each wavelength. It can then switch time slots from a frame of one wavelength to the frame of another, terminate virtual tributaries, and add new ones. The resulting W new streams of SONET/SDH frames are transmitted to the OXC, each over a different wavelength, and then are switched to various output fibers.

9.1.1 Lightpaths

An important feature of a wavelength routing network is that it is a circuit-switching network. That is, in order for a user to transmit data to a destination user, a connection has to be first set up. This connection is a circuit-switching connection and is established by using a wavelength on each hop along the connection's path. For example, let us consider that two IP routers (router A and router B) are connected via a three-node wavelength

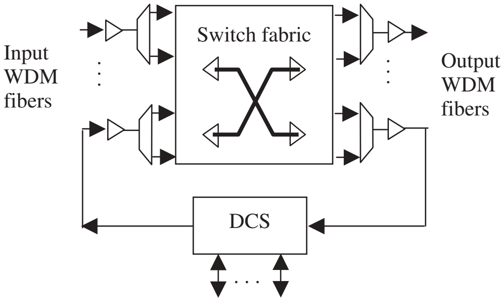
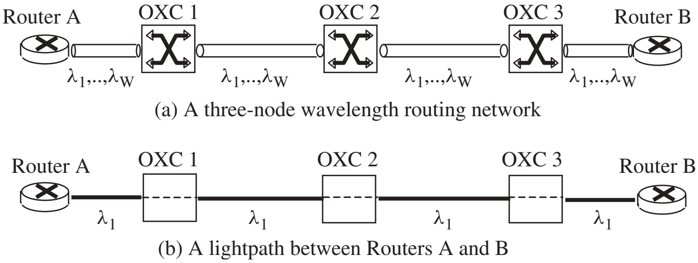


Figure 9.1 An OXC with an attached SONET/SDH DCS.



(a) A three-node wavelength routing network

(b) A lightpath between Routers A and B

Figure 9.2 A lightpath.

routing network (see Figure 9.2). The links from router A to OXC 1, from OXC 1 to OXC 2, from OXC 2 to OXC 3, and from OXC to router B, are assumed to be a single fiber carrying W wavelengths, which are referred to as $\lambda_1, \lambda_2, \ldots, \lambda_W$. Data is transmitted unidirectionally, from router A to router B. To transmit data in the opposite direction (i.e. from router B to router A), another set of fibers would need to be used.

Assume that IP router A wants to transmit data to IP router B. Using a signaling protocol, A requests the establishment of a connection to B. The connection between Routers A and B is established by allocating the same wavelength (say wavelength λ_1) on all of the links along the path from A to B (i.e., links A to OXC 1, OXC 1 to OXC 2, OXC 2 to OXC 3, and OXC 3 to B). Also, each OXC is instructed to switch λ_1 through its switch fabric transparently. As a result, an optical path is formed between Routers A and B, over which data is transmitted optically from A to B. This optical path is called a *lightpath*, and it connects Routers A and B in a unidirectional way from A to B. In order for B to communicate with A, a separate lightpath has to be established in the opposite way over a different set of fibers that are set up to transmit in the opposite direction.

When establishing a lightpath over a wavelength routing network, the same wavelength has to be used on every hop along the path. This is known as the *wavelength continuity constraint*. The required wavelength might not be available at the outgoing fiber of an OXC, through which the lightpath has to be routed. In this case, the establishment of the lightpath will be blocked, and a notification message will be sent back to the user requesting the lightpath. To decrease the possibility that a lightpath is blocked, the OXC can be equipped with converters. A *converter* can transform the optical signal transmitted over a wavelength to another wavelength. In an OXC, for each output fiber with W wavelengths, there might be c converters, where $0 \leq c \leq W$. When $c = 0$, we say that there is *no conversion*; when $0 < c < W$, we say that there is *partial conversion*; and when $c = W$, we say that there is *full conversion*. Converters are still expensive, and so they may be deployed in certain strategic OXCs in a wavelength routing network. A common assumption made in the literature is that a converter can transform a signal on a wavelength λ to *any* wavelength. However, currently, it can only transform it to another wavelength which is within a few nm from wavelength λ.

An example of different lightpaths established over a wavelength routing network is shown in Figure 9.3. The optical network consists of OXCs 1, 2, and 3. Only OXC 3 is assumed to be equipped with converters (at least two, for the sake of this example). IP Routers A and B are attached to OXC 1 at two different input ports; IP router C is attached to OXC 2; and IP router D is attached to OXC 3. The following lightpaths

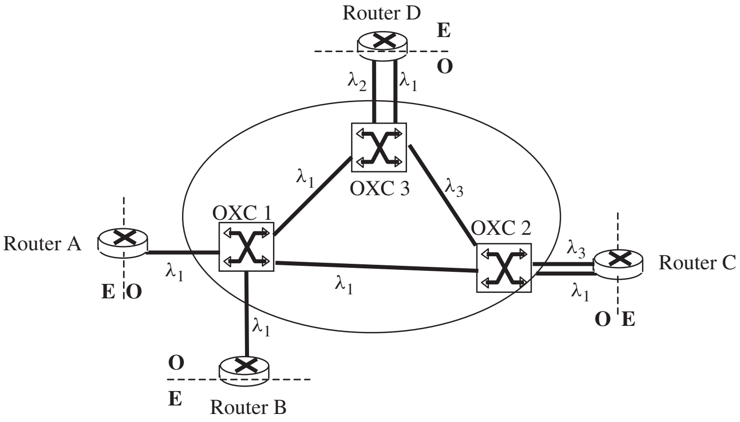


Figure 9.3 An example of different lightpaths.

have been established: from router A to router C over OXCs 1 and 2; from router B to router D over OXCs 1 and 3; and from router C to router D over OXCs 2 and 3. The wavelengths allocated to each lightpath are indicated in Figure 9.3. Wavelength λ_1 is used for the lightpath from router A to router C on all of the hops; that is, from A to OXC 1, then from OXC 1 to OXC 2, and finally from OXC 2 to C. The lightpath from router B to router D uses λ_1 on the hops from B to OXC 1 and from OXC 1 to OXC 3, and λ_2 on the hop from OXC 3 to D. Finally, the lightpath from router C to router D uses λ_3 on the hops from C to OXC 2 and then from OXC 2 to OXC 3, and λ_1 on the hop from OXC 3 to D.

As mentioned above, the transmission on a lightpath is unidirectional. In Figure 9.3, only the lightpaths from Routers A to C, B to D, and C to D are shown. For bidirectional communication between two routers, a separate lightpath has to be set up in the opposite direction through the same OXCs. For instance, for bidirectional communication between Routers A and C, another lightpath has to be set up from router C to router A via OXCs 2 and 1 using separate fiber links. Finally, note that the data transmission within the wavelength routing network occurs entirely in the optical domain. (In Figure 9.3, the dotted lines along the IP routers signify the boundary between the electrical domain [E] and the optical domain [O].)

Lightpaths can be either *static* (e.g. in an ATM PVC connection) or *dynamic* (e.g. in an ATM SVC connection). Static lightpaths are established using network management procedures, and generally remain up for a long time. *Virtual private networks (VPNs)* can also be set up using static lightpaths. Dynamic lightpaths are established in real-time using signaling protocols, such as IETF's GMPLS and the *user network interface (UNI)* proposed by the *Optical Internetworking Forum (OIF)*. These protocols are discussed in detail in Sections 9.5 and 9.6 below.

9.1.2 Traffic Grooming

A lightpath is used exclusively by a single client. Quite often, the bandwidth that a client requires is a lot less than the wavelength's bandwidth, which means that part of the lightpath's bandwidth goes unused. To resolve this, the bandwidth of a lightpath is divided into *subrate* units, so that it can carry traffic streams transmitted at lower rates. A client can request one or more of these subrate units. This technique, known as *traffic grooming*, allows the bandwidth of a lightpath to be shared by many clients. Compared to using an entire lightpath, traffic grooming improves wavelength utilization and client cost-savings.

As an example, let us consider the six-node optical network (see Figure 9.4). Information is transmitted over the optical network using SONET/SDH framing with a transmission rate of OC-48/STM-16 (2.488 Gbps). A lightpath, indicated by a dotted line, has been established from OXC 1 to OXC 3 through OXC 2 using wavelength λ_1. The subrate unit is an OC-3/STM-1 (155 Mbps), which means that 16 subrate units of OC-3/STM-1 are available on the lightpath. A user, attached to OXC 1, who wants to transmit data to another user, attached to OXC 3, can request any integer number of OC-3/STM-1 subrate units up to a total of 16. If the traffic between these two OXCs exceeds 2.488 Gbps, then more lightpaths can be established.

A lightpath can be seen as a tunnel between the originating and terminating OXCs. That is, the data streams transmitted on the lightpath between OXCs 1 and 3, can only originate at OXC 1 and terminate at OXC 3. No data can be added to the lightpath or dropped from the lightpath at OXC 2.

As explained in the previous section, if no conversion is available, then the same wavelength has to be allocated to a lightpath on all hops. However, the wavelength continuity constraint is not necessary if conversion is available. For example, the lightpath between OXCs 1 and 3 uses the same wavelength because it was assumed that OXC 2 is not equipped with converters. However, if OXC 2 is equipped with converters, then the lightpath can be established by using any wavelength on the links OXC 1 to 2, and OXC 2 to 3.

Finally, a data stream might traverse more than one lightpath in order to reach its destination. For example, assume that a user attached to OXC 1 requests four subrate OC-3/STM-1 units to transmit an OC-12/STM-4 (622 Mbps) data stream to a user attached to OXC 4. In this case, a new lightpath has to be established between OXC 1 and 4, possibly over OXCs 6 and 5. Assume that a lightpath between OXCs 3 and 4 already exists. This lightpath (shown in Figure 9.4 by a dotted line) is routed through OXC 5 and uses wavelength λ_2. In this case, the OC-12/STM-4 data stream will be routed to OXC 3 over the lightpath from OXC 1 to 3, and then to OXC 4 over the lightpath from OXC 3 to 4. This solution assumes that there is available capacity on both lightpaths to carry the 622-Mbps data stream. Also, it assumes that OXC 3 has a SONET/SDH DCS

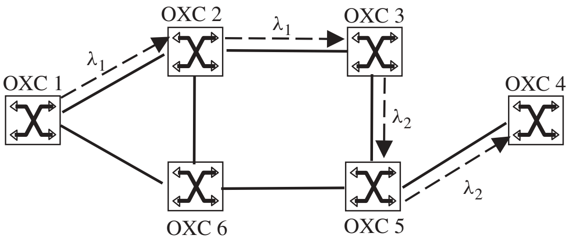


Figure 9.4 An example of traffic grooming.

that permits OXC 3 to extract the data stream from the incoming SONET/SDH frames on the first lightpath and to move the data stream into the SONET/SDH frames of the second lightpath.

Traffic-groomed lightpaths are conceptually similar to virtual path connections in ATM (see Section 4.6.4). A network operator can use them to provide subrate transport services to the users by adding a virtual network to the optical network.

9.2 PROTECTION SCHEMES

Optical networks will be used by telecommunications companies and other network providers, which typically require a *carrier grade* reliability. That is, the network has to be available 99.999% of the time, which translates to an average downtime for the network of six minutes per year!

In this section, we deal with protection schemes against failures of hardware components in an optical network. Link failures are the most common and occur when a fiber cable is accidentally cut when digging in an area through which fiber cables pass. ("Call before you dig," is in fact a real warning and heavy penalties are levied if one accidentally cuts a fiber cable!) A link can also fail if an amplifier that boosts the multiplexed signal of all of the wavelengths on a fiber fails. An individual wavelength within a fiber can also fail if its transmitter or receiver fails. Finally, an OXC can fail – but this is quite rare due to built-in redundancies.

Protection can be performed at the level of an individual lightpath or at the level of a single fiber. *Path protection* denotes schemes for the restoration of a lightpath, and *link protection* denotes schemes for the restoration of a single fiber, whereby all of the wavelengths are restored simultaneously. (Note that *link protection* is referred to in Section 2.6 as *line protection.*)

Below, we examine path and link protection schemes for point-to-point links, WDM optical rings, and mesh wavelength routing optical networks. (For this, a working knowledge of self-healing SONET/SDH rings is required; if needed, review the material in Sections 2.5 and 2.6.)

9.2.1 Point-to-point Links

The simplest optical network is a point-to-point WDM link that connects two nodes. Link protection can be done in a dedicated $1 + 1$ manner, or in a non-dedicated 1:1 or 1:N manner (see Section 2.6). In the $1 + 1$ scheme, the signal is transmitted simultaneously over two separate fibers, that are preferably *diversely routed* (that is, they follow different geographical paths). The receiver monitors the quality of the two signals and selects the best of the two. If one fiber fails, then the receiver continues to receive data on the other fiber. In the 1:1 scheme, there are still two diversely routed fibers, a *working fiber* and a *protection fiber.* The signal is transmitted over the working fiber. If it fails, then the source and destination both switch to the protection fiber. The 1:N scheme is a generalization of the 1:1 scheme, whereby N working fibers are protected by a single protection fiber. Since there is one protection fiber, only one working fiber can be protected at any time.

9.2.2 WDM Optical Rings

WDM optical rings can be seen as an extension of the SONET/SDH rings in the WDM domain. Many different WDM ring architectures have been proposed, varying from simple

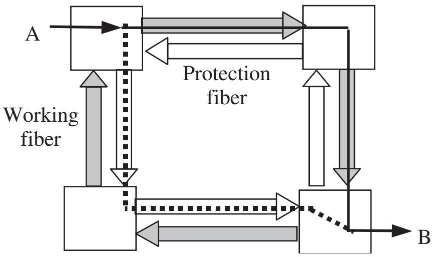


Figure 9.5 An optical unidirectional path sharing ring (OUPSR).

static rings to advanced dynamic rings. Below, we examine the protection scheme of three WDM rings: the *optical unidirectional path sharing ring (OUPSR)*, the *two-fiber optical bidirectional link sharing ring (2F-OBLSR)*, and the *four-fiber optical bidirectional link sharing ring (4F-OBLSR)*.

OUPSR is unidirectional. It consists of a working and a protection ring transmitting in opposite directions (see Figure 9.5). The $1 + 1$ protection scheme is used to implement a simple path protection scheme. That is, a lightpath is split at the source node and is transmitted over the working and protection rings (see Figure 9.5 from A to B). The destination selects the best signal. When a fiber link is broken, the receiver continues to receive the signal along the other path. The OUPSR provides a simple and robust architecture without complex protection signaling protocols. This type of ring is typically used as a metro edge ring, and it connects a small number of nodes (e.g. access networks and customer sites) to a *hub node*, which is attached to a metro core ring. The traffic transmitted on the ring is static and it exhibits hub behavior. That is, it is directed from the nodes to the hub and from the hub to the nodes. Static lightpaths are used.

The two-fiber and four-fiber optical bidirectional link shared rings are used in the metro core where the traffic patterns dynamically change. A signaling protocol is used to establish and tear down lightpaths, and protection schemes are implemented using a real-time distributed protection signaling protocol known as the *optical automatic protection switching (optical APS)*.

The *two-fiber optical bidirectional link shared ring (2F-0BLSR)* uses two rings, transmitting in opposite directions (as in OUPSR). Each fiber is partitioned into two sets of wavelengths; one set of working wavelengths and one set of protection wavelengths. If a fiber fails, the traffic will be rerouted onto the protection wavelengths of the other fiber.

The *four-fiber optical bidirectional link shared ring (4F-OBLSR)* uses two working fibers and two protection fibers (see Figure 9.6). Protection can be done at both the fiber level or at the lightpath level. Fiber protection switching is used to restore a network failure caused by a fiber cut or a failure of an optical amplifier. Lightpath protection switching is used to restore a lightpath that failed due to a transmitter or receiver failure.

Let us consider a lightpath from user A to user B (see the solid line in Figure 9.6). This lightpath is routed through nodes 1, 2, and 3. Assume that the lightpath fails on the link between nodes 2 and 3. In this case, the protection mechanism will switch the lightpath over to the protection fiber from nodes 2 to 3 (see the dotted line labeled “span switching” in Figure 9.6). If the working fiber from nodes 2 to 3 fails as well, then all of the lightpaths will be switched onto its protection fiber from nodes 2 to 3, as in the

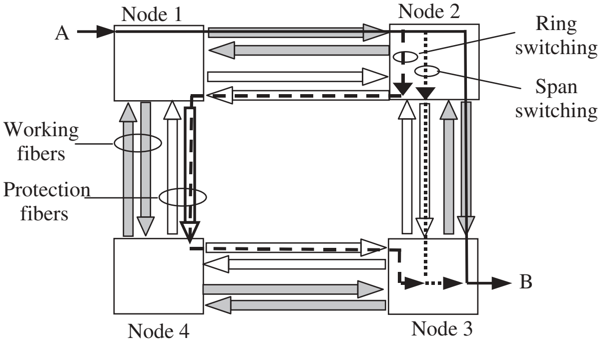


Figure 9.6 A four-fiber optical bidirectional link sharing ring (4F-OBLSR).

case of the lightpath above. This is known as *span switching*. When all four fibers are cut between nodes 2 and 3, then the traffic will be diverted to the working fibers in the opposite direction. This is known as *ring switching*. In this case, the lightpath from A to B will be diverted; that is, it will be routed back to node 1, and then to nodes 4 and 3. (See the dotted line labeled “ring switching” in Figure 9.6.)

9.2.3 Mesh Optical Networks

A mesh network can employ both path and link protection. Link protection can be implemented using the point-to-point $1 + 1$, 1:1, and 1:N schemes (see Section 9.2.1). Path protection uses dedicated or shared back-up paths. Alternatively, an arbitrary mesh topology can be organized into a set of WDM optical rings, which permits ring-based protection schemes.

The $1 + 1$ path protection scheme is the simplest form of protection. It is also the most expensive and bandwidth-inefficient. The user signal is split into two copies, and each copy is transmitted simultaneously over two separate lightpaths. The lightpaths might be diversely routed (i.e. they follow different geographical paths) or they might go through the same OXCs but use different fibers. The receiver monitors the quality of the two signals and selects the best of the two. If one lightpath fails, then the receiver continues to receive data on the other lightpath.

In the case of the 1:1 path protection, the user signal is carried over a working lightpath. The back-up protection lightpath has also been established, but it is not used. If the working lightpath fails, the source and destination switches to the protection lightpath. Since the bandwidth allocated to the protection lightpath is not utilized during normal operation, it can be shared by multiple working lightpaths. This is the 1:N path protection scheme.

An important concept in these protection schemes is the concept of the *shared risk link group (SRLG)*. An SRLG is a group of links that share the same physical resources, such as a cable, a conduit, and an OXC. Failure of these physical resources will cause failure of all of the links. Each common physical resource is associated with an identifier called the SRLG. When setting up a working and a protection lightpath, care is taken so that the two lightpaths are not routed through the same SRLG. For example, let us consider the optical network shown in Figure 9.7. The working lightpath from OXC 1 to OXC 2

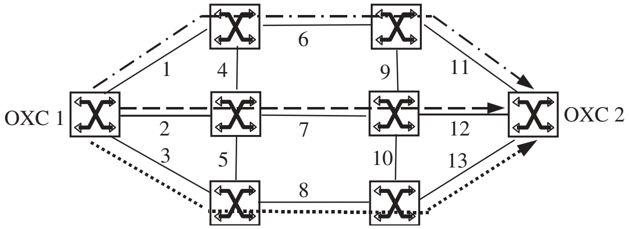


Figure 9.7 Path protection.

that uses links {1, 6, 11} and its protection lightpath that uses links {3, 8, 13} do not use the same SRLG. That is, they are SRLG-disjoint.

The concept of SRLG can also be used in the 1:N shared protection scheme. For instance, in Figure 9.7, the two working lightpaths {1, 6, 11} and {2, 7, 12} from OXC 1 to OXC 2 are SRLG-disjoint. Therefore, it makes sense that they both use the same SRLG-disjoint protection lightpath {3, 8, 13}. This is because a single failure of a physical resource along the path of either working lightpaths (excluding the originating and terminating OXCs) will not cause both working lightpaths to fail at the same time. That is, in this case, the protection lightpath will only be used by one of the two working lightpaths.

In protection schemes, the backup protections routes are pre-planned and the necessary resources (e.g. wavelengths, fibers, and bandwidth within an OXC) are allocated in advance. During the normal operation of the network, these resources are either kept idle, or they are used to transmit low priority traffic which can be preempted any time a failure occurs. This guarantees a fast recovery from a failure at the expense of inefficient resource utilization. An alternative strategy, known as *dynamic restoration*, is to calculate a protection path and allocate resources for recovery at the moment when a network failure occurs. This approach has a more efficient resource utilization but the recovery time is longer than in the case of a protection scheme. Dynamic restoration is a promising new approach that is being further studied.

9.3 THE ITU-T G.709 STANDARD – THE DIGITAL WRAPPER

Information on a lightpath is typically transmitted using SONET/SDH framing. Also, Ethernet frames can be transmitted over an optical network. In the future, it is expected that information will be transmitted over the optical network using the new ITU-T G.709 standard, otherwise known as the *digital wrapper*. This standard defines the network node interfaces between two optical network operators, or between subnetworks of vendors within the same network of an operator. The following are some of the features of the G.709 standard:

- *Types of traffic*: The standard permits the transmission of different types of traffic, such as IP packets and gigabit Ethernet frames using the *generic framing procedure (GFP)*, ATM cells, and SONET/SDH synchronous data.
- *Bit-rate granularity*: G.709 provides for three bit-rate granularities: 2.488 Gbps, 9.95 Gbps, and 39.81 Gbps. This granularity is coarser than that of SONET/SDH, but is appropriate for terabit networks, since it avoids the large number of low bit-rate paths that would have to be used with SONET/SDH.

- *Connection monitoring*: G.709 also provides for connection monitoring capabilities that go beyond those of SONET/SDH. Specifically, unlike SONET/SDH, it is possible to monitor a connection on an end-to-end basis over several carriers, as well as over a single carrier.
- *Forward error correction (FEC)*: As transmission rates increase to 10 Gbps and beyond, the physical parameters of the optical fiber play a significant role in the degradation of the transmitted optical signal. FEC can be used to detect and correct bit errors caused by physical impairments in the transmission links. FEC enables transmission at higher rates without degraded performance. It is useful for under-water transoceanic cables and intra-continental long-haul links.

In ITU-T, an optical network is referred to as the *optical transport network (OTN)*. It consists of three layers: the *optical channel (Och)*, the *optical multiplex section (OMS)*, and the *optical transmission section (OTS)*. (See Figure 9.8.) The optical channel is an optical connection between two users, and it takes up an entire lightpath. Optical channels are multiplexed and transmitted as a single signal over a fiber. The section between a multiplexer and a demultiplexer over which the multiplexed signal is transported is referred to as the optical multiplex section. Finally, the transport between two access points over which the multiplexed signal is transmitted is referred to as the optical transmission section. Each of the OTN layers is associated with a frame structure and appropriate overheads. Below, we examine the payload and overhead fields of the optical channel frame.

9.3.1 The Optical Channel (Och) Frame

The user data is transmitted in frames which contain several different types of overhead, the user payload, and the *forward error correction (FEC)*, as shown in Figure 9.9. The optical channel overheads are shown in Figure 9.10. The client's payload is encapsulated with the *Och payload unit (OPU)* overhead which includes information related to the type of traffic submitted by the user. The resulting OPU is then encapsulated with the *Och*

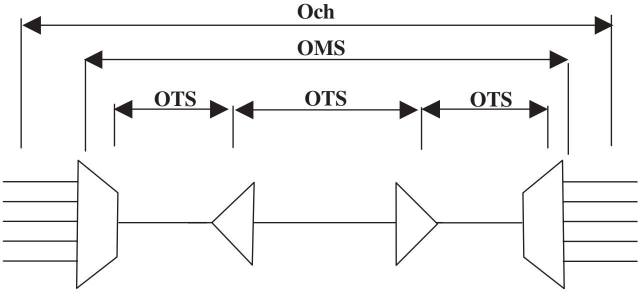


Figure 9.8 The OTN layer structure.

Och overhead	Och payload	FEC

Figure 9.9 The optical channel (Och) frame.

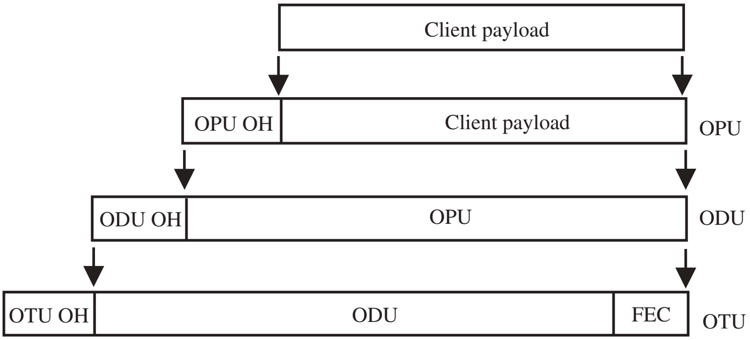


Figure 9.10 The optical channel (Och) overheads.

data unit (ODU) overhead, which provides information for tandem connection monitoring, and end-to-end path supervision. Finally, the resulting ODU is encapsulated with the *Och transport unit (OTU)* overhead, which includes information for the monitoring of the signal on a section. The ODU is also encapsulated with the FEC.

As in the SONET/SDH frame, the OTU frame is arranged in a matrix consisting of four rows of 4080 bytes each (see Figure 9.11). The data is transmitted serially row by row starting from the left of the first row. Recall that a SONET/SDH frame is transmitted every 125 μsec. Higher transmission rates in SONET/SDH are achieved by increasing the size of the SONET/SDH frame. Unlike SONET/SDH, the OTU frame size does not change as the transmission speed increases. Higher transmission rates are achieved by simply increasing the transmission rate of the OTU frame. This is a main departure from the traditional 125 μsec concept, that has been used in communication networks. Three transmission rates have been defined for the transmission of OTU frames: 2.488 Gbps, 9.95 Gbps, and 39.81 Gbps. The time to transmit an OTU frame is 48.971 μsecs when the transmission rate is 2.488 Gbps, 12.191 μsecs when the transmission rate is 9.95 Gbps, and 3.035 μsecs when the transmission rate is 39.81 Gbps.

9.3.2 Overhead Types

The OPU overhead

The *OPU overhead* fields are located in rows 1 to 4 and columns 15 and 16. They provide information related to the client signal; that is, the data transmitted by the user. This overhead is created at the point where the client signal is originated, and it is used at

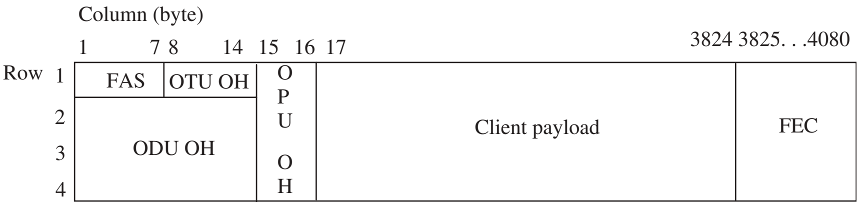


Figure 9.11 The format of the OTU frame.

the point where the client signal is terminated. All bytes are reserved except the *payload structure identifier (PSI)* byte, located on row 4 and column 15. This field is used to transport a 256-byte message over a multi-frame (see MFAS below). The first byte of this message contains the *payload type (PT)* which is used to identify the type of payload carried in the OPUU.

The ODU overhead

The ODU overhead fields are located on rows 2 to 4 and columns 1 to 14. It provides two important overheads: the *path monitoring (PM)* overhead, and the *tandem connection monitoring (TCM)* overhead. The ODU path monitoring overhead enables the monitoring of particular sections within the network as well as fault location in the network. The tandem connection monitoring enables signal management across multiple networks. As shown in Figure 9.12, the following fields have been defined:

- *RES*: Reserved
- *TCM/ACT*: Activation/deactivation of the TCM fields
- *TCMi*: Tandem connection monitoring of ith connection
- *FTFL*: Fault type and fault location reporting channel
- *PM*: Path monitoring
- *EXP*: Reserved for experimental purposes
- *GCC*: General communication channel
- *APS/PCC*: Automatic protection switching and protection communication channel

The *path monitoring (PM)* overhead occupies columns 10, 11, and 12 of the third row. Byte 10 carries the trail trace identifier, which is used to identify the signal from the source to the destination. This is similar to J0 in SONET. Byte 11 carries the result of the BIP-8, computed over the whole OPU and inserted two frames later. (The computation of BIP-8 is described in Section 2.3.2.)

The *tandem connection monitoring (TCM)* overhead occupies columns 5 to 13 of the second row, and columns 1 to 9 of the third row. The TCM functionality enables a

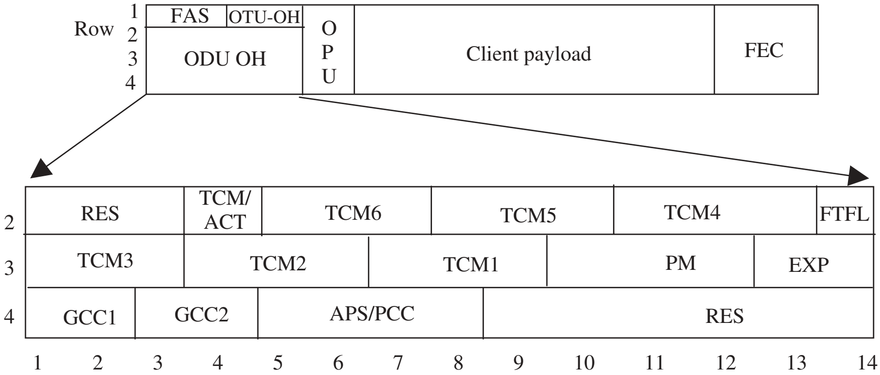


Figure 9.12 The ODU overhead fields.

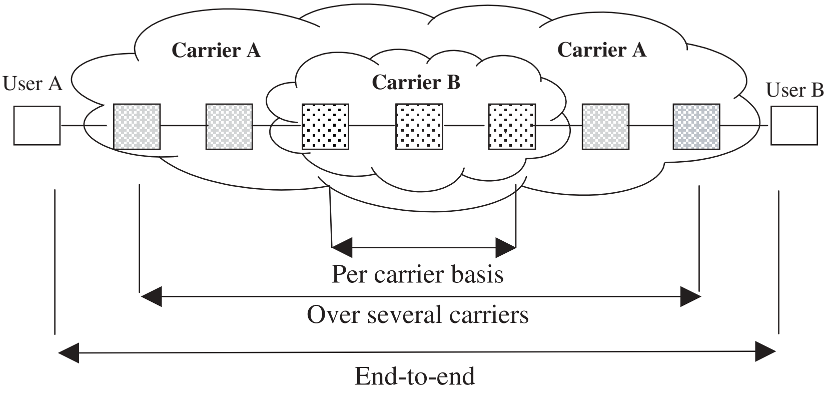


Figure 9.13 An example of networking monitoring.

network operator to monitor the error performance of a connection that originates and terminates within its own network, but traverses different operators. An example of such a connection is shown in Figure 9.13.

FAS and OTU overhead

The *frame alignment signal (FAS)* fields are located in columns 1 to 7 of the first row, as shown in Figure 9.14. FAS is carried in the first six bytes and is used by the receiving equipment to identify the beginning of the OTU frame. The FAS value is the same as in SONET/SDH (i.e., F6F6F6282828), and is transmitted unscrambled. Some of the overheads are transmitted over successive OTU frames. For instance, as we saw above, the payload structure identifier byte of the OPU overhead (located on row 4 and column 15) is used to transport a 256-byte message. In view of this, groups of successive ODU frames are organized logically into multi-frames. The position of an ODU frame within a multi-frame is indicated by the *multi-frame alignment signal (MFAS)* byte located in row 1 column 7. The value of the MFAS byte is incremented each frame, thereby providing a multi-frame consisting of 256 frames. It is transmitted scrambled along with the remainder of the OTU frame.

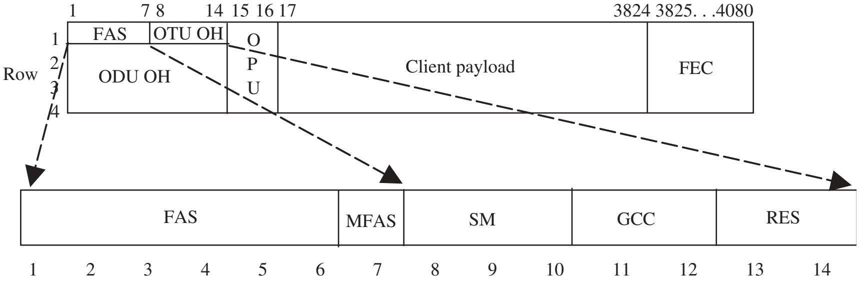


Figure 9.14 The FAS and OTU overheads.

In Figure 9.14, the OTU overhead fields are located in columns 8 to 14 of the first row. These fields provide supervisory functions for section monitoring and condition the signal for transport between *retiming, reshaping, and regeneration (3R)* points in the optical network. The following fields have been defined: *section monitoring (SM)* and *general communication channel (GCC).*

Finally, the forward error correction is carried in columns 3825 to 4080 of all four rows. The Reed-Solomon code RS (255/239) is used.

Client signals

As mentioned earlier on, the following types of traffic can be mapped onto the OPU payload:

- *SONET/SDH*: STS-48, STS-192, and STS-768 data streams are mapped onto an OPU payload using a locally generated clock or a clock derived from the SONET/SDH signal.
- *IP and Ethernet frames*: This is done using the *generic frame procedure (GFP).* (See Section 2.7.)
- *ATM cells*: A constant bit rate ATM cell stream with a capacity identical to the OPU payload is mapped by aligning the ATM cell bytes to the OPU bytes. A cell can straddle two successive OPU payloads. Decoupling of the cell rate and cell delineation are described in Section 3.4.1.
- *Test signals*: User data is used to carry out stress tests, stimulus response tests, and mapping/demapping of client signals.

9.4 CONTROL PLANE ARCHITECTURES

The control plane consists of protocols that are used to support the data plane, which is concerned with the transmission of data. The control plane protocols are concerned with signaling, routing, and network management. Signaling is used to set up, maintain, and tear-down connections. The ATM protocols Q.2931 and PNNI (see Chapter 5) and the label distribution protocols for setting up LSPs (see Chapter 7) are examples of signaling protocols. Routing is an important part of the network operations. It is used to construct and maintain routes that the data has to follow in order to reach a destination. Finally, network management is concerned with controlling a network so as to maximize its efficiency and productivity. ISO's model divides network management into five categories: fault management, accounting management, configuration management, security management and performance management.

There are basically two different control plane architectures. In the first one, the user is isolated from the network via a *user network interface (UNI).* The user is not aware of the network's topology, its control plane and its data plane. The nodes inside the network interact with each other via a *network-node interface (NNI).* A good example of this control plane architecture is the ATM network. A user can only access the ATM network via an ATM UNI, and the ATM switches inside an ATM network interact with each other via an NNI, such as PNNI in the case of a private network (See Chapters 3 and 5).

In the second control plane architecture, the user is not isolated from the network through a UNI, and the nodes inside the network do not interact with each other via a separate NNI. Rather, all users and nodes run the same set of protocols. A good example of this architecture is the IP network.

Both control plane architectures have been used to devise different control planes for wavelength routing networks. The Optical Internetworking Forum (OIF), following the first control plane architecture, has proposed a user-network interface. It is also working on a network-node interface. IETF has proposed three different control plane models for the transmission of IP traffic over an optical network, which are based on the above two control plane architectures.

An optical network provides interconnectivity to client networks (see Figure 9.15). These client networks could be packet-switching networks, such as IP, ATM, and frame relay networks, and circuit-switching networks, such as SONET/SDH.

A large optical network will typically consist of interconnected smaller optical subnetworks, each representing a separate *control domain*. Each of these smaller networks could be a different administrative system. Also, the equipment within a smaller network could all be of the same vendor, with their own administrative and control procedures.

Within the first control plane architecture, the following three interfaces have been defined: *user-network interface (UNI), internal network-node interface (I-NNI)*, and *external network node interface (E-NNI).* (See Figure 9.16.)

As mentioned above, OIF has specified a UNI which provides signaling procedures for clients to automatically create a connection, delete a connection, and query the status connection over an optical wavelength routing network. The UNI is based on the label distribution protocols LDP and RSVP-TE (see Section 9.6).

IETF has defined three different control plane models: the *peer model*, the *overlay model*, and the *augmented model*. In the discussion below and in Figure 9.15, we assume that the client networks are IP networks. The data plane for the networks is shown as

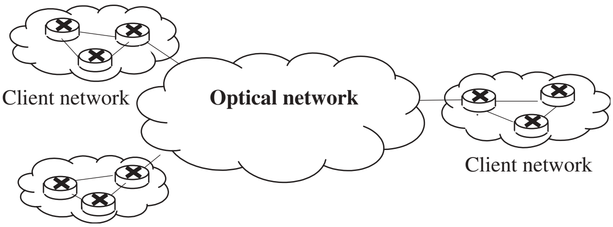


Figure 9.15 Client networks interconnected via an optical networks.

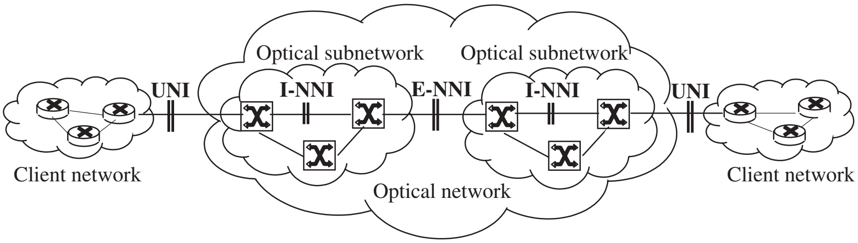


Figure 9.16 The interfaces UNI, I-NNI, and E-NNI.

a mixture of packet-switching and circuit-switching. Packet switching is used within the IP networks; circuit-switching is used within the optical network, where a circuit is a lightpath or subrate channel if traffic grooming is used.

The peer model uses the second control plane architecture described above. That is, the client networks and the optical network are treated as a single network from the point of view of the control plane. The *generalized MPLS (GMPLS)* architecture is used in the control plane. GMPLS is an extension of MPLS (for MPLS, see Chapter 6; for GMPLS, see Section 9.5). The IP and the optical networks run the same IP routing protocol – OSPF with suitable optical extensions. Consequently, all of the optical nodes and IP routers maintain the same topology and link state information. An IP router computes an LSP end-to-end, which is then established using the label distribution protocols CR-LDP or RSVP-TE (see Chapter 7), appropriately extended for GMPLS.

In the overlay model, the optical network uses the first control plane architecture described above (see also Figure 9.16). An IP client network is connected to the optical network via an edge IP router which has an optical interface to its ingress optical node, i.e. the optical node to which it is directly attached. Before an edge IP router can transmit over the optical network, it has to request a connection from its ingress optical node. This is done by using a signaling protocol defined over a UNI. A connection over the optical network can be a lightpath (permanent or switched) or a subchannel. The edge router is not aware of the topology of the optical network; nor is it aware of its control and data planes. The control plane of the optical network can be based on GMPLS. However, UNI maintains a strict separation of the client networks and the optical network.

Finally, in the augmented model, the IP client networks and the optical network use separate control planes. However, routing information from one network is passed to the other. For instance, IP addresses from one IP client network can be carried by the optical network to another IP client network to allow reachability. Routing within the IP and optical networks is separate, but both networks use the same routing protocol. The inter-domain IP routing protocol BGP can be adapted for exchanging information between IP and optical domains.

9.5 GENERALIZED MPLS (GMPLS)

The *generalized MPLS (GMPLS)* architecture is an extension of MPLS described in Chapter 6. MPLS was designed originally to introduce label-switching paths into the IP network, and as we saw in Chapter 6, it is also applicable to ATM, frame relay and Ethernet-based networks. The GMPLS architecture was designed with a view to applying label-switching techniques to *time-division multiplexing (TDM)* networks and wavelength routing networks in addition to packet-switching networks.

A TDM network is a network of SONET/SDH links interconnected by *digital cross connect systems (DCS)*; see Section 2.5. A DCS terminates the SONET/SDH signal on each incoming link, converts it into the electrical domain, and then switches the contents of some of the virtual tributaries to different outgoing SONET/SDH frames. It also drops some virtual tributaries, and adds new ones to the outgoing frames. The outgoing frames are then transmitted out over the SONET/SDH output links of the switch. Aggregation of SONET/SDH payloads to a higher SONET/SDH level can also be done at the output links. A circuit-switching connection through such a SONET/SDH network can be set up by allocating one or more slots of a SONET/SDH frame along the links that make up the path (see Section 2.5). GMPLS can be used to configure the SONET/SDH DCSs, so as to set up a circuit-switching connection.

GMPLS can also be used to set up a lightpath in a wavelength routing optical network. In addition, it can be used to configure an OXC so that to switch the entire optical signal of an input fiber to an output fiber.

In GMPLS, IP routers, ATM switches, frame relay switches, Ethernet switches, DCSs and OXCs are all treated as a single IP network from the control point of view. There are no UNIs and NNIs, since GMPLS is a peer-to-peer protocol.

GMPLS is an architecture and its implementation requires a signaling protocol. Both RSVP-TE and CR-LDP have been extended to support GMPLS.

In the rest of this section, we describe the basic features of the GMPLS architecture and the extensions proposed to CR-LDP and RSVP-TE.

9.5.1 Basic Features of GMPLS

A GMPLS-capable LSR can support one or more of the following interfaces:

1. *Packet-switch capable (PSC) interfaces*: These are the different interfaces used to receive and transmit packets, such as IP packets, ATM cells, frame relay frames, and Ethernet frames. Forwarding of these packets is based on: an encapsulated label, the VPI/VCI field of the ATM cell header, or the DLCI field of the frame relay frame.
2. *Time-division multiplex capable (TDM) interfaces*: They forward data based on the data's slot(s) within a frame. This interface is used in a SONET/SDH DCS.
3. *Lambda switch capable (LSC) interfaces*: They forward data from an incoming wavelength to an outgoing wavelength. This interface is used in OXCs.
4. *Fiber-switch capable (FSC) interfaces*: They forward data from one (or more) incoming fibers to one (or more) outgoing fibers. They are used in an OXC that can operate at the level of one (or more) fibers.

These four interfaces are hierarchically ordered (see Figure 9.17). At the top of the hierarchy is the FSC, followed by the LSC, then TDM, and finally PSC. This order of the interfaces is used by GMPLS to support hierarchical LSPs. (Recall from Section 6.2.4 that MPLS also supports hierarchical LSPs.) Consider an LSP that starts and ends at a packet-switching interface. This LSP can go through several types of networks, where it can be nested together with other LSPs into a higher-order LSP. The high-order LSP can start and end at a packet-switching interface, a time-division interface, a lambda switch interface, or a fiber-switch interface. In general, the nesting of LSPs into a high-order LSP is done following the hierarchy of the above four interfaces (see Figure 9.17).

An example of a hierarchical LSP is shown in Figure 9.18. Assume that a number of IP routers are connected to a SONET/SDH network, which in turn is connected to

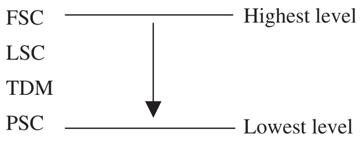


Figure 9.17 The hierarchy of the four types of interfaces.

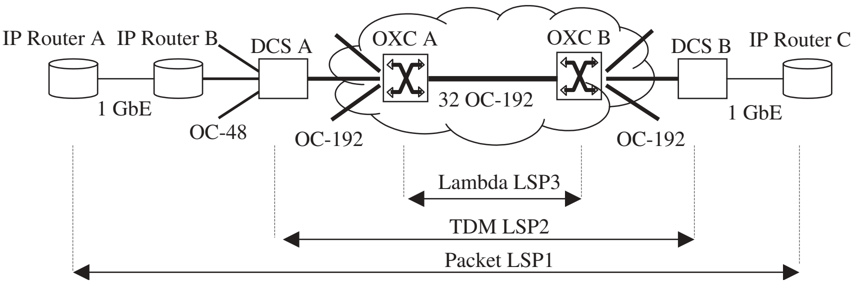


Figure 9.18 An example of hierarchical LSPs.

a backbone wavelength routing network. The LSP starts at IP router A and ends at IP router C. As can be seen, IP router A is connected to IP router B via a 1-GbE link, and IP router B is connected to DCS A via an OC-48/STM-16 SONET/SDH link. DCS A is connected to OXC A via an OC-192/STM-64 SONET/SDH link. OXCs A and B are part of a wavelength routing network, and are connected by a single fiber that has 32 wavelengths – with each wavelength carrying an OC-192/STM-64 SONET/SDH stream. At the other side of the wavelength routing optical network, OXC B is connected to DCS B via an OC-192/STM-64 SONET/SDH link, and DCS B is connected to IP router C via a 1-GbE link.

The interfaces along the path of the LSP from IP router A to IP router C can be easily deduced. The 1-GbE links between IP routers A and B, and DCS B and IP router C have PSC interfaces. The SONET/SDH links between IP router B and DCS A, DCS A and OXC A, and OXC B and DCS B have TDM interfaces. Finally, the link between OXCs A and B has an LSC interface.

As we move towards the wavelength routing optical network, the capacity of the links increase. (This is indicated in Figure 9.18 by using thicker lines). On the other side of the wavelength routing optical network, the link capacities decrease as we move towards the edge, and this is indicated by decreasing the thickness of the lines. The increase in the link capacity as we move closer to the backbone network is normal, since the links carry more traffic than those in the edge of the network.

In Figure 9.18, the LSP between IP routers A and C is labeled as *packet LSP1*. As can be seen, this LSP is nested together with other LSPs in the *TDM LSP2*, which in turn is nested in the *lambda LSP3*. When LSP1 is being established, DCS A will try to allocate bandwidth within its TDM LSP2. If this is not possible, DCS A will establish a new TDM LSP2 to DCS B. The new TDM LSP2 will be nested within the lightpath lambda LSP3, if bandwidth is available. Otherwise, OXC A will establish a new lightpath to OXC B. If LSPs 2 and 3 do not exist at the time when IP router A is attempting to establish LSP1, then the establishment of LSP1 will trigger DCS A to establish TDM LSP2, and OXC A to establish lambda LSP3.

The generalized label request

The generalized label request is used to communicate characteristics required to support the establishment of an LSP. The information required in a generalized label request is shown in Figure 9.19. The following fields have been defined:

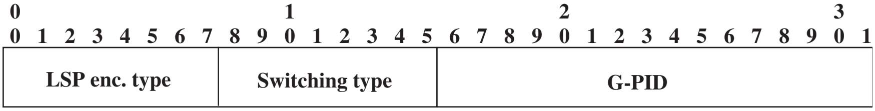


Figure 9.19 The information carried in a generalized label request.

- *LSP encoding type*: This 8-bit field indicates how the data to be transmitted over the LSP will be encoded. The following values have been defined:

Value	Type
1	Packet
2	Ethernet V2/DIX
3	ANSI PDH
4	ETSI PDH
5	SDH ITU-T G.707
6	SONET ANSI T1.105
7	Digital wrapper
8	Lambda (photonic)
9	Fiber
10	Ethernet 802.3
11	Fiber Channel

- *Switching type*: An 8-bit field used to indicate the type of switching that should be performed on a particular link. This field is used on links that advertise more than one type of switching capability.
- *Generalized payload identifier (G-PID)*: A 16-bit filed used to identify the payload carried by an LSP. It is used by the endpoints of the LSP. The following are some of the values specified:

Value	Type	Technology
0	Unknown	All
14	Byte synchronous mapping of E1	SONET/SDH
17	Bit synchronous mapping of DS1/T1	SONET/SDH
28	PoS- No scrambling, 16 bit CRC	SONET/SDH
32	ATM mapping	SONET, SDH
33	Ethernet	Lambda, Fiber
34	SDH	Lambda, Fiber
35	SONET	Lambda, Fiber
36	Digital wrapper	Lambda, Fiber
37	Lambda	Fiber

The generalized label

Since the scope of MPLS was widened into the optical and TDM domains, several new forms of labels are required. The generalized label not only allows for the MPLS-type label that travels in-band with the associated packet, but also allows for labels that identify time slots, wavelengths, or a fiber. These new forms of labels, which are collectively referred to as the *generalized label*, can carry a label that represents:

- Generic MPLS label, frame relay label, ATM label
- A set of time slots within a SONET/SDH frame
- A single wavelength within a waveband or fiber
- A single waveband within a fiber
- A single fiber in a bundle

Since the node using GMPLS knows the type of link used, the generalized label does not contain a type field. The generalized label is not hierarchical. When multiple level of labels are required, each LSP must be established separately. The information carried in the generalized label is shown in Figure 9.20. The interpretation of the label field depends on the type of the link over which the label is used.

The suggested label

GMPLS permits the use of a *suggested label.* This label is used to provide a downstream node with the upstream node's label preference. This permits the upstream node to start configuring its hardware with the proposed label before the label is communicated by the downstream node. This a useful option, if the configuration time is non-trivial. A suggested label can be over-ridden by the downstream node. The suggested label format is the same as the generalized label format.

The label set

The label set is used to limit the label choice of a downstream node to a set of acceptable labels. This limitation applies on a per-hop basis. The receiver of the label set must restrict its choice of labels according to the label set. A label set might be present over many hops, in which case each node generates its own label set possibly based on the incoming label set and the node's hardware capabilities.

A label set is useful in the optical domain in the following four cases:

- *Case 1*: The end equipment is only capable of transmitting or receiving on a small, specific set of wavelengths.
- *Case 2*: There is a sequence of interfaces that cannot support wavelength conversion, and require the same wavelength to be used over a sequence of hops or even the entire path.
- *Case 3*: Limit the number of wavelength conversions along the path.
- *Case 4*: Two ends of a link support different sets of wavelengths.

The information carried in a label set is shown in Figure 9.21. A label set is composed of one or more elements of the label set. Each element is referred to as a *subchannel*, and

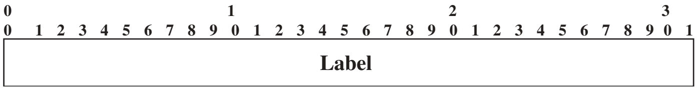


Figure 9.20 The information carried in the generalized label.

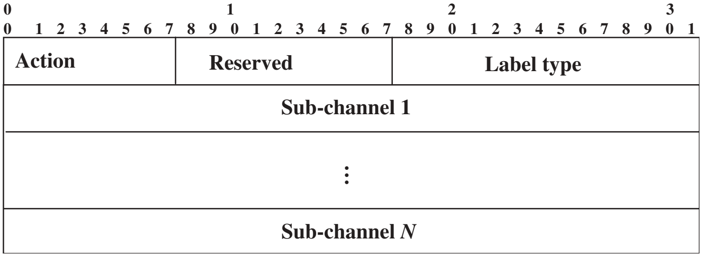


Figure 9.21 The information carried in the label set.

has the same format as the generalized label. The following fields have been defined in addition to the subchannel:

- *Action*: This 8-bit field indicates how the label set is to be interpreted. The following values have been defined:
  - *Inclusive list (value set to 0)*: Indicates that the label set contains one or more subchannel elements that should be included in the label set.
  - *Exclusive list (value set to 1)*: Indicates that the label set contains one or more subchannel elements that should be excluded from the label set.
  - *Inclusive range (value set to 2)*: Indicates that the label set contains a range of labels. The object/TLV contains two subchannel elements: the first one indicates the start of the range, and the second one indicates the end of the range. A value of 0 indicates that there is no bound on the corresponding portion of the range.
  - *Exclusive range (value set to 3)*: Indicates that the label set contains a range of labels that should be excluded from the label set. As above, the object/TLV contains two subchannel elements: the first one indicates the start of the range, and the second one indicates the end of the range. A value of 0 indicates that there is no bound on the corresponding portion of the range.
- *Label type*: A 14-bit field that is used to indicate the type and format of the labels that are carried in the object/TLV.

Bidirectional LSPs

In MPLS, a bidirectional LSP is established by setting up separately two unidirectional LSPs. GMPLS, unlike MPLS, supports the establishment of a bidirectional LSPs. That is, both directions of the LSP are established using a single set of signaling messages.

For a bidirectional LSP, two labels must be allocated over the same hop. Bidirectional setup is indicated by the presence of an upstream label object/TLV in the appropriate signaling message. An upstream label has the same format as the generalized label presented above.

Protection information

Protection information is used to indicate the protection type – dedicated 1 + 1, dedicated 1:1, shared 1:N, or unprotected – desired by the requested LSP on a link. Protection

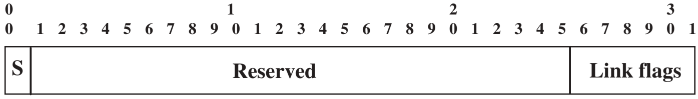


Figure 9.22 Required information in the protection information.

information also indicates whether the LSP is a primary or a secondary LSP. It is assumed that the protection capabilities of each link are known through the routing advertisements.

The required information in the protection information is shown in Figure 9.22. The following fields have been defined:

- *Secondary (S)*: A 1-bit field that is used to indicate that the requested LSP is a secondary LSP.
- *Reserved*: A 25-bit reserved field, set to 0.
- *Link flags*: This field indicates the desired protection type on a link. The following flags have been defined:
  - *Enhanced*: Indicates that a more reliable protection scheme than dedicated $1 + 1$ should be used (i.e., 4-fiber BLSR).
  - *Dedicated 1 + 1*: Indicates that a $1 + 1$ protection scheme should be used.
  - *Dedicated 1:1*: Indicates that a 1:1 protection scheme should be used.
  - *Shared*: It indicates that a shared protection scheme 1:N should be used.
  - *Unprotected*: No protection is required.
  - *Extra traffic*: Indicates that the requested LSP should use links that are protecting other primary LSPs. The requested LSP can be pre-empted if the links carrying the primary LSPs fail.

CR-LDP and RSVP-TE extensions for GMPLS

GMPLS is an architecture, and as in the case of MPLS, it requires a signaling protocol for the reliable distribution of label bindings. Both CR-LDP and RSVP-TE have been extended to support GMPLS. IS-IS and OSPF have also been extended to support GMPLS. In the following section, we present the CR-LDP extensions for GMPLS; the RSVP-TE extensions for GMPLS are presented in Section 9.5.3.

9.5.2 CR-LDP Extensions for GMPLS

New TLVs have been introduced in CR-LDP to support the generalized label operation. Specifically, the generalized label request TLV is shown in Figure 9.23, the generalized

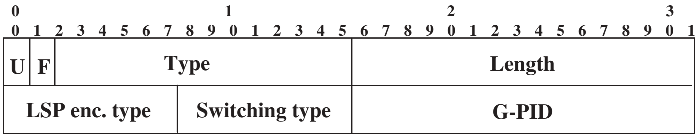


Figure 9.23 The CR-LDP generalized label request TLV.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

U	F	Type	Length
Label			

Figure 9.24 The CR-LDP generalized label TLV.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

U	F	Type	Length
Action		Reserved	Label type
Sub-channel 1			
⋮			
Sub-channel N			

Figure 9.25 The CR-LDP label set TLV.

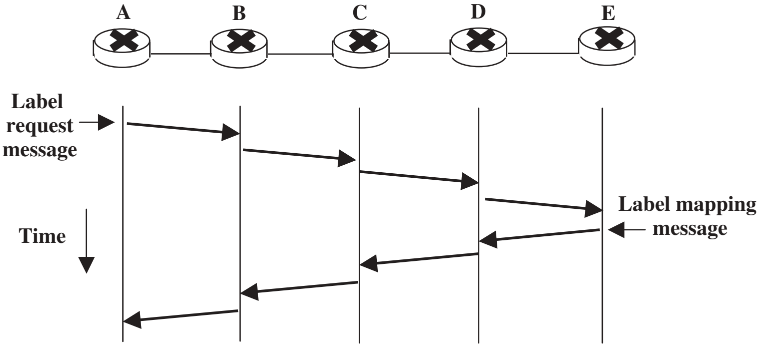


Figure 9.26 The establishment of a CR-LDP.

label TLV is shown in Figure 9.24, the suggested label TLV is the same as the generalized label TLV, and the label set TLV is shown in Figure 9.25.

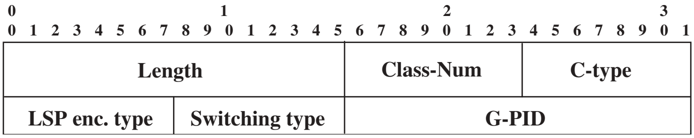
The process of establishing a bidirectional LSP is the same as the one used to establish a unidirectional LSP with some additions. A unidirectional LSP, from LSR A to LSR E, is set up. This is done by using a label request message in the downstream direction (from LSR A to LSR E), and a label mapping message in the upstream direction (from LSR E to LSR A). (See Figure 9.26; see also Section 7.2.1.) Labels for the unidirectional LSP from LSR A to LSR E are set up as the label mapping message travels upstream. This is because an CR-LSP is set up using downstream on demand with ordered control. To support a bidirectional LSP an upstream label is added to the label request message.

A receiving node provides a new upstream label and then forwards the request message to the next downstream node. In this way, as the request message propagates towards the destination LSR E, labels for the LSR E – LSR A path are being set up. The labels for the LSR A – LSR E path are set up as the mapping message propagates towards LSR A.

9.5.3 RSVP-TE Extensions For GMPLS

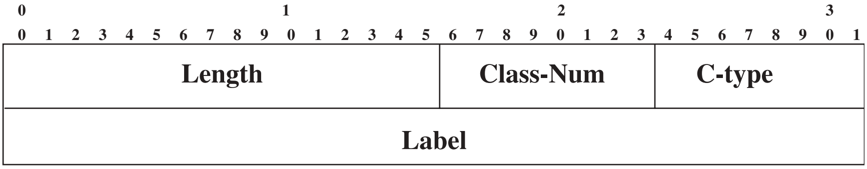
As in the case of CR-LDP, new objects have been introduced in RSVP-TE to support the generalized label operation. The generalized label request object and the suggested label object (which are the same) are shown in Figure 9.27; the generalized label object is shown in Figure 9.28; and the label set object is shown in Figure 9.29.

Bidirectional LSPs are set up using the same process of establishing a unidirectional LSP with some additions. An upstream label is added to the Path message, which permits the allocation of labels along the path from the destination LSR to the source LSR. Labels along the path from the destination LSR to the source LSR are allocated as in the unidirectional LSP using the Resv message.



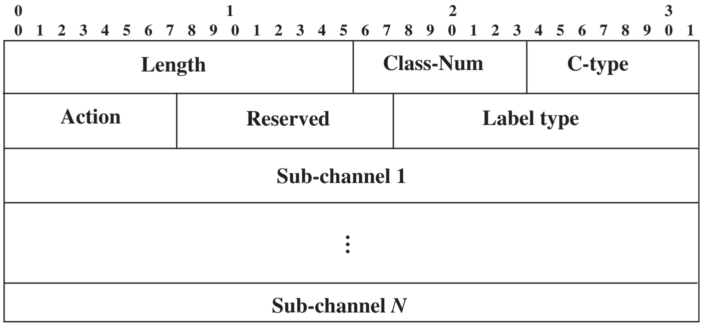
Length		Class-Num	C-type
LSP enc. type	Switching type	G-PID	

Figure 9.27 The RSVP-TE generalized label request object.



Length	Class-Num	C-type
Label		

Figure 9.28 The RSVP-TE generalized label object.



Length	Class-Num	C-type
Action	Reserved	Label type
Sub-channel 1		
⋮		
Sub-channel *N*		

Figure 9.29 The RSVP-TE label set object.

9.6 THE OIF UNI

The OIF UNI specifies signaling procedures for clients to automatically create a connection, delete a connection, and query the status of a connection over a wavelength routing network. The UNI signaling has been implemented by extending the label distribution protocols LDP and RSVP (see Chapter 7). It also uses extensions of the *Link Management Protocol (LMP)*. The client is a packet-switching equipment, such as an IP router and an ATM switch, or a SONET/SDH cross connect which is connected to the optical network. The client side of the OIF UNI is known as the *UNI-C* and the optical network side is known as the *UNI-N*.

A SONET/SDH link is used for the transmission of data between a client and its ingress optical node, known as the *Terminal Network Element (TNE)*. The transmission rate of the SONET/SDH link can be up to STS-768/STM-256.

The UNI signaling messages between the UNI-C and the UNI-N are transported in IP packets, over the *IP Control Channel (IPCC)*. This channel can be *in-fiber* or *out-of-fiber*. The in-fiber IPCC is carried over a channel that is imbedded in the SONET/SDH link used for the transmission of data. This channel is the line and section overhead D bytes (see Section 2.3.2). The out-of-fiber channel is separate from the SONET/SDH data link, and can be an Ethernet link between the UNI-C and UNI-N or an IP network (see Figure 9.30).

As mentioned above in Section 9.4, the UNI isolates the clients from the optical network. In view of this, the topology, resources, and addressing of the optical network is not revealed to the clients. The optical network can use internal addresses for internal routing, provisioning, and network management. These addresses are not revealed to the network clients, and are not within the scope of the OIF standardization process. In addition, the clients can use their own addresses. In view of this two sets of addresses, a *Transport Network Administrative (TNA)* address is used by the UNI to identify the address of a client. The TNA address is a globally uniquely defined address; it is distinct from the native address space of both the clients and the network. To maintain compatibility with network devices that use different addressing types, the TNA can be in the form of IPv4, IPv6, and NSAP (see Section 5.5). The UNI allows a connection between two different TNA type addresses.

The primary services offered to a client by the UNI is the ability to create and delete connections over the optical network on demand. In addition, neighbor and service discovery can be offered optionally. The neighbor discovery procedures allow a TNE and a directly attached client device to determine and identify each other, thus by-passing the necessary manual configuration of the corresponding UNI-C and UNI-N. Service discovery is a process by which a client device obtains information about available services from the optical network.

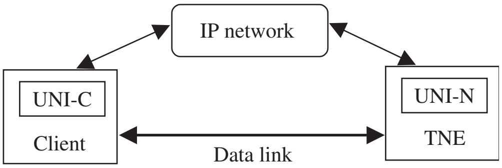


Figure 9.30 The out-of–fiber IPCC.

9.6.1 The UNI Abstract Messages

OIF has defined a number of abstract messages to be used over the UNI. The actual implementation of these messages depends on whether LDP or RSVP is used. These messages are used to create a connection, delete a connection, and query the status of a connection established over the UNI. Recall that a connection is a lightpath or a subrate channel of a lightpath. From the UNI point of view, a connection is a fixed-size bandwidth circuit between an ingress and an egress optical node with a specified frame. At this moment, only SONET/SDH framing is used. A connection can be unidirectional or bidirectional.

The following abstract messages have been defined:

- *Connection create request*: Sent by the source UNI-C to its ingress UNI-N to request the establishment of a connection. It is also sent from the destination egress UNI-N to the destination UNI-C to indicate an incoming connection request.
- *Connection create response*: Used to inform the source UNI-C (i.e. that initiated the connection request) of the establishment of the connection. It is sent from the destination UNI-C to the destination UNI-N, and from the source UNI-N to the source UNI-C, which can then start transmitting data upon receipt of this message.
- *Connection create confirmation*: Used from the source UNI-C to the source UNI-N to acknowledge completion of the connection establishment. It is also used by the destination UNI-N to the destination UNI-C to indicate that the connection has been successfully established.
- *Connection delete request*: Used to initiate the deletion of a connection; it can be sent by either UNI-C. It can also be sent by the network in case of internal failure.
- *Connection delete response*: Used to signal the completion of the deletion of a connection procedure.
- *Connection status enquiry*: This message is used to enquire about the status and attributes of a connection.
- *Connection status response*: Used to return the status of the specified connection and its attributes.
- *Notification*: Used by a UNI-N to either UNI-C to indicate a change of status of the connection.

Figure 9.31 shows the messages involved in the establishment of a connection. As can be seen, the source UNI-C sends a connection establishment request to its ingress UNI-N. This request is propagated to the destination egress UNI-N, which it sends to the destination UNI-C. A connection create response accepting the request is sent by the destination UNI-C to its UNI-N. This message is then propagated back to the source UNI-N, which it sends to the source UNI-C. Connection confirmation follows (see Figure 9.31).

Each message has a number of mandatory and optional attributes. These attributes are organized into the following logical categories:

- *Identification-related attributes*: These include: the source and destination TNA addresses; the client or TNA port number used for the connection; a generalized label; and a local connection ID that are used in all messages to identify which connection they apply to. The local connection ID is assigned by the source UNI-C and, as its name implies, has local significance. That is, it is only valid between the UNI-C and the ingress UNI-N. A similar connection ID is used between the destination UNI-N and UNI-C.

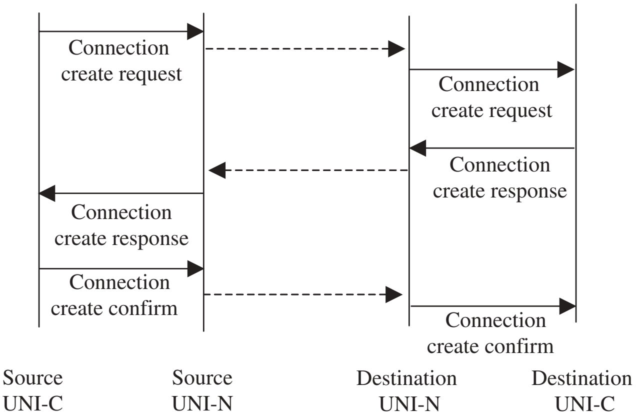


Figure 9.31 Successful connection establishment.

- *Service-related attributes*: These are: the encoding type; SONET/SDH traffic parameters; directionality; a generalized payload identifier; and the service level. The encoding type indicates whether the format is SONET or SDH. The SONET/SDH traffic parameters provide information regarding the signal type and the way in which the data has been concatenated. The directionality attribute indicates whether the connection is unidirectional or bidirectional. The generalized payload identifier indicates the payload carried within the established connection. Finally, the service level attribute indicates a class of service. A carrier can specify a range of different classes of service, such as gold, bronze, and silver. Since the optical network is a circuit-switching network, these classes of service are not related to the familiar QoS parameters in packet-switching networks, such as packet loss and end-to-end delay. Rather, they refer to issues such as the required restoration scheme (i.e., no restoration, $1 + 1$ protection, etc.) and the connection setup and hold priorities.
- *Routing-related attributes*: The only attribute defined is diversity. This attribute can be used when a new connection is being created, to indicate the diversity of the new connection with a list of n existing connections that start at the same UNI-C. The attribute contains n items of the form ⟨diversity type, local connection ID⟩, where the local connection ID indicates one of the n existing connections that start at the same UNI-C. The following diversity types are possible:
  - ○ *Node diverse*: The new connection shall not use any network nodes that are in the path of the connection denoted by the local connection ID.
  - ○ *Link diverse*: The new connection shall not use any network links that are in the path of the connection denoted by the local connection ID.
  - ○ *Shared risk link group (SRLG) diverse*: The new connection shall not use any network links that have the same SRLG as those in the path of the connection denoted by the local connection ID.
  - ○ *Shared path*: The new connection shall use the same network links as those used by the connection denoted by the local connection ID.
- *Policy-related attributes*: The only attribute defined is the contract ID, which is assigned by the service provider and configured in the clients.

- *Miscellaneous-related attributes*: These are the connection status and error codes. The connection status indicates whether the connection is: active, does not exist, unavailable, or pending. The error code is used to describe errors resulting from connection actions, such as: unauthorized sender, unauthorized receiver, service level not available, and diversity not available.

As an example, the following mandatory (M) and optional (O) attributes are used for the connection create request:

- Source TNA (M)
- Source logical port identifier (M)
- Source generalized label (O)
- Destination TNA address (M)
- Destination logical port identifier (O)
- Destination generalized label (O)
- Local connection ID (M)
- Contract ID(O)
- Encoding type (M)
- SONET/SDH traffic parameters (M)
- Directionality (O)
- Generalized payload identifier (O)
- Service level (O)
- Diversity (O)

The UNI signaling has been implemented by extending the label distribution protocols LDP and RSVP. Below, we describe the LDP and RSVP extensions.

9.6.2 LDP Extensions for UNI Signaling

Two guiding principles were used when extending LDP; limit the introduction of new LDP messages, and LDP extensions should be easily implemented as simple additions to the existing LDP implementation without violating the LDP semantics. As a result, only two new messages (a status enquiry message and a status response message) were introduced. The general TLV encodings for LDP and CR-LDP are also used in OIF UNI signaling. Additionally, new TLVs were introduced to carry the attributes defined in the UNI.

LDP session initialization

A single LDP session is established between the UNI-C and the UNI-N, regardless of the number of data links between the client and the TNE. The LDP hello and extended hello are used for neighbor discovery.

Connection establishment

The connection create request is implemented using the label request message (see Section 7.1.4). This message is sent from the source UNI-C to the source UNI-N. At the other end, the label request message is sent from the destination UNI-N to the destination UNI-C. The destination UNI-C responds to the destination UNI-N with a label

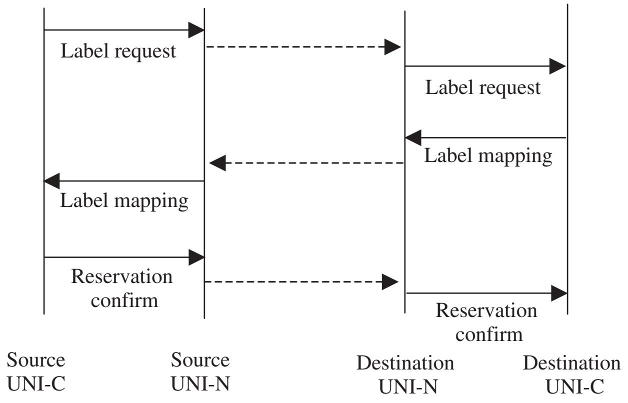


Figure 9.32 Successful connection establishment using LDP.

mapping message, which at the other end, it is sent from the source UNI-N to the source UNI-C. The destination UNI-C can indicate in its label mapping message that a reservation confirmation is required. In this case, a reservation confirmation is sent by the source UNI-C to the source UNI-N, and from the destination UNI-N to the destination UNI-C, as shown in Figure 9.32. The reservation confirm message is implemented using the LDP notification message with the status code set to "reserve confirm."

A connection create request usually specifies a bidirectional connection. Reception of the label request message by the destination UNI-C signifies that the resources to establish the connection with the specified attributes are available in the network. It does not, however, imply that the connection is available for data transport. Specifically, the configuration of the intermediate cross-connects might not have occurred yet. This process starts when the destination UNI-C sends a label mapping message in response to the label request message.

The connection create message might fail for a number of reasons, such as no bandwidth available, *service level agreement (SLA)* violation, and connection rejected by destination UNI-C. In these cases, the failure is indicated to the source UNI-C using the LDP notification message, with the status code set to the reason for the failure.

Connection deletion

To inform its peer to stop using a particular label, an LSR in LDP can employ one of two different messages: the label withdraw message or the label release message. LSR A sends a label withdraw message to a peer LSR B to indicate that B should stop using a specific label that A had previously advertised. Alternatively, LSR A sends a label release message to a peer LSR B to indicate that it no longer needs a specific label that B had previously advertised.

In the UNI LDP extensions, both the label withdraw and the label release messages are used. The choice of which message to use depends on the entity that initiates the deletion. The label withdraw message is used when connection deletion is in the upstream directions, i.e., from the destination UNI-C towards the source UNI-C. The label release message is used in the downstream direction, i.e., from the source UNI-C to the destination UNI-C.

Failure detection and recovery

The LDP keepAlive message is used to detect signaling communication failures between a UNI-C and a UNI-N, unless an alternative mechanism is in place to detect signaling failures more efficiently. During a signaling communication failure, all active connections are maintained whereas all connections which are in the process of being established are cleared.

9.6.3 RSVP Extensions For UNI Signaling

The RSVP protocol definitions apply only to the UNI signaling – that is, between the source UNI-C and UNI-N and the destination UNI-N and UNI-C. The network is assumed to provide coordination of the signaling messages between the source and destination side of the connection.

Most of the UNI abstract messages are directly supported by re-using existing procedures, messages, and objects defined in RSVP-TE and GMPLS extensions of RSVP-TE. Table 9.1 gives the mapping between the OIF UNI abstract messages and the RSVP messages.

Connection establishment

To create a connection, the source UNI-C sends a Path message to its source UNI-N. The Path message includes a GENERALIZED_LABEL_REQUEST object which indicates that a label binding is requested for this connection. The traffic parameters of the connection are encoded in a SONET/SDH SENDER_TSPEC object in the Path message and a SONET/SDH FLOWSPEC object in the corresponding Resv message. Figure 9.33 shows the message flow during a successful connection establishment. It is assumed that the source UNI-N sends a Resv message to the source UNI-C after the segment of the connection within the optical network has been established.

To request a bidirectional connection, a UNI-C must insert an UPSTREAM_LABEL object in the Path message to select the upstream label(s) for the connection.

Connection deletion

A connection in RSVP can be deleted by either using a single PathTear message or an ResvTear and PathTear message combination. Upon receipt of the PathTear message, a

Table 9.1 Mapping between abstract messages and RSVP messages.

Abstract message	RSVP message
Connection create request	Path
Connection create response	Path, PathErr
Connection create confirmation	ResvConf
Connection delete request	Path or Resv
Connection delete response	PathErr, PathTear
Connection status enquiry	implicit
Connection status response	implicit
Notification	PathErr, ResvErr

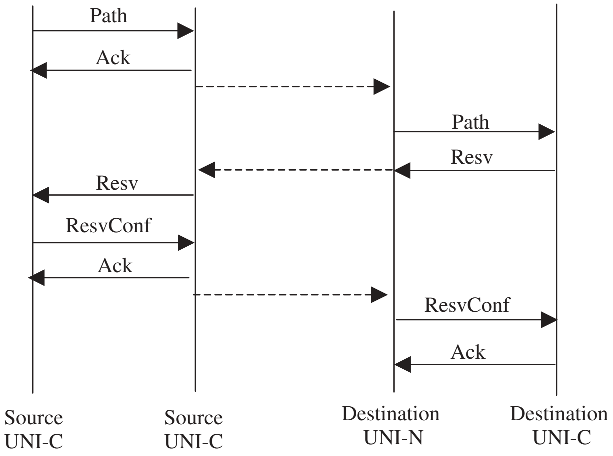


Figure 9.33 Successful connection establishment using RSVP.

node deletes the connection state and forwards the message. In optical networks, however, the deletion of a connection in a node can cause the downstream nodes to think that the connection has failed, which can then lead to network management alarms and perhaps the triggering of a restoration/protection mechanism for the connection. In view of this, a graceful connection deletion mechanism is proposed in the GMPLS extensions for RSVP-TE. Under this procedure, a Path or Resv message with the "deletion in progress" bit in the ADMIN_STATUS object set, is sent along the connection's path to inform all nodes en route of the intended deletion.

PROBLEMS

1. Explain in your own words why traffic cannot be dropped or added to a lightpath at intermediate OXCs.
2. Consider the optical network shown in Figure 9.4. SONET/SDH framing is used on each fiber link with a transmission rate of OC-48/STM-16 (2.488 Gbps). The subrate unit is an OC-3/STM-1 (155 Mbps). Assuming that none of the OXCs are equipped with a SONET/SDH DCS, identify the set of lightpaths that will satisfy the demand, expressed in subrate units, given in the table below.

	OXC 1	OXC 2	OXC 3	OXC 4	OXC 5	OXC 6
OXC 1	–	5	8	10	12	16
OXC 2	–	–	5	1	12	16
OXC 3	–	–	–	5	10	–

3. In the above problem, assume that OXCs 3 and 5 are equipped with a SONET/SDH DCS.
   a. Identify any set of lightpaths that will satisfy the demand given in the table, which requires fewer lightpaths than the set identified in Problem 3.
   b. How many wavelengths are required on each fiber to carry these lightpaths?
4. Consider the unidirectional path sharing ring (OUPSR) described in Section 9.2.2. What is the available capacity of the ring for the transmission of working traffic?

5. Identify all of the lightpaths that are SRLG-disjoint to the lightpath {2, 7, 12} in the optical network shown in Figure 9.7.
6. Explain what is a control plane and what is a data plane?
7. In GMPLS, what is the difference between a suggested label and a label set? Give an example where the label set can be used.
8. What are the differences between GMPLS and the OIF UNI?

APPENDIX: SIMULATION PROJECT: CALCULATION OF CALL BLOCKING PROBABILITIES IN A WAVELENGTH ROUTING NETWORK

The objective of this simulation project is to calculate the blocking probability of dynamically arriving call requests for the establishment of lightpaths in a network of OXCs. Each OXC might have no converters, partial conversion, or full conversion.

Project Description

You will simulate five OXCs, arranged in series and numbered from one to five. OXC i and $i+1$, $i = 1, 2, 3, 4$, are linked by a single fiber that carries W wavelengths. The value of W will be specified as input to the simulation. You will only simulate calls in one direction. That is, calls arrive at each OXC i and their destination is always an OXC j, where $j > i$.

Calls originating at OXC 1 can require a 1-hop, 2-hop, 3-hop or 4-hop path. A 1-hop path is a path from OXC 1 to OXC 2. That is, it originates at OXC 1, uses the link between OXC 1 and 2, and terminates at OXC 2. A 2-hop call uses the link between OXCs 1 and 2, and 2 and 3, and terminates at OXC 3. The same goes for the 3-hop and 4-hop calls. A call originating at OXC 2 can be a 1-hop, 2-hop, or 3-hop call. A 1-hop call uses the link between OXC 2 and 3 and terminates at OXC 3. A 2-hop call uses the link between OXCs 2 and 3 and OXCs 3 and 4, and terminates at OXC 4. Finally, a 3-hop call uses the links between OXCs 2 and 3, OXCs 3 and 4, and OXCs 4 and 5, and terminates at OXC 5. Similarly, for OXC 3, we can have 1-hop and 2-hop calls to OXC 4 and 5, respectively; and for OXC 4, a 1-hop call to OXC 5.

Calls originate at each OXC i in a Poisson fashion with a rate of arrival γ_i, where $i = 1, 2, 3, 4$. Each call chooses a destination OXC j with probability p_{ij}, where $j > i$. Each call has an exponential duration with mean $1/\mu$.

When a call originates at an OXC it is allocated to a free wavelength which is randomly selected from a pool of free wavelengths. If no wavelength is free, the call is lost.

An OXC can have a varied number of converters, from no conversion, to partial conversion, to full conversion (i.e., one converter per wavelength). If the OXC has no converters, and if λ_i is unavailable on the outgoing link, then an incoming call on wavelength λ_i will be blocked (i.e., lost). If the OXC has full conversion, and if λ_i is not available, then an incoming call on wavelength λ_i can be switched out on any randomly selected free wavelength λ_j. If all of the wavelengths on the outgoing link are busy, then the call will be blocked. If an OXC i has partial conversion, then it has c_i converters, where $0 < c_i < W$. In this case, an incoming call on wavelength λ_i is switched out on the same outgoing wavelength λ_i if it is free. If wavelength λ_i is not free, then the call can be switched on any randomly selected free wavelength, provided that there is a free converter. If no converter is free, then the call will be blocked – even if other wavelengths are available. The call will also be blocked if all wavelengths are busy.

Structure of the Simulation Model

The simulation model will be driven by two events. The first event is the arrivals of calls originating at OXCs 1, 2, 3, and 4. The second event is the departures (i.e., call completions). Consequently, you need to maintain an event list with all of the pending arrivals and departures, and a data structure for each OXC where you will keep information such as which wavelengths are used and by which calls.

The event list can be implemented as a linked list. Each element of the linked list will contain the following information:

- Time that the event will occur
- Type of event (i.e., arrival or departure)
- The OXC node that the call will arrive to, if the event is an arrival
- The path the departure is associated with and the wavelength(s) used, if the event is a departure

The linked list will be kept sorted out in an ascending order of the time of occurrence of the event. The next event to occur is always the one at the top of the list. When a new event is generated, it will have to be inserted at the appropriate point, so that the list is always sorted out.

The simulation will use a *master clock (MC)* which keeps the time of the simulation. The main logic of the simulation model involves the following three steps:

Step 1. Choose next event

This is the event at the top of the event list. Advance MC to that event time. If it is an arrival go to Step 2; otherwise, go to Step 3.

Step 2. Next event is an arrival

1. Set MC equal to the time of occurrence of this event.
2. Remove event from the event list.
3. Draw a random number to choose the destination of the call.
4. Check to see if a lightpath can be set up.
5. If no, update statistics and go to Step 7.
6. If yes, the call is established; follow these steps:
   - update data structures in the relevant OXCs;
   - generate an exponential variate which represents the duration of the call;
   - create and insert in the event list a new event associated with the established call which shows when the cal will be completed (departure event).
7. Generate a new inter-arrival time for this OXC; add it to the current value of MC and insert the arrival event into the event list.
8. Go to Step: choose next event.

Step 3. Next event is a departure

1. Set MC equal to the time of occurrence of this event.
2. Remove event from the event list.
3. Release lightpath (i.e. wavelengths used) and update the OXC's data structures.
4. Update statistics.
5. Go to Step: choose next event.

Declare all clocks as real variables.

The inter-arrival times and call duration times are all exponentially distributed. The following procedure can be used to generate a random number X from an exponentially distribution with a mean equal to m.

- Draw a pseudo-random number r from (0, 1).
- $X = -\text{m}\log_e r$

The following procedure can be used to generate a destination OXC. Assume that there are three destinations with probability p_1, p_2, and p_3.

- Draw a pseudo-random number r from (0, 1).
- If $r \leq p_1$, then choose destination 1.
- If $p_1 < r \leq p_1 + p_2$, choose destination 2.
- If $r > p_1 + p_2$, choose destination 3.

Remember to draw a new pseudo-random number every time you need one!

The simulation will run until a predefined number of departures, given as input to the simulation, occurs.

Simulation Inputs

The inputs are: W; the number of converters c_i for each OXC i (where $i = 1, 2, 3, 4$); the arrival rates γ_i (where $i = 1, 2, 3, 4$); the destination probabilities p_{ij} (where $i = 1, 2, 3, 4, 5$; $j = 2, 3, 4, 5$; and $i < j$); the mean call duration $1/\mu$; and the simulation run (that is, the number of simulated departures).

Simulation Outputs

Calculate the total call blocking probability over all paths and also the call blocking probabilities for each path i and j (where $i = 1, 2, 3, 4$; and $j = 2, 3, 4, 5$; and $i < j$). Present your results for the path call blocking probability in a graph where each point on the x axis represents a path. To calculate the output statistics, run your simulation for 100 departures (warmup period), initialize all your statistic counters and then run it for the total number of departures specified as input to the simulation. If your curves are not smooth, increase the simulation run.

Debugging

Run your simulation for several departures, and then print out the content of all of the variables in your simulation, each time an arrival or a departure occurs. Go over the print out by hand and verify that the logic of your program is correct. That is, it advances correctly from event to event, and for each event the data structures are correctly maintained. This is a boring exercise, but it is the only way that you can make sure that your simulation works correctly!

Simulation Experiments

Now that your simulation is ready, use it to carry out the following experiments. Set $W = 5$, all arrival rates $\gamma_i = \gamma, i = 1, 2, 3, 4$, and make the destination probabilities p_{ij}, for each OXC i equaprobable. Run your simulation program for different values of γ for the following configurations of converters:

- No converters: $c_i = 0, i = 1, 2, 3, 4$
- Full conversion: $c_i = W, i = 1, 2, 3, 4$
- Partial conversion 1: $c_i = 2, i = 1, 2, 3, 4$
- Partial conversion 2: $c_i = i + 1, i = 1, 2, 3, 4$

10

Optical Burst Switching

In a wavelength routing network, a connection has to be set up before data can be transmitted. The resources remain allocated to this connection even when there is no traffic transmitted. In view of this, connection utilization can be low when the traffic is bursty. In this chapter, we examine a different optical networking scheme, which is better suited for the transmission of bursty traffic. Because the data is transmitted in bursts, this scheme is known as *optical burst switching (OBS).*

OBS has not as yet been standardized, but it is regarded as a viable solution to the problem of transmitting bursty traffic over an optical network. In an OBS network, the user data is collected at the edge of the network, sorted per destination address, and transmitted across the network in variable size bursts. Prior to transmitting a burst, a control packet is sent into the network in order to set up a bufferless optical connection all of the way to the destination. After a delay, the data burst is transmitted optically without knowing whether the connection has been successfully established all of the way to the destination node. The connection is set up uniquely for the transmission of a single burst, and is torn down after the burst has been transmitted. That is, for each burst that has to be transmitted through the network, a new connection has to be set up.

OBS was preceded by an earlier scheme known as *optical packet switching (OPS).* This scheme was also known as *optical ATM*, since it copied many features of the ATM technology in the optical domain. An optical packet network consists of optical packet switches interconnected with WDM fibers. The switches are either adjacent or connected by lightpaths. Several different designs of optical packet switches have been proposed in the literature. The user data is transmitted in optical packets, which are switched within each optical packet switch entirely in the optical domain. Thus, the user data remains as an optical signal for the entire path from source to destination, and no optical-to-electrical and electrical-to-optical conversions are required. As will be seen, an optical packet switch makes use of optical memories to store optical packets destined to go out of the same output port. Since optical memories do not exist yet, *fiber delay lines (FDLs)* are used in their place. FDLs are useful in prototypes, but they are not suitable for commercial switches. In view of this, optical packet switching has not been commercialized.

In this chapter, we first examine some of the features of optical packet switching and give an example of an optical packet switch. Subsequently, we discuss the main features of OBS, and describe the *Jumpstart* signaling protocol, a proof-of-concept protocol developed to demonstrate the viability of OBS.

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

10.1 OPTICAL PACKET SWITCHING

A WDM optical packet network consists of optical packet switches interconnected by fiber links. An optical packet switch switches incoming optical packets to their desired output ports. As will be seen, the switching of the packets is done in the optical domain. Two switches are interconnected by one or more fibers, each running *W* different wavelengths. Also, it is possible that two switches be interconnected by lightpaths. In this section, we examine some of the features of optical packet switches, and give an example of a switch.

Optical packet switches operate in a slotted manner and switch fixed-size packets. Unlike ATM, the size of a fixed packet is not limited to 53 bytes. In fact, the packet size can be variable, but the time it takes to transmit it is fixed. Variable packet sizes can be transmitted within a time slot of a specified duration by varying the transmission rate.

An optical packet switch consists of input interfaces, a switching fabric, output interfaces, and a control unit. When a packet arrives at an optical packet switch, it is first processed by the input interface. The header and the payload of the packet are separated, as shown in Figure 10.1. The header is sent to the control unit, where it is processed after it has been converted to the electrical domain. The payload remains as an optical signal and is switched to the destination output interface through the switch fabric. At the output interface, it is combined with the header and then transmitted out.

The separation of the header from its payload is necessitated by the fact that it is not technically feasible at this moment to process the header optically. In view of this, the header has to be converted to the electrical domain so that it can be processed by the CPU.

The header can either be transmitted on the same wavelength as the payload, or placed in an electrical subcarrier above the baseband frequencies occupied by the packet payload, and then transmitted both optically in the same time slot. An alternative solution is to transmit the payload and the header in separate wavelengths, but in the same time slot. Also, there should be a gap between the header and the payload, so that there is time to process the header prior to the arrival of the payload.

An optical packet switch (and in general any packet switch) requires buffering. If the switch is non-blocking (which is typically the case), then these buffers have to be placed at the output ports as shown in Figure 10.2. Since each fiber consists of multiple wavelengths, it is possible that multiple packets might arrive at the same time, each carried on the same wavelength i, but originating from differing input fibers. Let us assume that the destination of these packets is the same output fiber. If there are no converters present, then only one packet will be allowed to be transmitted out on the ith wavelength, and the rest will have to be buffered. On the other hand, if full conversion is available, up to *W*

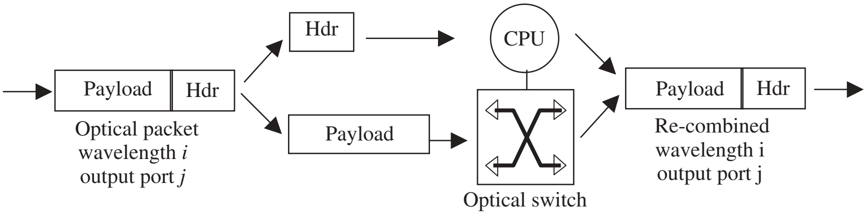


Figure 10.1 The header and payload are separated.

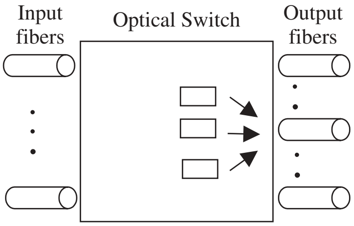


Figure 10.2 Contention in an optical packet switch.

packets (where W is the total number of wavelengths) can be simultaneously transmitted. If more than W packets arrive at the same time, then the additional packets will have to be buffered.

Optical buffers are technologically infeasible. One solution is to convert the optical packets into the electrical domain and buffer them into electrical memories. However, this approach will significantly slow down the operation of the switch. Currently, optical buffers are implemented by *fiber delay lines (FDL)*. An FDL can delay a packet for a specified amount of time, which is related to the length of the delay line. A buffer for N packets with a FIFO discipline can be implemented using N delay lines of different length. Delay line i can delay a packet for i timeslots. Since there are W wavelengths, each delay line can delay up to W packets. Fiber delay lines require lengthy pieces of fiber and so cannot be commercialized. The lack of optical buffering is the Achilles' heel of optical packet switching.

An alternative solution is to use *deflection routing*. When there is a conflict between two packets, one is routed to the correct output port, and the other is routed (i.e., *deflected*) to an alternative output port. The alternate path for routing deflected packets to their destination might be longer. Also, deflection routing imposes an additional load on the links used for the deflected packets, which has to be taken into account when planning the network and the routes that deflected packets will follow. Finally, only a limited number of packets can be deflected at any time.

10.1.1 A Space Switch

In this section, we describe an example of an optical packet switch that uses a space switch fabric. The switch consists of N incoming and N outgoing fiber links, with W wavelengths running on each fiber link (see Figure 10.3). The switch is slotted, and the slot is long enough so that an optical packet can be transmitted and propagated from an input port to an output optical buffer.

The switch fabric consists of three parts: *packet encoder, space switch*, and *packet buffer*. The packet encoder works as follows. For each incoming fiber link, there is an optical demultiplexer which separates the incoming optical signal to W different wavelengths. Each optical signal is fed to a different tunable wavelength converter which converts its wavelength to a wavelength on which it will be transmitted out of the switch.

The space switch fabric can switch a packet to any of the N output optical buffers. The output of a tunable wavelength converter is fed to a decoupler which distributes the same signal to N different output fibers, one per output buffer. The signal on each of

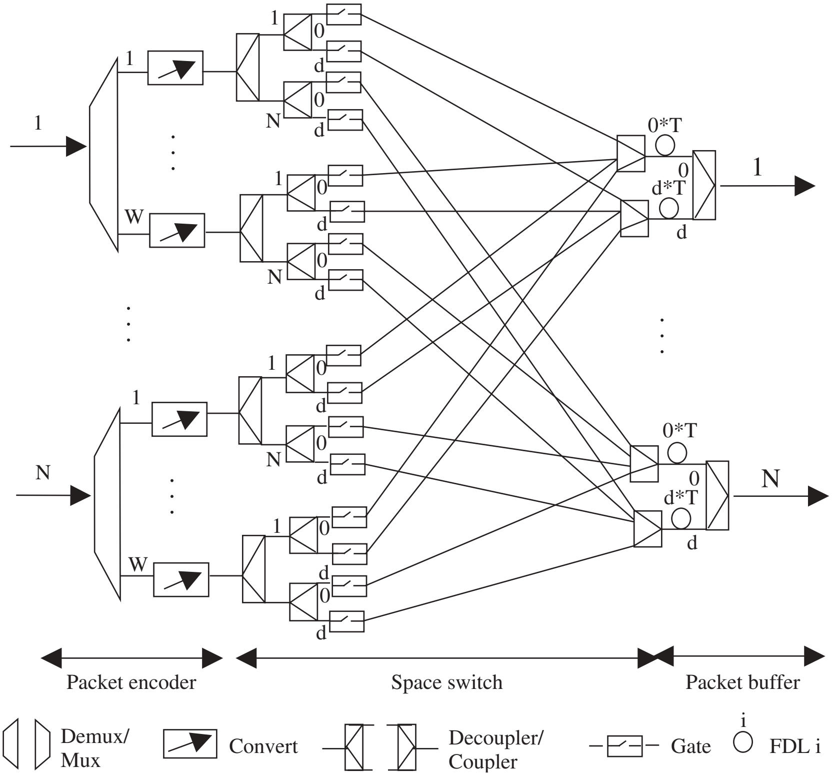


Figure 10.3 An architecture with a space switch fabric.

these output fibers goes through another decoupler which distributes it to $d + 1$ different output fibers, and each output fiber is connected through an optical gate to one of the FDLs of the destination output buffer.

The packet buffer consists of couplers and output buffers, which are implemented in FDLs. Specifically, an output buffer consists of $d + 1$ FDLs, numbered from 0 to d. FDL i delays a packet for a fixed delay equal to i slots. FDL 0 provides zero delay, and a packet arriving at this FDL is simply transmitted immediately out of the output port. Each FDL can delay packets on each of the W wavelengths. For instance, at the beginning of a slot FDL 1 can accept W optical packets – one per wavelength – and delay them for one slot. FDL 2 can accept W optical packets at the beginning of each time slot and delay them for two slots. That is, at slot t, it can accept W packets (again, one per wavelength) and delay them for two slots, in which case, these packets will exit at the beginning of slot $t + 2$. However, at the beginning of slot $t + 1$, it can also accept another batch of W packets. Thus, a maximum of $2W$ packets can be in transit within FDL 2. The same goes for FDL 3 through d.

The information regarding which wavelength a tunable wavelength converter should convert the wavelength of an incoming packet and the decision as to which FDL of the destination output buffer the packet will be switched to is provided by the control unit, which has knowledge of the state of the entire switch.

10.2 OPTICAL BURST SWITCHING (OBS)

OBS was designed to efficiently support the transmission of bursty traffic over an optical network. OBS was based on the *ATM block transfer (ABT)*, an ITU-T standard for burst switching in ATM networks (see Section 4.6.3). OBS is still been developed and it has not as yet been standardized.

An OBS network consists of OBS nodes interconnected with WDM fiber in a mesh topology. An OBS node is an OXC (see Section 8.3.5). It consists of amplifiers, multiplexers/demultiplexers, a switch fabric, and an electronic control unit (see Figure 10.4). The OBS node can switch an optical signal on wavelength λ_i of an input fiber to the same wavelength of an output fiber. If it is equipped with converters, it can switch the optical signal of the incoming wavelength λ_i to another free wavelength of the same output fiber, should wavelength λ_i of the output fiber be in use. (Assume that full conversion applies, and that each converter can convert an optical signal to any other wavelength). Unlike wavelength routing networks, where a connection can remain active for a long time, the switch fabric of an OBS node demands an extremely short configuration time.

The OBS network is accessed by OBS end devices, which are IP routers, ATM switches, or frame relay switches, equipped with an OBS interface. (Devices which produce analogue signals, such as radar, can also be attached to the OBS network.) Each OBS end device is connected to an ingress OBS node.

An OBS end device collects traffic from various electrical networks, such as ATM, IP and frame relay, and it then transmits it to destination OBS end devices optically through the OBS network. The collected data is sorted based on a destination OBS end device address and is assembled into larger size units, called *bursts*. As shown in Figure 10.5, in order to transmit a burst, the end device first transmits a control packet, and after a

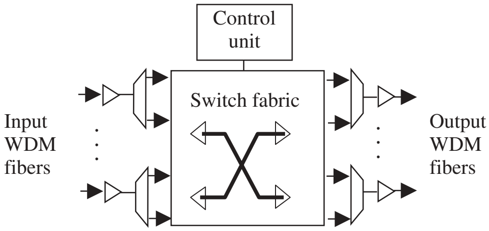


Figure 10.4 Reference OBS node.

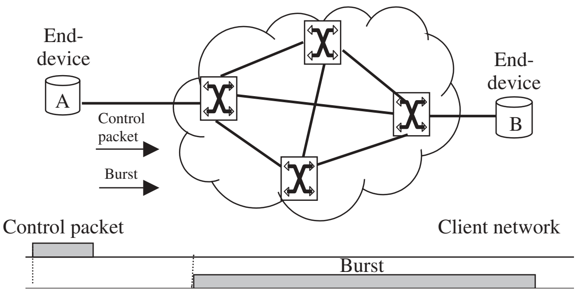


Figure 10.5 A control packet is transmitted prior to transmitting a burst.

delay, known as the *offset*, it transmits its burst. The control packet contains information such as the burst length and the burst destination address. It is basically a request to set up a connection (i.e., a lightpath, end-to-end). After the transmission is completed, the connection is torn down.

As in wavelength routing networks, two adjacent OBS nodes can be linked by one or more optical fibers, each carrying W wavelengths. Therefore, up to W bursts per fiber can be simultaneously transmitted out. An end device might also have the ability to transmit W bursts to its ingress OBS node simultaneously, or it might have only one wavelength available on which to transmit one burst at a time.

Control packets can be transmitted either optically (on a designated signaling wavelength) or electrically (over a packet-switching network, such as an IP or ATM network). In either case, the control packet can only be electronically processed by each OBS node. This means that if it is transmitted in the optical domain, it will have to be converted back to the electrical domain. As can be seen, there exists a separation of control and data, both in time and physical space. This is one of the main features of OBS. It facilitates efficient electronic control while it allows for a great flexibility in the format and transmission rate of the user data since the bursts are transmitted entirely as optical signals which remain transparent throughout the network.

The time it takes for the connection to be set up depends on the end-to-end propagation delay of the control packet, the sum of all of the processing delays of the control packet at all of the intermediate OBS nodes, and configuration delays. (Depending upon the propagation delay between adjacent OBS nodes, the configuration at each node might overlap to some extent.) The time it takes for a burst to reach the destination end device is equal to the end-to-end propagation delay, since it is transmitted as an optical signal that traverses the OBS switches without any processing or buffering delays. In view of this, the transmission of a burst is delayed by an offset so that it always arrives at an OBS node, after the control unit of the node had the chance to process the control packet associated with the burst and configure the OXC. Let t_{proc} be the time it takes to process a control packet at a node, t_{conf} be the time it takes to configure an OXC, and N the total number of OBS nodes along the path of the burst. Then, a good upper bound for the offset can be obtained using the expression: $Nt_{\text{proc}} + t_{\text{conf}}$.

10.2.1 Connection Setup Schemes

To set up a connection, there are two options: *on-the-fly connection setup* and *confirmed connection setup*. In the on-the-fly connection setup scheme, the burst is transmitted after an offset without any knowledge of whether the connection has been successfully established end-to-end. In the confirmed connection setup scheme, a burst is transmitted after the end device receives a confirmation from the OBS network that the connection has been established. This scheme is also known as *Tell and Wait (TAW)*.

An example of the on-the-fly connection setup scheme is shown in Figure 10.6. End-devices A and B are connected via two OBS nodes. The vertical line under each device in Figure 10.6 is a time line and it shows the actions taken by the device. End device A transmits a control packet to its ingress OBS node. The control packet is processed by the control unit of the node and if the connection can be accepted it is forwarded to the next node. This processing time is shown by a vertical shaded box. The control packet is received by the next OBS node, it is processed, and assuming that the node can accept

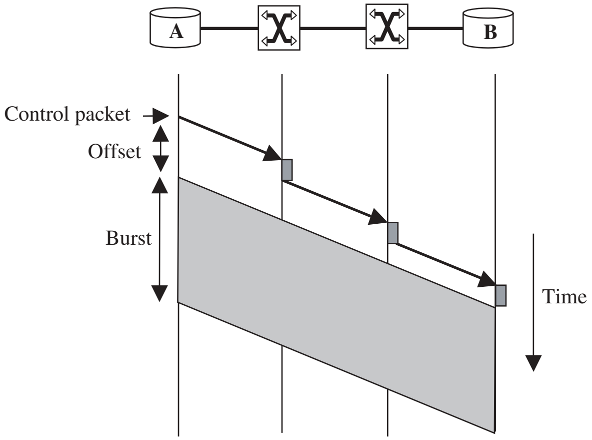


Figure 10.6 The on-the-fly connection setup scheme.

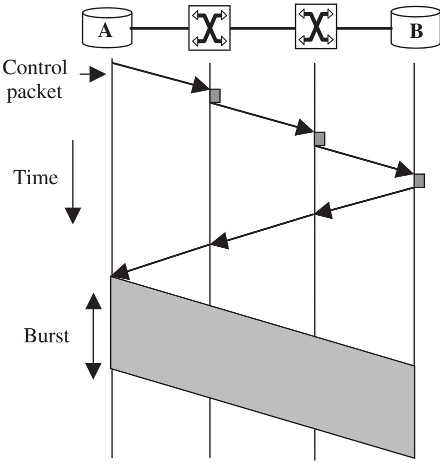


Figure 10.7 The confirmed connection setup scheme.

the connection, it is forwarded to the destination end device node. In the mean time, after an offset delay, end device A starts transmitting the burst which is propagated through the two OBS nodes to the end device B. As we can see in this example, burst transmission begins before the control packet has reached the destination. In this scheme, a burst might be lost if the control packet cannot reserve resources at an OBS node along the burst's path. The OBS architecture is not concerned with retransmissions, as this is left to the upper networking layers. Also, it is important that the offset is calculated correctly. If it is too short, then the burst might arrive at a node prior to the control packet, and it will be lost. If it is too long, then this will reduce the throughput of the end device.

An example of the confirmed connection setup scheme is shown in Figure 10.7. End device A transmits a control packet which is propagated and processed at each node along the path as in the previous scheme. However, the transmission of the burst does not start until A receives a confirmation that the connection has been established. In this case, there is no burst loss and the offset can be seen as being the time it takes to establish the connection and return a confirmation message to the transmitting end device.

In the rest of this section, we will assume the on-the-fly connection setup scheme, unless otherwise stated.

10.2.2 Reservation and Release of Resources in an OXC

In configuring an OXC, two schemes are available: the *immediate setup scheme* and the *delayed setup scheme*. In the immediate setup scheme, the control unit configures the OXC to switch the burst to the output port immediately after it has processed the control packet. In the delayed setup scheme, the control unit calculates the time of arrival t_{arrival} of the burst at the node, and it then waits to configures the OXC at t_{arrival}.

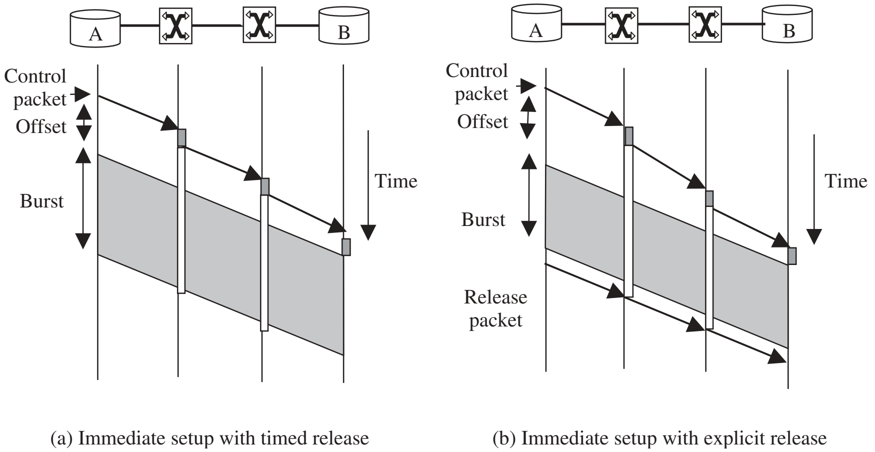
Two different schemes also exist as to when the control unit will instruct the OXC to release the resources allocated for switching the burst. These schemes are the *timed release scheme* and the *explicit release scheme*. In the timed release scheme, the control unit calculates when the burst will completely go through the OXC, and when this time occurs it instructs the OXC to release the allocated resources. This requires knowledge of the burst duration. An alternative scheme is the explicit release scheme, where the transmitting end device sends a message to inform the OBS nodes along the path of the burst that it has finished its transmission. A control unit instructs the OXC to release the connection when it receives this message.

Combining the two setup schemes with the two release schemes gives the following four different mechanisms:

1. Immediate setup with explicit release
2. Immediate setup with timed release
3. Delayed setup with timed release
4. Delayed setup with explicit release

In Figure 10.8, we show the case of immediate setup with timed release, and the case of immediate setup with explicit release. These two schemes have been used in the *Just-In-Time (JIT)* OBS architecture (see Section 10.3). The total amount of time an OXC remains configured for the transmission of the burst is shown by a vertical white box along the time line associated with the switch. The timed release scheme does not require a release message, which has to be processed before the OXC is instructed to release the resources allocated to the burst. In view of this, the resources might be released quicker than in the explicit release scheme. The timed release scheme is more complicated to implement than the explicit release scheme. However, in the explicit release scheme, the duration of burst resource allocation exceeds the duration of the actual burst transmission.

The delayed setup scheme can be combined with timed release or with explicit release. Figure 10.9 shows an example of the delayed setup scheme with timed release, also known as the *Just-Enough-Time (JET)* scheme. The total amount of time an OXC remains configured for the transmission of the burst is shown by a vertical white box along the time line associated with the switch. The timed setup with timed release is the most efficient of all of the four combinations.



(a) Immediate setup with timed release

(b) Immediate setup with explicit release

Figure 10.8 Immediate setup with timed or explicit release.

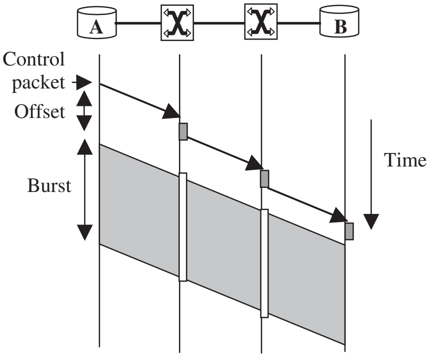


Figure 10.9 Delayed setup with timed release.

10.2.3 Scheduling of Bursts at an OBS Node

Depending on the scheme used to reserve and release OXC resources, there may or may not be a need for a scheduling algorithm. Let us consider first the case of immediate setup with explicit release. Assume that an arriving control packet requests a connection setup on a specific wavelength i of output fiber j. Assuming that this wavelength is free, the control unit immediately configures the OXC, and then forwards the control packet to the next OBS node. Wavelength i on output fiber j remains reserved until the control unit receives a release message. During that time, the control unit cannot schedule any other bursts on this wavelength. Only after it has received the release message it can accept a new request for the same wavelength. This simple scheme is fairly easy to implement, which means that the control unit can decide very quickly whether it can schedule a burst on a wavelength of a given output fiber. Fast processing time of the control packets is critical to the network's throughput. Therefore, the time to process a control packet should be such that the control unit is not the bottleneck in the OBS network. For instance, if it takes 1 msec to process a control packet, then it can process 1000 control packets per

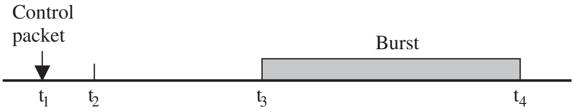


Figure 10.10 The delayed setup scheme.

second, which might be a lot less than the number of bursts that the OXC can switch. That is, the control unit imposes an upper bound on the number of control packets it can process which in turn limits the throughput of the OBS node. Obviously, the control unit has to be able to process more control packets than the number of bursts that the OXC can switch.

Consider the case of immediate setup with timed release. In this case, the control unit knows when a wavelength of an output fiber will become available. Consequently, it can accept a new burst as long as the offset of the burst is such that it will arrive after the wavelength becomes free. For instance, if the wavelength becomes free at time t_1, and the burst is due to arrive at time t_2, where $t_2 > t_1$, then it will accept the new burst as long as there is enough time to configure the OXC between t_1 and t_2.

Now consider the case of delayed setup. In this case, the OXC is not configured until just before the burst arrives. Let us assume that the control packet arrives at time t_1 (see Figure 10.10). At time t_2, the control unit has finished processing the control packet and the burst is not due to arrive until time t_3. Depending upon the length of the gap $t_3 - t_2$, it might be possible to squeeze one or more bursts in-between t_3 and t_4. This technique is known as *void filling*. As mentioned above, the offset is a function of the number of OBS nodes along the path. If there are many nodes, then the time between t_3 and t_4 might be large, and consequently we might be able to do void filling when the node is close to the end device. However, this time gets shorter as we move away from the transmitting end device, which implies that void filling might not be feasible (see Figure 10.9). Finally, because of the timed release scheme, a new burst can be always accepted as long as its offset is such that it will arrive after the wavelength becomes free.

So far, we have assumed that there are no converters. Therefore, an incoming burst on wavelength i has to switched out on the same wavelength. Assuming full or partial conversion, then the scheduling of burst gets more complicated, since now it is possible to schedule the burst on another free wavelengths. Let us assume full conversion. That is, there are as many converters as the number of wavelengths. In this case, a burst can be always switched out as long as there is a free converter. (This assumes that a converter can convert the signal on an incoming wavelength to any other wavelength. This might not always be the case. That is, a converter might only be able to convert an optical signal on a wavelength to other wavelengths that are within certain nanometers.) In the immediate setup with timed or explicit release, scheduling a burst whose wavelength is not available simply involves choosing one of the free wavelengths. In the case of the delayed setup with timed or explicit release, the scheduling algorithm becomes complex and CPU intensive.

10.2.4 Lost Bursts

Burst loss is an important performance measure of an OBS network, and it occurs when the on-the-fly connection setup is used. When a control unit receives a control packet, it extracts the burst's destination from the packet and it then makes a decision as to whether its OXC can switch the burst or not. If it cannot switch the burst, it will not forward the control packet to the next OBS node, and it will send a negative acknowledgment back towards the transmitting end device in order to release any resources already allocated. However, the end device might have already transmitted the burst by the time it receives the negative acknowledgement, which means that the burst will get lost upon arrival at the node.

A small number of FDLs can be used for reducing the burst loss rate, but, as with optical packet switching, FDLs are not commercializable. An alternative solution is to use converters. In this case, if a burst is coming into an OXC on a wavelength which is in use at the destination output fiber, the burst can be converted to another free wavelength.

Finally, deflection routing can be used to divert a burst that would otherwise be lost. The alternative path, however, might have more hops than the original one and therefore the offset might not be long enough to prevent burst loss. A possible remedy is to delay the deflected burst in an FDL so that its offset from the control packet is sufficiently long. Another solution is to use an offset which is the upper bound of all offsets within the OBS autonomous system. This simple solution avoids the need for FDLs, at the expense of delaying the transmission of some of the bursts more than necessary. Finally, deflected bursts impose an extra load on the alternative links, and this has to be taken into account when planning the capacity of the OBS network.

10.2.5 Burst Assembly

Each end device maintains one queue per destination end device. Packets arriving at the end device are placed accordingly into the various destination queues, from where they are transmitted over the OBS network in bursts. The size of the burst is controlled by a timer. When the timer associated with a destination queue expires, all of the data in the queue are grouped into a single burst, which is then transmitted out according to the OBS scheme. The burst size has to be less than a maximum since large size bursts tend to occupy the network resources for long periods of time thus blocking the transmission of other bursts. Also, the size of the burst has to be greater than a minimum because of the signaling overheads for setting up a connection. Therefore, when the timer expires, a burst is assembled if its size is greater than the required minimum. Also, a burst can be assembled before the timer expires if the data in the queue reaches the maximum size. The duration of the timer, and the maximum and minimum burst sizes can be estimated using modelling techniques, such as simulation.

The end device can also introduce priorities when transmitting bursts. Specifically, each destination queue can be further subdivided into multiple QoS queues. The arriving packets are grouped into these queues, which are served according to a scheduler. In addition, different timers and maximum/minimum burst sizes can be used for different queues.

10.3 THE JUMPSTART PROJECT

The Jumpstart project was carried out by MCNC, a non-profit research organization in the Research Triangle Park, North Carolina, and North Carolina State University, and it was funded by the Advanced Research and Development Activity (ARDA). The objectives of the Jumpstart project were to a) define a signaling architecture for an OBS network, and b) create a prototype network using the optical layer of the ADTnet in the Washington, D.C., area. In this section, we present the basic features of the Jumpstart signaling architecture.

The Jumpstart architecture is based on the immediate setup with timed or explicit release and it uses both on-the-fly and confirmed connection setup methods. A single high-capacity signaling channel is assumed for each WDM fiber. The signaling channel can be one of the W wavelengths of the fiber, or it can be implemented in the electrical domain using a packet-switching network. In the Jumpstart prototype network, it was implemented using an ATM network. The signaling messages associated with the establishment and tearing down of connections for burst transmission were processed in hardware in order to assure fast connection establishment. Other control messages were processed in software. The bursts are transmitted through the intermediate OBS nodes transparently without any electro-optical conversion. In view of this, optical digital signals of different formats and modulations, as well as analog signals can be transmitted. No global synchronization scheme is required between the OBS nodes Finally, both unicast and multicast connections are supported in the Jumpstart signaling architecture.

10.3.1 Signaling Messages

The following are some of the signaling messages that were defined:

- SETUP
- SETUP ACK
- KEEP ALIVE
- RELEASE
- CONNECT
- FAILURE

Figure 10.11 shows how these messages can be used to set up a connection on-the-fly with timed or explicit release. The end device sends a SETUP message to its ingress OBS node requesting a on-the-fly connection. The SETUP message is, in fact, the control packet mentioned in the previous section. The ingress switch processes the SETUP message and if it can accept the setup request it returns a SETUP ACK message to the end device in which it indicates the offset. Also, it forwards the SETUP message to the next hop along the route. If it cannot accept the setup request it sends back a FAILURE message.

The SETUP message is processed by the subsequent OBS nodes as described in the previous section, and the burst is transmitted by the end device after the offset. Figure 10.11 shows both cases of timed release and explicit release. The white box along the time line associated with each OBS node shows the amount of time the OXC remains configured for the burst if the timed burst scheme is used. The shaded box beneath shows the additional delay if the explicit release scheme is used. In this case, the end device sends a RELEASE message, as shown in Figure 10.11 to indicate the end of the burst transmission. An optional CONNECT message can be sent back from the destination end device to indicate that the path has been successfully established.

In the case of explicit release, to guard against lost RELEASE messages, the control unit of each OBS node associates the transmission of a burst with a timer. The control unit assumes that the transmission of a burst has been completed if the timer expires and it has not received a RELEASE message. In view of this, when an end device transmits a

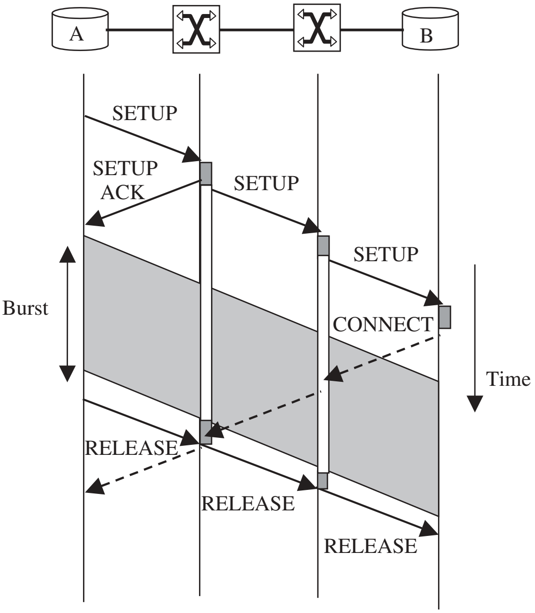


Figure 10.11 On-the-fly connection setup.

very long burst, it must periodically send KEEP ALIVE messages to the network which are used by each control unit to reset the timer.

The on-the-fly connection is set up and torn down each time an end device wants to transmit a burst. The routing protocol determines the route that the SETUP messages follows through the OBS network. Over time, the route between two end devices might vary. In addition to the on-the-fly connection setup, a *persistent connection setup*, which guarantees that a series of burst from an end device to the same destination end device follow the same path through the network, can be used. (In MPLS, it is called a *pinned route*). To set up a persistent connection the following additional messages are used:

- SESSION DECLARATION
- DECLARATION ACK
- SESSION RELEASE

The immediate setup with explicit release scheme is used for persistent connections. The flow of signaling messages is shown in Figure 10.12. A SESSION DECLARATION message is first sent from the transmitting end device to the destination end device to set up the persistent connection. This message is acknowledged by a DECLARATION ACK message. The transmission of bursts can start after the DECLARATION ACK has been received by the transmitting end device. SETUP messages are not required to be transmitted for the burst, since resources have already been allocated within each OBS node along the persistent path for the entire duration of the persistent session. KEEP ALIVE messages might have to be sent, as described above. The session is terminated by sending a SESSION RELEASE message. When an OBS node receives this message, it releases the resources allocated for switching bursts that belong to this session.

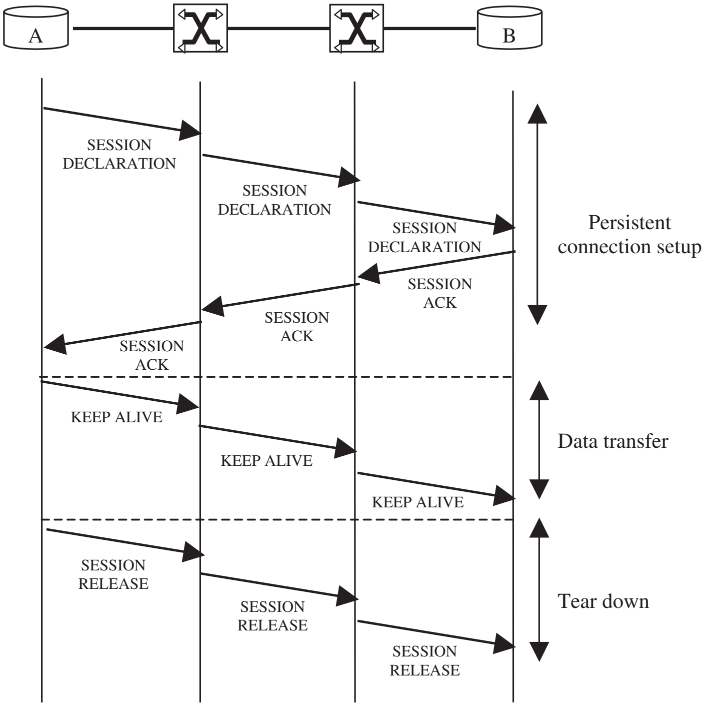


Figure 10.12 Signaling for persistent connection setup.

10.3.2 The Signaling Message Structure

The information carried in a signaling message is organized into *information elements (IE)*, as in ATM's signaling protocol Q.2931. Each IE contains data that are relevant to a particular aspect of the signaling protocol. The IEs are organized into *hardpath IEs* or *softpath IEs*, dependent upon whether they are to be processed in hardware or software.

The format of the signaling message is shown in Figure 10.13. It consists of the fields: common header, hardpath IEs, softpath IEs, and CRC 32.

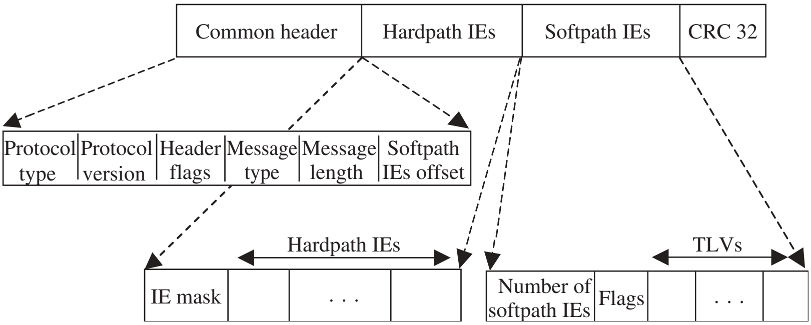


Figure 10.13 The signaling message format.

The common header consists of the subfields: protocol type (1 byte), protocol version (1 byte), header flags (1 byte), message type (1 byte), message length (2 bytes), softpath IEs offset (2 bytes). The signaling messages are not used exclusively for setting up and tearing down connections. They are also used by the routing and the network management protocols. The type of protocol used is indicated in the protocol type field. The message type indicates the message being carried, and the message length gives the length of the entire signaling message. Finally, the softpath IEs offset gives the offset from the end of the common header to the beginning of the softpath IEs field. This offset permits to directly access the softpath IEs field.

The hardpath IEs field contains all of the IEs that are to be processed in hardware. The first subfield, IE mask, is a 32-bit vector that indicates which hardpath IE is present. Each bit is associated with a particular hardpath IE, and up to 32 different hardpath IEs can be accommodated. The IE mask subfield makes it easy for the hardware to parse the hardpath IEs field and determine invalid IE combinations. The hardpath IEs are a multiple of 32 bits; they are fixed and relatively inflexible in format. They are given immediately after the IE mask subfield. Some defined hardpath IEs include: source address, destination address, call reference number, burst descriptor, delay estimator, channel descriptor, QoS descriptor, remaining connection time, session scope, party address, cause, TTL, and bearer signal class.

The softpath IEs field contains all of the IEs that are to be processed by software. Softpath IEs are structured using the *type-length-value (TLV)* format (see Section 7.1.2). Unlike the hardpath IEs field, which permits the parser to see which particular IEs are present, the softpath IEs field simply contains a sequence of IEs. Thus, the software must scan the entire field before it knows which software IEs are present. As shown in Figure 10.13, the softpath IEs field consists of a subfield that gives the number of softpath IEs present, a flags subfield, and the TLVs of the softpath IEs.

Finally, each signaling message is optionally appended with a CRC 32 for integrity verification. The CRC is made optional because the signaling message may be carried over a packet-switching network which provides its own CRC, thus making the CRC 32 field redundant.

10.3.3 Addressing

Jumpstart uses a hierarchical addressing scheme with variable length addresses similar in spirit to the NSAP address format (see Section 5.5). Each address field is represented by an address *LV (length, value)* tuple. The maximum address length is 2048 bits (256 bytes). The hierarchical addressing schemes allows different administrative entities to be responsible for assigning their part of the address. They can decide on the length and the further hierarchical subdivision of the address space.

Figure 10.14 shows a hierarchical network administrative structure that has three levels and eight administrative organizations. For presentation purposes the interconnectivity between the OBS nodes is not shown. The top level domain consists of the domains 0xA and 0xB, with four bits allocated to the top domain addressing level. (The notation 0x indicates that the number following is in hexadecimal). Within domain 0xA, the second level is allocated 8 bits, and is subdivided into domains 0x01, 0x02 and 0x03. These are the lowest level domains, and OBS node addresses in these domains are allotted 8 bits. In domain 0xB, the second level domains are allotted 16 bits, and the OBS node addresses

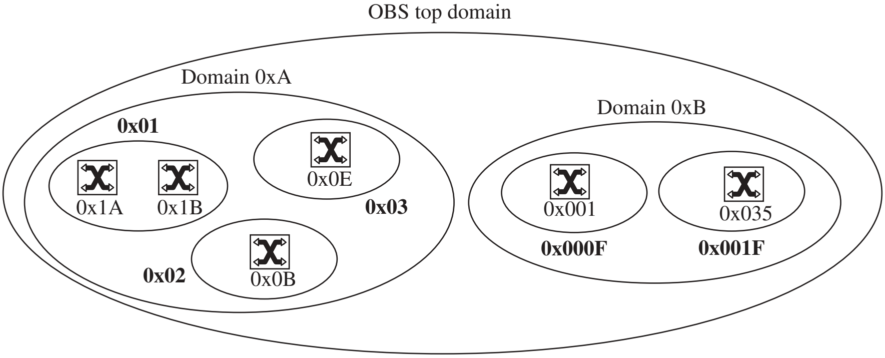


Figure 10.14 Address hierarchy.

within these domains are allotted 12 bits. As an example, the address of the OBS node in domain 0x001F is: 0xB.0x001F.0x035.

10.3.4 The Routing Architecture

In the Jumpstart OBS architecture there is a clear separation between the control plane and the data plane. The data plane is all optical and is responsible for transporting bursts. The control plane is an electronic packet-switching network and is responsible for signaling, routing, and network management.

In the Jumpstart prototype network, an OBS node consists of an OXC (which is a 2D MEMS switch fabric) and a control unit (which is known as the *JITPAC controller [JITPAC])*. The JITPAC controllers communicated with each other via an ATM network. Figure 10.15 shows the control and data planes.

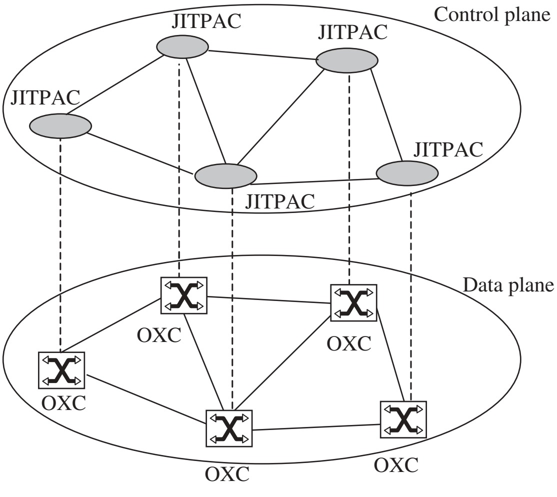


Figure 10.15 The data and control planes in Jumpstart.

Because of the nature of OBS, the signaling messages are associated with bursts and they follow the same path as their bursts. For instance, a SETUP message precedes the transmission of each data burst and is responsible for setting up the path for the burst, while a RELEASE message is responsible for releasing the resources at each OXC after the end of a burst's transmission. Other such signaling messages are the optional CONNECT message, which confirms the establishment of a path, and the FAILURE message, which is used to release network resources when a burst is dropped within the network. By definition, these messages have to go through all of the OBS nodes along the burst's path, either in the forward direction (i.e., the SETUP and RELEASE messages) or the reverse direction (i.e., the CONNECT and FAILURE messages).

Unlike the signaling messages, there is no requirement for other control messages, such as those used to exchange routing information and report network failures to the network management system, to take the same path as data bursts. Below, we will refer to all of the messages except the signaling messages, as control messages.

Jumpstart uses different routing architectures for control messages and data bursts. The routing for signaling messages was not considered, since they use the same routes as the data bursts. Each JITPAC maintains two forwarding tables, one for control messages, hereafter referred to as the *control forwarding table*, and one for data bursts, hereafter referred to as the *burst forwarding table*. This distinction is a logical one, and the two forwarding tables can be implemented as a single data structure. Also, two discrete path computation components have been defined; one for maintaining the control forwarding table, and the other for maintaining the burst forwarding table. Each path computation component uses its own routing protocol for topology discovery and link status updates, as well as its own routing algorithm for computing paths.

The decision to support two different routing architectures, one for the data plane and one for the control plane, was motivated by several observations. As will be seen below, a transparent optical path which carries a data burst between two OBS nodes must satisfy a completely different set of requirements than the electrical or electro-optic path that carries control messages between the JITPACs. Similarly, the routing information needed to find appropriate optical paths is very different than that needed to route control messages. Therefore, implementing two sets of routing protocols and algorithms allows each set to be optimized for the specific requirements of a given plane (data or control). As a result, any modifications or extensions to the data plane routing architecture will not affect the control plane, and vice versa. For instance, the network designer can upgrade the algorithm to compute optical data paths, or modify the link information collected for routing data bursts, without affecting the control plane routing infrastructure. The decoupling of control and data planes also reduces the overall complexity of the implementation, and makes it easier to debug and deploy the individual pieces of the routing architecture. Furthermore, this decoupling allows the use of existing routing protocols and algorithms whenever appropriate, especially within the control plane.

Below, we describe the intra-domain routing architecture for control messages and data bursts.

The routing architecture for control messages

The control plane of the Jumpstart OBS network is implemented on an electrical packet-switching network. The primary routing goal in this plane is the computation of shortest paths between JITPAC controllers to enable the efficient exchange of control messages. To that effect a link-state protocol such as OSPF or IS-IS can be used to establish paths for control messages.

Each JITPAC broadcasts *link-state advertisements* (LSAs) to all other JITPACs in its domain. An LSA is a message that contains information about the status and attributes of the interfaces of a JITPAC. LSAs are flooded reliably to all of the JITPACs within the domain. As a result, each JITPAC obtains complete information regarding the network topology of the JITPACs. Each JITPAC stores this information in its own control routing database, and runs a routing algorithm to determine how to reach other nodes in its domain. The routing algorithm is typically a variant of Dijkstra's shortest path algorithm, and the link costs are based on an administratively selected metric. The results of the routing algorithm are used by a JITPAC controller to populate its control forwarding table which contains the output interface of the JITPAC for each JITPAC destination address.

Following the initial flooding of LSAs, each JITPAC broadcasts an LSA whenever there is a change in the status of its control interfaces. The link-state protocol can also dictate a periodic LSA broadcast, even in the absence of changes. This process ensures that each JITPAC not only establishes shortest paths to all other JITPACs over the control network, but also that these paths are periodically updated to reflect the current state of the network.

Routing architecture for data bursts

In Jumpstart, it was assumed that transmission quality for the optical fiber links in an OBS network is not the same for all fibers. That is, different fibers might have different linear and non-linear impairments (see Section 8.2.2). To improve the quality of the transmitted signals, service circuits can be strategically located at some of the OBS nodes. These service circuits can provide gain compensation, chromatic compensation, and polarization mode dispersion. The required QoS for the transmitted signal, referred to as the *optical QoS (OQoS)*, is indicated in the SETUP message. Determining a path involves both the selection of a set of physical links between the ingress and egress OBS nodes which provide the requested OQoS, and the assignment of a wavelength, or a set of wavelengths if we assume that OBS nodes are capable of wavelength conversion.

Jumpstart employs a centralized architecture for computing paths for data bursts within a network domain. The forwarding component of the architecture uses a burst forwarding table stored locally at each OBS node. The burst forwarding table contains information not only about the output interface for a burst, but possibly also about the output wavelength, the OQoS degradation expected on this output interface, the offset for the burst (needed at the ingress node), as well as other information necessary to forward a burst. The availability of this information ensures that each JITPAC controller can make a forwarding decision locally using the information provided in the SETUP message.

The path computation is the responsibility of the *routing data node (RDN)*, a server attached to one of the OBS nodes. It is responsible for collecting routing information regarding the data plane, computing the burst forwarding tables for each JITPAC controller, and downloading the tables to the JITPAC controllers.

Each JITPAC controller monitors the status of its outgoing optical data interfaces, and summarizes this information into an *optical LSA (OLSA)*. The link information in an OLSA consists of all link attributes that are necessary for computing paths which provide

OQoS guarantees for the optical signal carrying the data bursts, including the status of the interface, the availability of optical resources such as wavelengths and converters, and optical layer impairments that are relevant to routing. Each JITPAC controller transmits its OLSA to the RDN via a reliable connection over the control plane.

Once the RDN has collected the OLSA from each JITPAC controller in the domain, it uses an algorithm to compute data paths between pairs of OXCs in its domain. Subsequently, the burst forwarding table for each JITPAC controller is constructed and downloaded to the controller via a reliable connection over the control plane. Along with a route and a wavelength, the RDN also provides an estimate of the offset to be used for a burst transmission. An ingress OBS node returns the offset value stored in its local forwarding table in the SETUP ACK to an end device in response to a SETUP message.

In both the control and data plane routing architectures, the forwarding of a burst or a control message can be done by using either the next hop information (as in the IP network) or the labels (as in the MPLS scheme).

PROBLEMS

1. Note that optical packet switching and OBS are conceptually similar. Their difference lies in the implementation! Identify which conceptual features are common to these two schemes. Then describe their main implementation differences.

2. An OBS end device can have a single transmitter to transmit bursts to its ingress OBS node. However, it should be equipped with W receivers in order to be able to receive W bursts simultaneously. (Assume that each fiber can carry W wavelengths.) Give an intuitive argument to justify this.

3. Consider the diagram given in Figure 10.6. Assume that $t_{\text{proc}} = 1$ msec, $t_{\text{conf}} = 1$ msec, and a 1-hop propagation delay of 5 msec. The control packet is sent by A at time 0. Calculate the offset and the time at which the control packet will arrive at each OXC, the time at which it will depart from each OXC, and the time at which the burst will arrive at each OXC.

4. Explain why the timed setup with timed release is the most efficient scheme for the reservation and release of resources in an OXC.

5. Show through a numerical example that the offset gets smaller as the control packet moves away from the transmitting end device.

6. A number of OBS end devices are connected to a single OBS node in a star configuration. That is, there is a fiber from each end device to the OBS node, and from the OBS node there is a fiber to each end device. Assume that each fiber carries a single wavelength and that the OBS node has full conversion. Describe a simple scheduling algorithm that guarantees zero burst loss.

7. In Jumpstart, why are there two forwarding tables in a JITPAC?

8. In Jumpstart, the forwarding of a burst or a control message can be done by using either the next hop information (as in the IP network) or labels (as in the MPLS scheme). Explain how the scheme using MPLS labels can be implemented.

11

Access Networks

An access network is a packet-switching network that provides high-speed Internet connectivity to homes. It is anticipated that access networks will also provide additional services, such as voice over IP or ATM, and video on demand. Access networks have different features and requirements than LANs, MANs, and WANs. Currently, there are two different access networks in place; one is provided over the telephone line and the other over the TV cable. New access networks, such as the *ATM passive optical network (APON)*, and Ethernet-based and wireless-based access networks are also beginning to emerge.

Telephone operators provide high-speed access to the Internet over the telephone line in addition to basic telephone services. Video on demand and voice over IP or ATM will also be provided in the future. A family of modems known as *x-type digital subscriber line (xDSL)* has been developed to provide high-speed access to the Internet over the telephone line. Of the xDSL modems, the *asymmetric DSL (ADSL)* is the most popular one.

Cable operators provide access to the Internet over their cable network in addition to the distribution of TV channels. In addition, voice over IP and video on demand services over the cable have been introduced recently. The cable-based access network uses the *data-over-cable service interface specification (DOCSIS)*.

APON is a cost-effective alternative to the telephone-based and cable-based access networks. An APON uses an optical distribution network, which consists of optical fibers and passive splitters. It can be used to provide high-speed Internet connection, voice over IP, voice over ATM, and video on demand services.

In this chapter, we describe the ADSL-based access network, the cable-based access network, and the APON. The ADSL-based access network and the APON have been designed to support ATM and consequently they are connection-oriented networks. The cable-based access network supports the IP network. Although the cable-based access network is not connection-oriented, it has been included in this chapter due to its importance in the access network market.

11.1 THE ADSL-BASED ACCESS NETWORKS

ADSL is one of the access technologies that can be used to convert the telephone line into a high-speed digital link. It is part of a family of technologies called the *x-type digital subscriber line (xDSL)*, where x stands for one of several letters of the alphabet and it indicates a different transmission technique. Examples of the xDSL family technologies

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

are: *asymmetric DSL (ADSL), high data rate DSL (HDSL), symmetric DSL (SDSL), ISDN DSL (IDSL),* and *very high data rate DSL (VDSL).* Some of the xDSL technologies use analog signaling methods to transport analog or digital information over the twisted pair, while others use true digital signaling to transport digital information. A list of specifications for the xDSL family technologies is given in Table 11.1. In access networks, *downstream* means from the network to the user, and *upstream* means from the user to the network. These specifications are likely to change as the technology evolves.

The *very high data rate DSL (VDSL)*, as its name implies, achieves very high data rates over the twisted pair. However, the distance over which such rates can be transported is limited. Currently, it can achieve a downstream data rate of 52 Mbps and an upstream data rate of 6 Mbps over a distance of up to 1000 feet. For the same distance, it can also provide symmetric rates of 26 Mbps downstream and 26 Mbps upstream. The longest distance it can be transported is currently 5000 feet, for which it can achieve 13 Mbps downstream and 1.6 Mbps upstream. VDSL can be used to deliver high quality video together with access to the Internet and regular telephone services. Because of the distance limitation, it is envisioned that it will be used to deliver information from a cabinet in the street which is connected to an APON.

The *asymmetric digital subscriber line (ADSL)* technology utilizes the existing twisted pair from the central office to the home to transport data in addition to the basic telephone services. It was originally designed to provide video on demand services transported over switched DS1 or E1 links. This type of traffic is referred to in the ADSL standard as the *synchronous transfer mode (STM)* traffic. In its current standard (ITU-T G.992.1) full rate ADSL has been defined to carry either ATM or STM traffic or both. ADSL is primarily used for ATM traffic, and there is a limited number of applications for STM traffic.

As its name implies, ADSL provides asymmetrical data rates with the downstream rate being considerably higher than the upstream rate. The data rate depends on the length of the twisted pair, the wire gauge, presence of bridged taps, and cross-couple interference. Ignoring bridged taps, currently ADSL can deliver a full DS1 or E1 signal downstream over a single unloaded 24 gauge twisted pair for a maximum distance of 18,000 feet. Up to 6.1 Mbps is possible for a maximum distance of 12,000 feet, and 8.128 Mbps for a maximum distance of 9000 feet. Upstream data rates currently range between 64 Kbps and 800 Kbps. The ADSL data rates and reach have recently been improved with two new standards: ADSL2 and ADSL2+ (see Section 11.1.5).

The deployment of ADSL over the twisted pair, requires an ADSL transmission unit at either end of the line. The ADSL transmission unit at the customer site is referred to

Table 11.1 xDSL maximum data rates.

xDSL type	Maximum data rate		Usage
	Downstream	Upstream	
ADSL	8.128 Mbps	800 Kbps	Data
HDSL	1.544 Mbps	2.048 Mbps	T1/E1 replacement
SDSL	2.3 Mbps	2.3 Mbps	Data
ISDL	144 Kbps	144 Kbps	Data
VDSL	52 Mbps	6 Mbps	Video/data

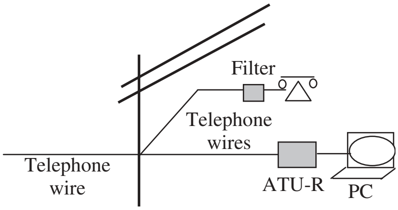


Figure 11.1 Deployment of ADSL at the customer site.

as the *ADSL transceiver unit, remote terminal (ATU-R),* and the ADSL transmission unit at the central office is referred to as the *ADSL transceiver unit, central office (ATU-C).* The signal transmitted over the twisted pair, which contains both ADSL data and voice, is propagated throughout the home telephone wires (see Figure 11.1). The voice signal is filtered out using a high pass filter inside the ATU-R. On the other hand, the ADSL signal can cause a strong audible noise through the telephone set. Therefore, each phone is attached to a telephone plug through a filter, which filters out the ADSL signal and at the same time isolates voice events, such as ring and on/off hook, from the ADSL signal. The ATU-R can be plugged in to any telephone plug.

Consider the ATU-C at the central office. In the downstream direction, the voice signal is added after the ADSL signal leaves the ATU-C (see Figure 11.2). In the upstream direction, the voice signal is extracted from the ADSL signal, before the ATU-C. The ATU-C generates the ADSL signal in the downstream direction, and terminates the ADSL signal in the upstream direction. A number of ATU-Cs are serviced by an *ADSL access multiplexer*, known as *DSLAM*, which provides connectivity to IP and ATM networks.

The DSLAM is an ATM switch. It has an OC-3/STM-1 or higher link to an ATM access backbone network, and has ADSL links serving a number of customer sites. Each ADSL link at the DSLAM is associated with an ATU-C, which is the physical layer associated with the link.

We now proceed to describe how an ATU-C or an ATU-R works. The protocols used to provide IP and ATM services over ADSL are described in Section 11.1.4.

11.1.1 The Discrete Multi-tone (DMT) Technique

The *discrete multi-tone (DMT)* technology is the standardized line coding technique used for ADSL. DMT devices can easily adjust to changing line conditions (such as moisture or interference).

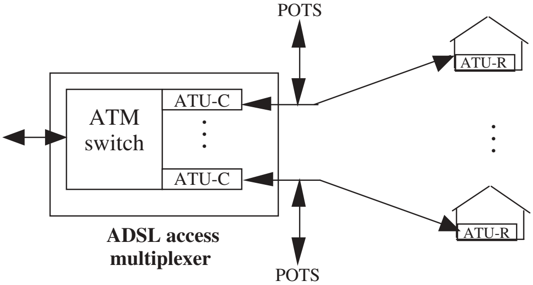


Figure 11.2 ADSL access multiplexer (DSLAM).

In the DMT technique, the entire bandwidth of the twisted pair is divided into a large number of equally spaced subchannels, also known as *tones*. The twisted pair's bandwidth extends to 1.1 MHz; it is divided to 256 subchannels, each occupying 4.3125 KHz. Subchannels 1 through 6 are reserved for the voiceband region, and are used to provide basic telephone services, or *plain old telephone service (POTS)* in networking lingo. ADSL uses the remaining subchannels.

ADSL is bidirectional which means that both the upstream and downstream data is sent over the same twisted pair. In ADSL, bidirectional transmission over the twisted pair can be implemented using either *frequency division multiplexing (FDM)* or *echo cancellation*. In FDM, there are up to 32 upstream subchannels (i.e., from the customer sites to the network) occupying the frequencies immediately above the voiceband region. Also, there are up to 218 downstream subchannels (i.e., from the network to the customer site) occupying the frequencies above the upstream subchannels. An alternative solution is to let the upstream and downstream subchannels use the same frequencies, and separate them using echo cancellation. Not all of the subchannels are used for the transfer of information. Some are used for network management and performance monitoring. All subchannels are monitored constantly for performance and errors and the speed of each subchannel or group of subchannels can actually vary with a granularity of 32 Kbps.

Transmission is achieved by dividing time into fixed-sized intervals. Within each interval, DMT transmits a data frame which consists of a fixed number of bits. The bits in a data frame are divided into groups of bits and each group is transmitted over a different subchannel. The number of bits sent over each subchannel can vary, depending upon the signal and noise level in each subchannel. Using the *quadrature amplitude modulation (QAM)* technique, the bits allocated to each subchannel are converted into a complex number which is used to set the subchannel's amplitude and phase for the interval. The signals are all added up and sent to the twisted pair. This signal resulting from each data frame is known as the *DMT symbol*.

11.1.2 Bearer Channels

A diagram of the ATU-R showing the flow of data in the downstream direction is given in Figure 11.3. The flow of data in the upstream direction in the ATU-C has a similar structure. The data transported by an ATU-R or ATU-C is organized into seven independent logical bearer channels. Of these seven channels, four are unidirectional channels from the network to the customer site. These four channels are referred to as the *simplex bearer*

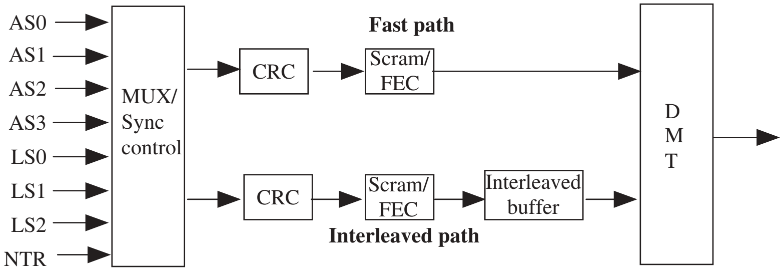


Figure 11.3 The fast path and the interleaved path in ATU-R.

Table 11.2 Data rates for bearer channels.

Bearer channel	Required multiple		Corresponding highest data rate
	Lowest	Highest	
AS0	1	192	6144 Kbps
AS1	1	144	4608 Kbps
AS2	1	96	3072 Kbps
AS3	1	48	1536 Kbps
LS0	1	20	640 Kbps
LS1	1	20	640 Kbps
LS2	1	20	640 Kbps

channels; they are designated as AS0, AS1, AS2, and AS3. The remaining three channels are duplex, and they are referred to as the *duplex bearer channels*. They are bidirectional channels between the network and the customer site, designated as LS0, LS1, LS2. The three duplex bearer channels can also be configured as independent unidirectional simplex channels. All bearer channels can be programmed to transmit at a speed which is an integer multiple of 32 Kbps (see Table 11.2). The maximum total data rate of the ADSL system depends on the characteristics of the twisted pair on which the system is deployed.

STM traffic is mapped in bearer channels AS0 and LS0 in the downstream direction, and in LS0 in the upstream direction. Other bearer channels can also be provisioned. ATM traffic is mapped in the downstream direction in bearer channel AS0 and in LS0 in the upstream direction. Other bearer channels can also be provisioned.

Some applications running at the customer site might require a reference clock. In view of this, in addition to transporting user data, an ATU-C can optionally transport a *network timing reference (NTR)* to an ATU-R.

A bearer channel in an ATU-R or ATU-C can be assigned either to the *fast path* or to the *interleaved path*. The two paths in the ATU-R are shown in Figure 11.3. The fast path provides low delay, whereas the interleaved path provides greater delay but lower error rate. CRC, forward error correction (see Figure 11.4, labeled *FEC*) and scrambling can be applied to each path. Bearer channel AS0 carrying downstream ATM traffic can

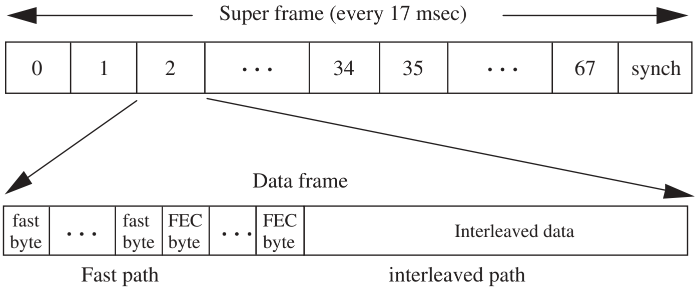


Figure 11.4 The super frame.

be transmitted over either the fast path or the interleaved path. Upstream ATM data are transmitted in LS0 either over the fast path or the interleaved path. The bearer channels carrying STM traffic in either direction are transmitted over either the fast path or the interleaved path. The choice between the fast path and the interleaved path in the downstream direction can be independent of that in the upstream direction.

11.1.3 The ADSL Super Frame

As mentioned above, a data frame consists of a fixed number of bits. Using the DMT technique, it is transmitted every fixed interval. Each data frame combines bits interleaved from the fast and the interleaved paths. The data frames are combined into a super frame consisting of 68 data frames plus a synchronization data frame (see Figure 11.4). Each data frame is transmitted on the twisted pair as a DMT symbol. The rate of transmission of DMT symbols is 4000 symbol/sec. Since a synchronization data frame is transmitted for every 68 data frames, the transmission rate on the twisted pair is actually slightly higher: $(69/68) \times 4000$ symbols/sec. That is, the super frame repeats every 17 msec.

11.1.4 Schemes for Accessing Network Service Providers

Network Service Providers (NSPs) include content providers, ISPs, and corporate networks. Providing access to NSPs is an important service to the ADSL users. This section discusses two schemes that provide connectivity to NSPs: the *L2TP Access Aggregation scheme* and the *PPP Terminated Aggregation scheme*. These schemes are discussed below.

The ADSL Forum's reference architecture (see Figure 11.5) defines the connectivity between ADSL users and NSPs. This reference architecture consists of customer premises, an access network, a regional public network, and NSPs. A customer premises could include a residence (e.g. for home office or recreational purposes) or a small business

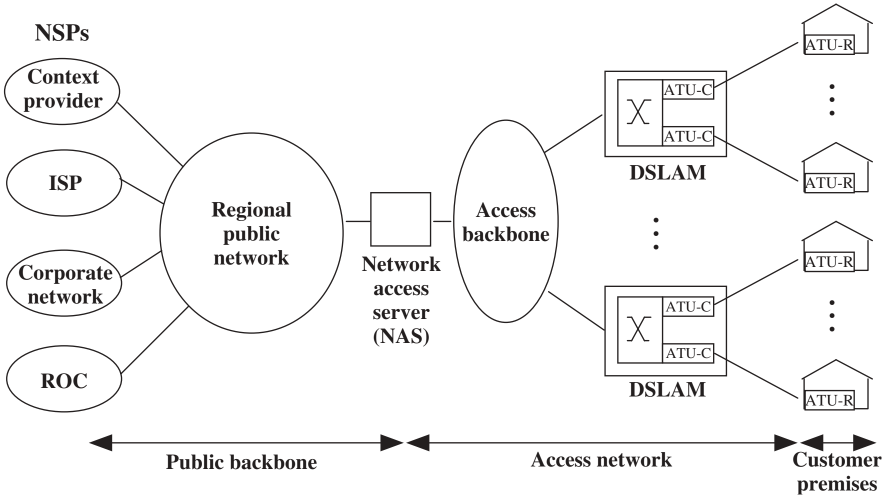


Figure 11.5 The ADSL reference architecture.

office. At a customer premises, there might be one or more computers interconnected by a network. The access network includes the ATU-Rs at the customer site, the DSLAMs that serve these ATU-Rs, and an access backbone network that interconnects all of the DSLAMs. Connectivity to NSPs and to a *Regional Operations Center (ROC)* is provided through a regional public network. Typically, the access network is managed by a telephone operator, who controls it via a ROC. The telephone operator could either be local (known as the *Incumbent Local Exchange Carrier [ILEC]*) or national or newcomer (known as the *Competitive Local Exchange Carrier [CLEC]*).

In most cases, the ATU-Rs do not have the necessary functionality to set up SVCs. Instead, PVCs are used. Providing each ATU-R with a PVC to each NSP requires a large number of PVCs to be set up and managed. A more scaleable approach is to provide a *Network Access Server (NAS)*, as shown in Figure 11.5. The role of the NAS is to terminate all of the PVCs from the ATU-Rs and then aggregate the traffic into a single connection for each NSP.

ADSL users set up sessions to NSPs using the *point-to-point protocol (PPP)*. This protocol was designed to provide a standard method for transporting packets from different protocols over a full-duplex link. PPP provides a number of functions, such as assignment of an IP address by a destination network, domain name auto-configuration, multiplexing of different network layer protocols, authentication, encryption, compression, and billing. PPP frames are transported using a default HDLC-like encapsulation. When PPP runs on top of ATM, PPP frames are mapped into AAL 5 PDUs using either the *VC-multiplexed PPP* scheme or the *LLC encapsulated PPP* scheme. In the former scheme, a PPP frame is directly carried in an AAL 5 PDU. In the latter scheme, a PPP frame is also carried in an AAL 5 PDU after it is further encapsulated with a 2-byte LLC header and a 1-byte network layer protocol identifier.

The L2TP access aggregation scheme

This scheme is based on IETF's *Layer 2 Tunneling Protocol (L2TP)*. The protocol stacks involved in this scheme are shown in Figure 11.6. For simplicity we assume that an ADSL user at a customer site is a single computer, rather than a network of computers interconnected via an ATM network. An ADSL user is connected to the DSLAM over ADSL,

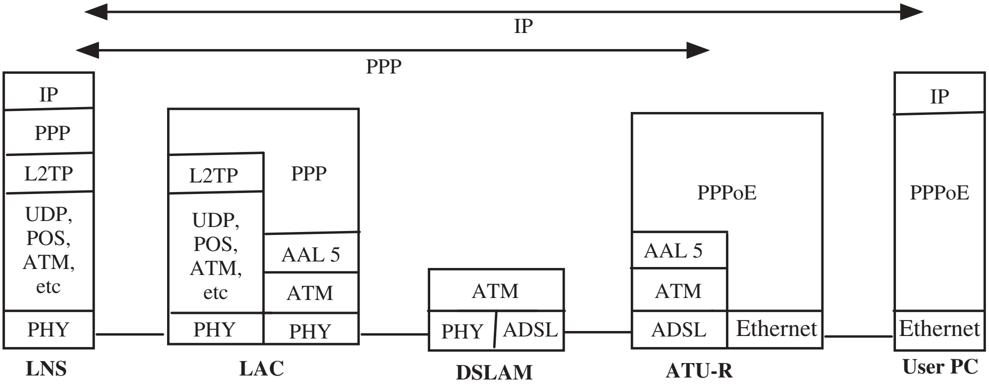


Figure 11.6 The L2TP access aggregation scheme.

and the DSLAM is connected to an NAS, referred to as the *L2TP Access Concentrator (LAC)*, over an ATM network. Finally, the LAC is connected to the *L2TP Network Server (LNS)* of each NSP over a network, such as IP, frame relay, and ATM.

The ADSL user is connected to the LAC with an ATM PVC via the DSLAM. This connection uses AAL 5. The LAC and the LNS of an NSP are connected by an L2TP *tunnel*. An L2TP tunnel is not an actual connection as in a connection-oriented network. Rather, it is a logical connection between the L2TP on the LAC and its peer L2TP on the LNS. A PPP session between the ADSL user and the LNS is established as follows. The ADSL user sends a request to the LAC over AAL 5 to initiate a PPP session to an LNS. This request is forwarded by the LAC to the LNS over an L2TP tunnel. Once the PPP session is established, IP packets can begin to flow between the ADSL user and the LNS.

A tunnel between the LAC and an LNS can multiplex several PPP sessions, each associated with a different ADSL user. Also, there might be several tunnels between the LAC and an LNS. L2TP uses two types of messages: *control messages* and *data messages*. Control messages are used to establish, maintain, and clear tunnels and PPP sessions on demand. Data messages are used to carry PPP frames over a tunnel.

The control and data messages are encapsulated with a common L2TP header. Some of the fields in this header are: type bit (T), length bit (L), priority bit (P), sequence bit (S), length, tunnel ID, session ID, sequence number (Ns), and expected sequence number (Nr). The type bit field indicates whether the L2TP packet is a control or a data message. The length bit field indicates whether the length field is present. If it is present, the length field gives the total length of the L2TP packet in bytes. The priority bit is used for data messages. If it is set to 1, then the L2TP packet is to be given preferential treatment within the L2TP queues. The L2TP packet is associated with a tunnel ID and a PPP session ID, given in the tunnel IP field and the session ID field, respectively. The sequence bit indicates whether sequence numbers are being used. If they are used, then they are carried in the Ns and Nr fields, which are similar to the N(R) and N(S) fields in the HDLC header. That is, the Ns field contains the sequence number of the transmitted L2TP packet, and the Nr field contains the next sequence number the transmitting L2TP expects to receive from its peer L2TP.

A reliable channel between two L2TP peers is provided by L2TP for control messages only. The Ns and Nr sequence numbers are used to detect out-of-sequence packets and missing packets. Lost packets are recovered by retransmission. Data messages can optionally use sequence numbers to reorder packets and detect lost packets. However, no retransmission of data messages takes place. L2TP runs over a network such as IP using UDP, frame relay, and ATM.

The establishment of a session within a tunnel is triggered when the LAC receives a request from an ADSL user to initiate a PPP session to an NLS. Each session within a tunnel corresponds to a single PPP session. Once the session is established, PPP frames can flow between the ADSL user and the LNS. Specifically, PPP frames are transmitted to the LAC over the ATM PVC. The LAC receives the PPP frames from AAL 5, encapsulates each frame with an L2TP header, and transmits it to the LNS as a data message.

The PPP terminated aggregation scheme

This scheme is based on the *remote authentication dial in user service (RADIUS)* protocol. This is a client/server protocol used for authentication, authorization, and accounting.

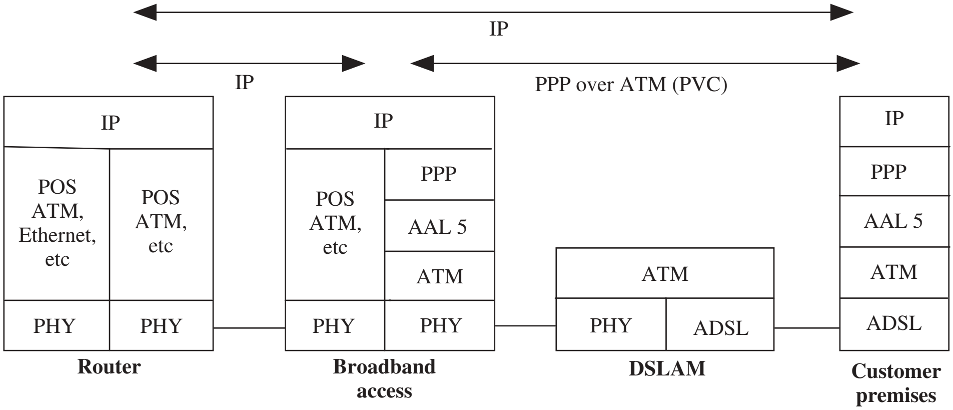


Figure 11.7 The PPP terminated aggregation scheme.

A NAS, referred to as the *Broadband Access Server (BAS)*, acts as a client to a RADIUS server which is managed by an NSP. The BAS is responsible for passing authentication information, such as user login and password, to the RADIUS server. This authentication information is submitted by an ADSL user when it initiates a PPP session. The RADIUS server is responsible for authenticating the ADSL user, and then returning configuration information necessary for the BAS to deliver service to the ADSL user. A BAS also sends the RADIUS server accounting information.

The protocol stacks involved in this scheme are shown in Figure 11.7. As in the previous scheme, we assume that an ADSL user at a customer premises is a single computer, rather than a network of computers interconnected via a home network. The ADSL user is connected to the DSLAM using ADSL. On the side of the access backbone network, the DSLAM is connected to the BAS via an ATM network. Finally, the BAS is connected to NSP routers over a network, such as IP, frame relay, and ATM.

An ADSL user is connected to the BAS with an ATM PVC via the DSLAM. A PPP session initiated by an ADSL user terminates at the BAS, instead of being tunneled to the NSP as in the previous scheme. The BAS sends the user authentication information to the appropriate RADIUS server, and the PPP session is established after the RADIUS server validates the user. The ADSL user can now transmit IP packets, which are forwarded by the BAS to the router of the appropriate NSP.

11.1.5 The ADSL2 and ADSL2+ Standards

ADSL2 adds new features and functionality to ADSL and it was standardized by ITU-T in 2002 (G.992.3 and G.992.4). ADSL2 achieves a maximum downstream data rate of 12 Mbps and a maximum upstream data rate of 1 Mbps. It also extends the reach of ADSL by 600 feet. ADSL2+ was standardized by ITU-T in 2003 (G.992.5) and it doubles the downstream bandwidth, thereby increasing significantly the downstream rate for telephone lines shorter than 5000 feet. For instance, it can achieve 26 Mbps at 1000 feet, and 20 Mbps at 5000 feet. Below, we first describe the main features of ADSL2 and subsequently we describe ADSL2+.

ADSL2

ADSL2 provides a number of features that increase the performance and functionality of ADSL. The following are some of these features:

- *Rate and reach improvement*: ADSL2 increases the data rate and reach of ADSL by doing the following: improving modulation efficiency, reducing framing overhead, achieving a higher coding gain, improving the initialization state machine, and providing enhanced signal processing algorithms.
- *Diagnostics*: ADSL transceivers have been enhanced with extensive diagnostic capabilities that provide troubleshooting tools for installation (during and after) and performance monitoring.
- *Low power/Sleep mode*: ADSL transceivers operate continuously in full power mode, even when they are not in use. The ADSL2 ATU-C is equipped with a low power mode, which it can enter and exit rapidly based on the traffic running over the ADSL connection. In addition, both ATU-C and ATU-R transceivers have a sleep mode that is activated when the ADSL connection is not used for a long time. These two features permit saving a significant amount of electricity, particularly since there are millions of deployed ADSL modems.
- *Rate adaptation*: Telephone wires are bundled together in multi-pair binders, containing 25 or more twisted wire pairs. As a result, electrical signals from one pair might electro-magnetically couple onto adjacent pairs in the binder. This phenomenon, called *crosstalk*, can impede the data rate of ADSL. ADSL2 addresses this problem by seamlessly adapting the data rate. Specifically, the receiver monitors the link's *signal-to-noise ratio (SNR)*, and if it determines that a data rate change is necessary, then it sends a message to the transmitter. The transmitter then sends a Sync Flag signal, which is used as a marker to designate the exact time at which the new data rate is to be used. The Sync Flag signal is detected by the receiver, and the transmitter and receiver both change to the new data rate.
- *Bonding for higher data rates*: Data rates can be significantly increased by bonding multiple phone links together. ADSL2 supports the ATM Forum's inverse ATM multiplexer standard, which was developed for transporting ATM traffic over multiple low-speed links. This standard permits ADSL2 to bind two or more twisted pairs to a single ADSL connection, which results in higher downstream data rates.

In addition to the above features, ADSL2 has a channelization capability, whereby it can split the bandwidth into different channels for different applications. ADSL channelization capability provides support for *Channelized Voice over DSL (CVoDSL)*. Specifically, ADSL2 reserves a number of subchannels *(tones)* in the upstream and downstream spectrum for the transport of PCM 64-Kbps voice data (see Figure 11.8).

With the current ADSL standard, voice can be transmitted over IP *(VoIP)* by encapsulating the IP packets in PPP and subsequently transmitting the PPP frames over AAL5/ATM/ADSL. Voice can also be carried over ATM *(VoATM)* by transmitting the voice packets over AAL2/ATM/ADSL. The protocol stacks for VoATM and VoIP are shown in Figure 11.9. VoIP and VoATM is in contrast with CVoDSL which is transmitted directly over the ADSL physical layer, without having to use any higher-level protocols. Recall that POTS is transmitted directly over the twisted pair using subchannels 1 to 6.

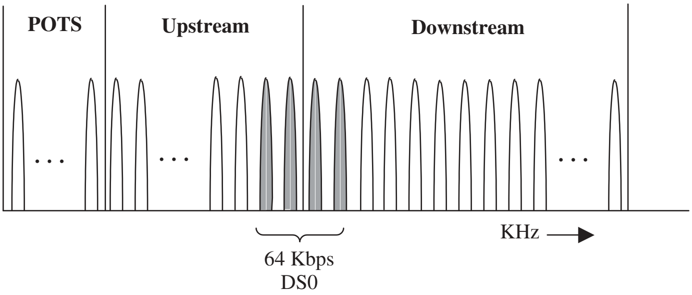


Figure 11.8 Channelized voice over DSL (CVoDSL).

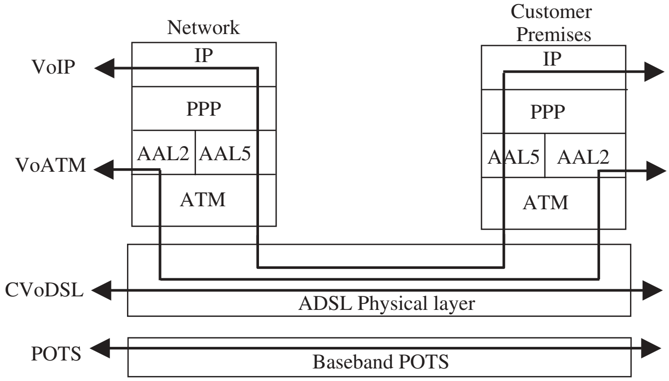


Figure 11.9 CVoDSL, VoATM, VoIP.

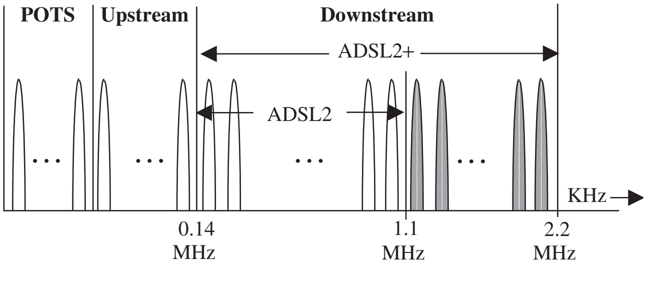


Figure 11.10 The ADSL2+ downstream bandwidth.

ADSL2+

ADSL2+ doubles the downstream bandwidth thereby increasing the downstream data rate on telephone lines shorter than 5000 feet. ADSL and ADSL2 use a bandwidth of 1.1 MHz, whereas ADSL2+ specifies a bandwidth of 2.2 MHz (see Figure 11.10). This results in a significant increase in the downstream speed on short lines.

11.2 THE CABLE-BASED ACCESS NETWORK

A cable network architecture consists of the headend, multiple optical fiber trunks extending from the headend, and coaxial cables. The headend transmits the TV channels which are distributed to the homes over the cable network. Each fiber trunk extending from the headend terminates at an *optical network unit (ONU).* From the ONU, a number of coaxial cables fan out into the neighbourhood, each serving a number of homes (see Figure 11.11). Typically, about 500 homes are served by the same optical fiber. Due to the combination of fiber optics and coaxial cables, this architecture is known as the *hybrid fiber coaxial (HFC)* architecture.

High-speed access to the home is provided over an HFC plant using the *data-over-cable service interface specification (DOCSIS).* This specification was developed by Cable Television Laboratories (CableLabs) for the cable industry in North America, Europe, and other regions. It is also applicable to older all-coax cable TV plants.

DOCSIS permits a transparent bidirectional transfer of IP traffic between the cable system's headend and the homes. This is realized using a *cable modem termination system (CMTS)* at the headend, and a *cable modem (CM)* at each home (see Figure 11.12). The CMTS is a packet switch that is equipped with network interfaces and interfaces to the data-over-cable system. The network interfaces are used to communicate with one or more MAN/WAN networks to which it is connected, and the interfaces to the data-over-cable system are used to transmit and receive data from the CMs over the HFC cable network. The maximum distance between the CMTS and a CM is 100 miles, but it is typically limited to 10 to 15 miles.

The cable network is a shared-medium tree-like network with analog two-way transmission. In the downstream direction, the cable network operates in the range of 50 MHz to 864 MHz. Within this range multiple analog television signals are transmitted in 6-MHz channels, as well as other narrowband and wideband digital signals. In the upstream direction, the cable network operates between 5 MHz and 42 MHz. Within this passband, analog television signals in 6-MHz channels as well as other signals might be present.

In the reference architecture shown in Figure 11.12, we show a single interface to the WAN/MAN network, and a single interface to the data-over-cable access network. Data that is transmitted to the CMs is modulated onto a carrier; it is then multiplexed with all of the television signals and the other signals in the downstream direction. The resulting signal is transmitted out on the optical fiber which terminates at the ONU, and from there it is distributed to all of the homes attached to the coax cables that fan out from the ONU. The data stream transmitted by the CMTS is extracted by each CM, from where it extracts the IP packets destined to it. On the upstream side, each CM transmits IP packets towards the CMTS. These packets are modulated on a

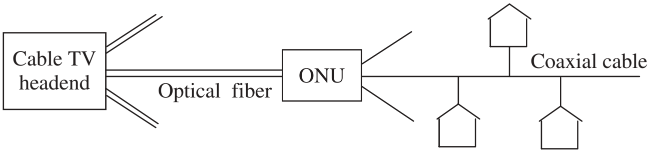


Figure 11.11 The hybrid fiber coaxial (HFC) architecture.

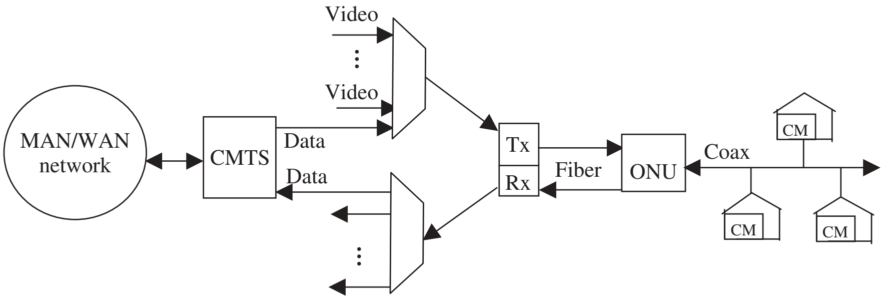


Figure 11.12 The DOCSIS reference architecture.

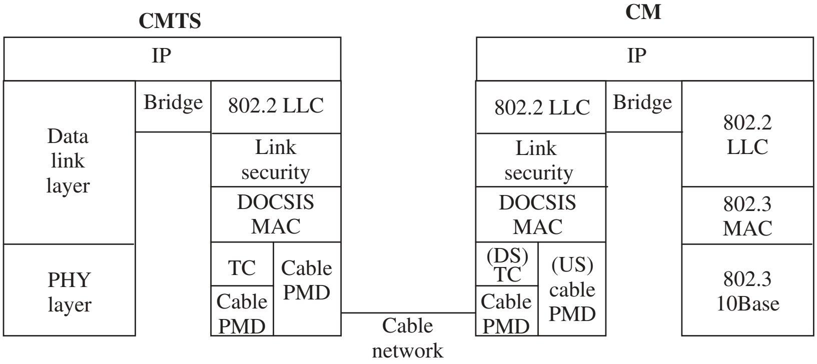


Figure 11.13 The protocol stacks of CMTS and CM.

carrier ranging from 5 MHz to 42 MHz. Multiple carriers can also be used. A specially designed MAC, referred to as the *DOCSIS MAC*, assures that there are no collisions in the upstream direction.

Both CM and CMTS support the IP protocol and other IP-related protocols. The protocol stacks of the CMTS and CM are shown in Figure 11.13. The protocol stack of the CM at the data-over-cable interface, i.e., at the side of the cable network, consists of the physical layer, the DOCSIS MAC protocol, the link security layer, and the IEEE 802.2 LLC. The physical layer consists of the *transmission convergence (TC)* sublayer and the cable *physical medium dependent (PMD)* sublayer. The TC sublayer is only present in the *downstream direction (DS)* – that is, from the CMTS to the CM. It is not present in the *upstream direction (US)* – that is, from the CM to CMTS. The DOCSIS MAC protocol controls the upstream transmission of the CMs, and provides QoS and other features. The CM communicates with the *customer premises equipment (CPE)* via Ethernet. The CM could also be part of a router that forwards packets to CPEs using MPLS label switching or classical IP-based forwarding.

The protocol stack of the CMTS at the data-over-cable interface (i.e., at the side of the cable network) is similar to that of the CM's protocol at the side of the cable network. The stack of the CMTS on the interface to the MAN/WAN network (i.e., at the network side) consists of the appropriate LLC and physical layer necessary to communicate with the MAN/WAN network, to which the CMTS is connected. Packets can be forwarded between the data-over-cable interface and the MAN/WAN network interface via Ethernet. However, it is more likely that the CMTS is also a router that forwards packets to the MAN/WAN network using MPLS label switching or classical IP-based forwarding. In this case, the IP packets are recovered from the LLC frames and sent to the MPLS or IP protocol, from where they are routed out to the network.

11.2.1 The Physical Layer

As mentioned above, the physical layer comprises of the sublayers: *transmission convergence (TC)* and *physical media dependent (PMD)*. The TC sublayer is present only in the downstream direction.

The upstream PMD sublayer uses an FDMA/TDMA mode (referred to as the *TDMA mode*) or an FDMA/TDMA/S-CDMA mode (referred to as the *S-CDMA mode*). Using *frequency division multiple access (FDMA)*, multiple *radio frequency (RF)* channels can coexist in the upstream band. A CM transmits on a single RF channel until it is reconfigured by the CMTS to change. *Time division multiple access (TDMA)* permits the upstream transmission on a channel to be slotted. Access to the slots, referred to in DOCSIS as *mini-slots*, is controlled by the DOCSIS MAC protocol. In the S-CDMA mode, multiple CMs can transmit simultaneously on the same RF channel and during the same TDMA mini-slot. In TDMA mode, a mini-slot is a power-of-two multiple of 6.25 μs; that is, it is equal to $T \times 6.25\,\mu\text{s}$, where $T = 2^n$, and $n = 0, 1, \ldots, 7$. That is, $T = 1, 2, 4, 8, 16, 32, 64, 128$. In S-CDMA mode, a mini-slot is not restricted to be a power-of-two multiple of 6.25 μs increments. Instead, a mini-slot is a unit of capacity that is dependent on the modulation rate, number of spreading codes, and number of spreading intervals configured for the upstream channel. The length of the mini-slot is unrelated to the length of the MAC frames, so that transmitting one MAC frame might require several contiguous mini-slots. The CM upstream transmission speed typically ranges between 500 Kbps and 2.5 Mbps, although it can go up to 10 Mbps.

The downstream PMD uses a 6-MHz channel in the range of 91 MHz to 857 MHz frequencies, and it conforms to the ITU-T standard J.83, Annex B, for low-delay video applications. The downstream transmission speed typically ranges between 1 Mbps and 3 Mbps, but it can reach up to 27 Mbps.

The downstream TC sublayer was defined in order to provide a common receiving hardware at the CM for both video and data. This permits future video services to be offered in addition to the data services. The TC sublayer receives MAC frames from the DOCSIS MAC layer and produces a continuous bit-stream of 188-byte MPEG packets. This continuous stream is passed on to the PMD sublayer which is responsible for transmitting it out. The MPEG packet format is shown in Figure 11.14. The following fields have been defined:

MPEG header	Pointer_field (1 byte)	DOCSIS payload (183 or 184 bytes)

Figure 11.14 The MPEG packet format.

- *MPEG header*: Consists of four bytes and contains various fields, such as: an 8-bit *sync_byte* field; a 1-bit *payload_unit_start_indicator (PUSI)* that is used to indicate the presence of a *pointer_field* in the first byte of the DOCSIS payload; and a 13-bit *packet identifier (PID)* field that carries the DOCSIS data_over_cable well-known PID 0x1FFE.
- *DOCSIS payload*: This field carries DOCSIS MAC frames. It might contain a pointer_field; if it does, then the DOCSIS payload is 183 bytes. Otherwise, it is 184 bytes.

The standard defines a *stuff-byte* pattern that is used within the DOCSIS payload to fill gaps between DOCSIS MAC frames. The value of this pattern is 0xFF, which cannot be used as the first byte of a DOCSIS MAC frame.

A DOCSIS MAC frame can begin anywhere within an MPEG packet and span over several MPEG packets. Also, several frames can potentially exist within the same MPEG packet. The pointer_field is used to correctly recover MAC frames from the MPEG packets. The pointer_field is present in the fifth byte of the MPEG packet, whenever the PUSI in the MPEG header is set to one. It contains the number of bytes in the MPEG packet that immediately follow the pointer_field, and which the CM decoder must skip before looking for the beginning of a DOCSIS MAC frame. The pointer field gives the beginning of the first DOCSIS MAC frame in the MPEG packet or the first stuff-byte preceding a DOCSIS MAC frame.

11.2.2 The DOCSIS MAC Frame Format

The DOCSIS MAC protocol controls the upstream transmission of the CMs, and it provides QoS and other features. For simplicity, we will refer to it as the MAC protocol. In our discussion below, we will assume that there is a single downstream and a single upstream channel between the CMTS and the CMs. (Multiple upstream and downstream channels between the CMTS and the CMs can be also used.)

The MAC frame format consists of the MAC header, followed by an optional data PDU. The MAC header format is shown in Figure 11.15. The following fields have been defined:

- *Frame control (FC)*: An 8-bit field used to identify the type of MAC header. It is broken down to the following subfields:
  - *FC_TYPE*: This 2-bit field specifies one of the following four possible MAC frame formats: MAC header with packet PDU, MAC header with ATM cells, MAC header reserved for future PDU types, and MAC header used for specific control functions.

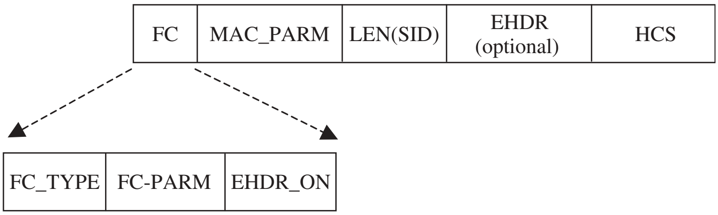


Figure 11.15 The MAC header format.

  - ○ *FC_PARM*: A 5-bit field that contains parameters dependent on the value of the FC_TYPE.
  - ○ *EHDR_ON*: A 1-bit field indicating whether or not an extended header is present.
- *MAC-PARM*: This 1-byte field contains parameters whose use depend on the value of the FC field.
- *LEN (SID)*: A 2-byte field that gives the length of the extended header (if present), plus the number of bytes that follow after the HCS field. In certain cases, this field is used to carry the value of a SID (see Section 11.2.4).
- *Extended header (EHDR)*: This is an optional variable-size extended header.
- *Header check sequence (HCS)*: The integrity of the MAC header is ensured by a CRC. The HCS is a 2-byte field that contains the FCS, which is obtained using the pattern $x^{16} + x^{12} + x^5 + 1$. The HCS covers the entire header (i.e. it starts from the FC field and includes the extended header).

As stated above, the following MAC frame formats are possible: MAC header with packet PDU, MAC header with ATM cells, reserved MAC header for future PDU types, and MAC header used for specific control functions.

The MAC header with packet PDU is used to transport Ethernet packets. In this case, the MAC header is followed by a data PDU with the following fields:

- *Destination address (DA)*: A 48-bit field populated with the destination address.
- *Source address (SA)*: A 48-bit field populated with the source address.
- *Type/len*: A 16-bit Ethernet type length field.
- *User data*: A variable-length field that contains user data of up to 1500 bytes.
- *CRC*: This 32-bit field contains the FCS obtained using the Ethernet-prescribed CRC used to protect the data PDU.

The MAC header with ATM cells is to be defined in the future. The following MAC headers are used for specific control functions:

- *MAC header for timing*: Used in the downstream direction to transport the global timing reference, to which all of the cable modems synchronize. In the upstream direction, it is used as part of the ranging message needed for the cable modem's timing and power adjustment.
- *MAC header for requesting bandwidth*: This header is the basic mechanism that a cable modem uses to request bandwidth. It is only used in the upstream direction. There is no data PDU following this MAC header. Because there is no data PDU, the LEN field is not required and is replaced with a SID. This is the SID allocated to a CM by the CMTS for the upstream service flow.
- *MAC management header*: Used to transport all management messages.
- *MAC header for fragmentation*: In the upstream direction, a large data PDU can be split into a number of smaller pieces, and each piece can be transmitted individually. The pieces are then reassembled into the original data PDU at the CMTS. This MAC header provides the mechanism to do this fragmentation.
- *MAC header for concatenation*: Used to transport several MAC frames (MAC header plus optional data PDU) in a single transmission.

11.2.3 The DOCSIS MAC Protocol Operation

As mentioned above, the upstream channel is divided into mini-slots. The access to the mini-slots by the CMs is controlled by the CMTS. This is done through a MAC management message known as the *MAP management message* (see Figure 11.16). The CMTS issues continuously MAP messages to describe how groups of contiguous mini-slots are to be used. As can be seen in Figure 11.16, the CMTS issues the first MAP message at time t_1. The MAP message propagates through the cable network and it arrives at the CMs at time t_2. (For simplicity, we assumed that all of the CMs are at the same distance from the CMTS). The MAP describes how a group of n_1 contiguous mini-slots are to be used by the CMs and how the CMs should access these mini-slots. The first time slot of this group of mini-slots is expected to arrive at the MCTS at time t_3. Since it takes some delay before the CMs receive a MAP message and process it, the second MAP message is issued by the CMTS at time t_4 and it arrives to the CMs at time t_5. This MAP message describes how the second batch of n_2 contiguous mini-slots that the CMTS expects to start receiving at time t_6 should be used. The third MAP message is issued at time t_7 and it describes how the third group of n_3 mini-slots should be used. The first time slot of this group is expected to arrive at the CMTS at time t_9.

The group of mini-slots mapped in a MAP message is divided into intervals of consecutive mini-slots, referred to as *intervals*. Each of these intervals is designated by the CMTS for different type of use. For instance, one interval can be used by the CMs to transmit bandwidth requests, another interval can be used by the CMs to transmit data PDUs, and a third interval can be used by new CMs to join the access network.

Within an interval of mini-slots, the CMs can start transmitting at the beginning of any mini-slot, with the possibility that their transmissions might collide. A contention resolution algorithm based on the exponential back-off algorithm is used to decide when the CMs that collided can re-transmit again. Alternatively, within an interval each CM transmits over a number of consecutive mini-slots allocated to it by the CMTS without collision. The allocation of mini-slots to CMs is done based on requests for bandwidth that the CMTS receives from the CMs. The CMs send bandwidth requests to the CMTS using the MAC header for requesting bandwidth described above. The CMTS allocates contiguous time slots to the requesting CMs using a scheduling algorithm, which is outside the scope of DOCSIS.

The different intervals and how they can be accessed by the CMs are described in the MAP management message by different *information elements (IE)*. Each information element describes a particular type of interval. The following are some of the supported information elements:

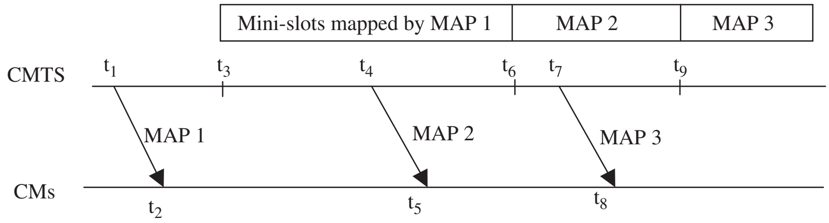


Figure 11.16 An example of MAP messages.

- *The request IE*: Specifies an interval of contiguous mini-slots, during which the CMs can send bandwidth requests to the CMTS. The transmission during these mini-slots is contention-based, so that multiple CMs might potentially transmit at the same time. A contention-resolution algorithm is used to recover from collisions. Alternatively, the CMTS can specify a particular CM to transmit its bandwidth request, in which case there will be no collisions. (Requests for bandwidth can also be transmitted during other intervals, such as the request/data interval and the short or long data interval, described below.)
- *The request/data IE*: Provides an interval of mini-slots, in which either requests for bandwidth or short data PDUs can be transmitted on a contention basis. Since transmissions might collide, the CMTS must acknowledge any data PDUs that it received correctly. (Note that this IE is different from the request IE.)
- *The initial maintenance IE*: The IE provides an interval of mini-slots, during which new CMs can join the network. The transmission during these mini-slots is contention-based.
- *The station maintenance IE*: Defines an interval of mini-slots, during which CMs are expected to perform some aspect of routine maintenance, such as ranging or power adjustment. The transmission during these mini-slots is contention-based. Alternatively, the CMTS can specify a particular CM to perform a network maintenance task.
- *The short and long data grants IEs*: Define an interval of mini-slots, during which CMs can transmit short data PDUs and long data PDUs, respectively. These IEs are issued either in response to bandwidth requests, or to an administrative policy to provide bandwidth to some CMs. A data grant cannot be larger than 255 mini-slots.
- *Data acknowledge IE*: Used by the CMTS to acknowledge that a data PDU has been received. This IE is used only in conjunction with contention transmission. It was not designed to provide a reliable transport between CMs and CMTS.

A CM must request a number of mini-slots in order to transmit an entire frame. This frame might be: a single MAC frame; a MAC frame that has been formed by concatenating multiple MAC frames; or a MAC frame containing a fragment of a long data PDU. The request must be large enough in order to accommodate the MAC frame and the necessary physical layer overheads. A CM cannot request more than the required number of mini-slots necessary to transmit one MAC frame, and it can only have one request outstanding at a time.

In Figure 11.17, we give an example of the upstream transmission scheme between the CMTS and a CM. The following events take place:

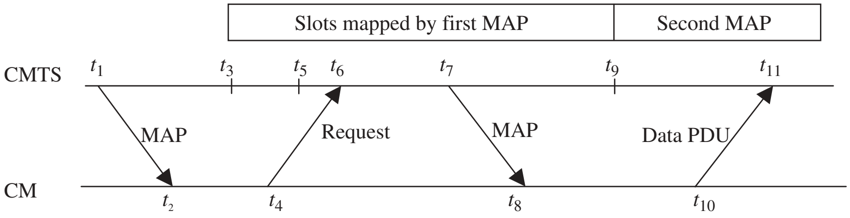


Figure 11.17 An example of the upstream transmission scheme.

- At time t_1, the CMTS transmits a MAP message to a CM that starts at time t$_3$. That is, the CMTS expects to receive the first mini-slot of this mapped group at time t_3. The difference between t_1 and t_3 is needed to allow for all of the delays involved from the time that the CMTS transmits the MAP to the time that a CM can respond. These delays include: the worst-case round-trip propagation delay, queueing delays within the CMTS, processing delays within the CM, and other delays introduced by the PMD layer. Within this MAP, there is a request IE which will start at t_5.
- At t_2, the CM receives the MAP message and scans it for request opportunities. In order to minimize request collisions, it calculates t_6 as a random offset.
- At time t_4, the CM transmits a request for as many mini-slots as are needed to transmit the data PDU. Time t_4 is chosen based on the *ranging offset* so that the request will arrive at the CMTS at t_6. The ranging offset is used to place all of the CMs at the same virtual distance from the CMTS. This is necessitated by the fact that the CMs are at different distances from the CMTS. Because of this, a MAC frame transmitted by a CM might not arrive at the mini-slot at which it is expected. A ranging protocol is used to calculate a ranging offset by which a CM delays its transmissions. The other purpose of the ranging process is to make all CMs transmit at a power level that makes all of their transmissions arrive at the CMTS at the same level of power.
- At t_6, the CMTS receives the request and schedules it for service in the next MAP message.
- At time t_7, the CMTS transmits a new MAP message, whose effective starting time is t_9. Within this MAP, there is a data grant for the CM that starts at t_{11}.
- At t_8 the CM receives the MAP message and scans it for its data grant.
- At t_{10}, the CM transmits its data PDU so that it will arrive at the CMTS at t_{11}. Time t_{10} is calculated from the ranging offset.

As seen above, for a given interval, the CMs can use contention transmission. In this case, a CM selects a random number within its back-off window, which is determined by the CMTS. This random number indicates the number of *transmit opportunities* that the CM must skip before it starts its transmission. (For simplicity, assume that each mini-slot is a transmit opportunity, although this is not generally the case.) That is, if the CM has a back-off window from 0 to 10, and it randomly selects 8, then the CM must skip eight slots before it starts transmitting. For instance, let us assume that it has a bandwidth request to transmit to the CMTS, and let us assume that the request IE in the first MAP message defines an interval of six slots and in the second MAP message it defines an interval of four. Then, the CM will skip the mini-slots in the first interval, and it will transmit in the third mini-slot of the next interval.

In case of contention-based transmission, a CM does not know if its transmission has collided with a transmission from another CM. Therefore, the CM expects to receive a confirmation from the CMTS. This confirmation varies depending on what the CM transmitted. For instance, if it transmitted a bandwidth request, then it expects to receive a data grant. If it transmitted a data PDU, then it expects to receive an acknowledgement. If it does not receive a confirmation in a subsequent MAP message, the CM repeats the above process by doubling its back-off window. This re-try process is repeated 16 times, which is the maximum number of retries, after which the CM discards the MAC frame.

11.2.4 Quality of Service (QoS)

In the MAC protocol, a QoS scheme has been defined which provides different priorities for the transport of different flows of packets across the cable network. The QoS scheme uses the concept of *service flow*. This is a unidirectional flow of packets from a CM to the CMTS, or from the CMTS to a CM. The CMTS and CM provide QoS by shaping, policing, and scheduling the transmission of packets associated with a service flow.

In a basic CM implementation, only two service flows are used: an upstream flow and a downstream flow. These two service flows are known as *primary service flows*. These two primary service flows are used to implement best effort service. In more complex modems, multiple service flows can be implemented in addition to the primary ones. The additional service flows can be established to provide different service classes. For the network to function properly, all CMs must support at least the primary service flows.

A service flow could be in different states, such as *provisioned, admitted*, and *active*. To keep things simple, we will only discuss active service flows, service that is, flows for which data packets are being transmitted across the cable network. An active upstream service flow is assigned a unique *Service Id (SID).*

Packets delivered to the MAC protocol from the upper-level bridge or router are classified by a classifier into different service flows. The classifier can use the packet's IP address, for instance, to decide which service flow the packet belongs to. There is a downstream classifier in the CMTS, and an upstream classifier in a CM. Classifiers and service flows can be provisioned into the access network via a provisioning server, or they can be set up dynamically by a CM or the CMTS.

A service flow is associated with QoS parameters (e.g. delay and throughput) and with a *scheduling service*. Based on the QoS parameters and the scheduling service, the CMTS can provide opportunities to the CM to send up bandwidth requests for the service flow and receive data grants, so that the QoS associated with the service flow is achieved. The following scheduling services have been defined:

- Unsolicited grant service (UGS)
- Real-time polling service (rtPS)
- Unsolicited grant service with activity detection (USG-AD)
- Non-real-time polling service (nrtPS)
- Best effort (BE) service

The *unsolicited grant service (UGS)* is designed to support real-time service flows that generate fixed-size data packets on a periodic basis, such as voice over IP. The CMTS provides fixed size data grants to the flow on a periodic basis, so that to meet the flow's real-time needs. This eliminates the need for the CM to send bandwidth requests to the CMTS.

The *real-time polling service (rtPS)* was designed to support real-time service flows that generate variable size data packets on a periodic basis, such as MPEG video. The CMTS solicits periodically bandwidth requests from the CM and allows the CM to specify the size of the data grant.

The *unsolicited grant service with activity detection (USG-AD)* was designed to support real-time service flows, such as those supported by the *unsolicited grant service (UGS)*, such as voice over IP with silence suppression, which can become idle for substantial portions of time (i.e., tens of milliseconds or more). The CMTS provides periodic data grants when the flow is active. When the flow is inactive, it solicits bandwidth requests from the CM for this flow.

The *non-real-time polling service (nrtPS)* was designed to support non-real time flows that require variable size data grants on a regular basis, such as high-bandwidth file transfers. The CMTS solicits bandwidth requests from the CM for this flow on a periodic or non-periodic time interval on the order of one second or less.

Finally, the *best effort (BE) service* provides an efficient service for best effort traffic. The CMs send bandwidth requests on a contention basis in order to receive data grants for the flow.

11.3 THE ATM PASSIVE OPTICAL NETWORK

An APON is a cost-effective alternative to the telephone-based and cable-based access networks. An APON consists of an *optical line terminator (OLT)*, an *optical distribution network (ODN)* and *optical network units (ONU)*. The OLT, which resides at the premises of the APON operator, is responsible for transmitting and receiving traffic to and from the ONUs, which reside at the customer site. Also, the OLT has interfaces to a packet-switching backbone network. The OLT is connected to multiple ONUs via an optical distribution network. An APON, as its name implies, was designed with a view to carrying ATM traffic.

As shown in the example given in Figure 11.18, the optical distribution network consists of optical fibers connected in the form of a tree. The signal transmitted from the OLT is passively split between multiple fibers, each leading to a different ONU. Passive splitters (i.e., without electronic components), indicated by circles in Figure 11.18, are used to split the signal. These are made by twisting and heating optical fibers until the power output is evenly distributed. When a signal is split, there is always power loss, which means that there is a limit on how many times it can be split.

An APON is a point-to-multipoint broadcast system in the downstream direction (i.e., from the OLT to the ONUs), and a multipoint-to-point shared medium in the upstream direction (i.e., from the ONUs to the OLT). The OLT transmits ATM cells which are received by all of the ONUs. The transmitted cells are scrambled using a churning key so that an ONU cannot read the cells destined to another ONU. Each ONU selects only the cells destined for it. In the upstream direction, only one ONU can transmit at a time;

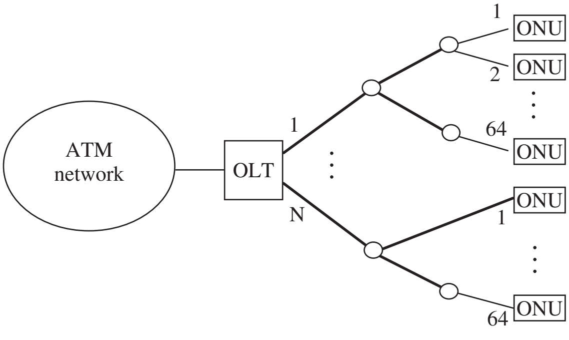


Figure 11.18 An example of an APON.

otherwise, cells transmitted from different ONUs might collide. A medium access protocol permits users to transmit in the upstream direction without collisions. The mechanism used for the downstream and upstream transmission is described below.

An example of downstream/upstream transmission is given in Figure 11.19. The OLT transmits three cells: one for ONU A, one for ONU B, and one for ONU C. (In Figure 11.19, a cell is represented by a square, with the name of the destination ONU written inside.) The optical signal carrying these cells is split into three, and each ONU receives the same optical signal with all three ATM cells, of which it reads only the one destined for it. In the upstream direction, each ONU transmits one cell, and thanks to the medium access mechanism, the cells arrive at the OLT one after the other without any collisions. In this example, collisions can only occur on the link between the splitter, indicated by the circle, and the OLT. The link between an ONU and the splitter is not shared by other ONUs. Each cell transmitted by an ONU is propagated to the splitter with no possibility of colliding with cells from other ONUs. If all three of the ONUs transmit a cell at the same time (and assuming that their distance from the splitter is the same), the cells will arrive at the splitter at the same time and will collide. The splitter will combine the three signals into a single signal, resulting in garbled information.

As can be deduced from the above discussion, the splitter has two functions. On the downstream direction it splits the signal, and in the upstream direction it combines the incoming signals into a single signal. Thus, it works as a splitter and as a combiner at the same time. The downstream and upstream signals are transmitted on different wavelengths, and thus it is possible for both transmissions to take place at the same time.

The *optical line terminator (OLT)* consists of an ATM switch, ATM interfaces to the backbone network, and ODN interfaces on the user side (see Figure 11.20). Each ODN interface serves a different APON, and there are as many APONs as ODN interfaces. For instance, in the example given in Figure 11.18, there are *N* ODN interfaces and *N* different APONs, and in the example given in Figure 11.19 there is a single APON.

APON was standardized by ITU-T in 1998 in recommendation G.983.1. APON has been defined by the *full service access networks (FSAN)* initiative as the common optical transport technology. FSAN is an initiative from telecommunication operators and manufacturers formed in 1995 to develop a consensus on the system required in the local access network to deliver a full set of telecommunications services both narrowband and broadband.

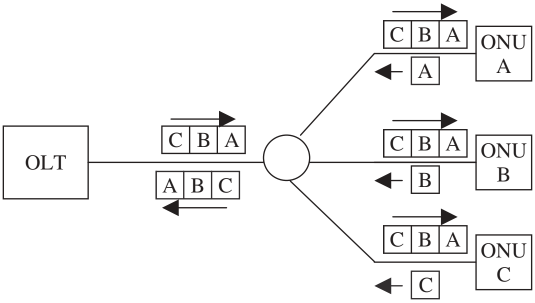


Figure 11.19 An example of downstream/upstream transmission.

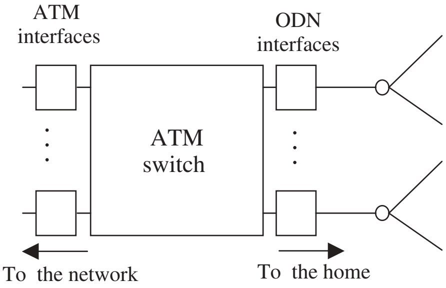


Figure 11.20 The optical line terminator (OLT).

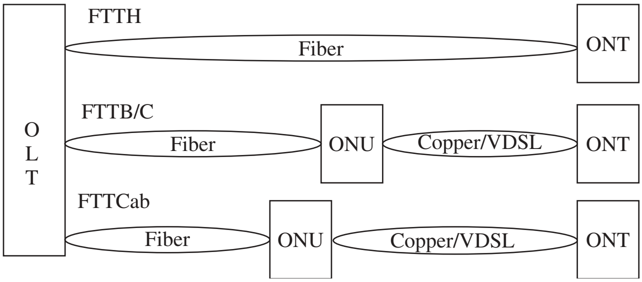


Figure 11.21 The G.983.1 network architecture.

The G.983.1 network architecture is shown in Figure 11.21 Depending on the location of the ONU, we have the following three possible configurations:

- *Fiber to the home (FTTH)*: The ONU is in the home.
- *Fiber to the basement/curb (FTTB/C)*: The ONU is in a building or a curb. Distribution to the home is done over copper using ADSL or VDSL.
- *Fiber to the cabinet (FTTCab)*: The ONU is in a cabinet, and distribution to the home is done over copper via ADSL or VDSL.

The FTTB/C and FTTCab are treated the same by the G.983.1 network architecture. An ONU terminates the optical access network and provides user-side interface(s) over copper using ADSL/VDSL. An *optical network terminator (ONT)* is the device used at the customer site. For simplicity we shall refer to the ONTs as ONUs.

The APON architecture uses different wavelengths for the downstream and upstream transmissions. A wavelength is an optical signal channel carried across an optical fiber (see Section 8.3.1). A transmitter emits a laser at a specific wavelength which is used as the carrier of the optical signal. The laser is modulated by the digital signal to produce an optical signal which is guided through the optical fiber to an optical receiver. The APON architecture uses two wavelengths for downstream transmission and one wavelength for upstream transmission. Specifically, for downstream transmission a wavelength in 1490 nm is used for transmitting ATM traffic and a second one in 1559 nm is used for video distribution. For the upstream transmission, it uses a wavelength in 1310 nm which is shared by all of the ONUs. The G.983.1 standard also permits the use of additional unidirectional fibers, operating at 1310 nm.

Two transmission options can be used: *symmetric* and *asymmetric*. In the symmetric option, both the upstream and downstream transmission rate for ATM traffic is 155.52 Mbps. In the asymmetric option, the downstream and upstream transmission rate for ATM traffic is 622.08 Mbps and 155.52 Mbps, respectively. The maximum fiber distance from an ONU to an OLT is 20 km, the minimum supported number of splits for a passive splitter is 16 or 32, and the minimum number of ONUs supported in an APON is 64. These specifications will probably change as the technology evolves.

APON can provide high-speed access for Internet traffic, voice over ATM, voice over IP and video services. APON can be deployed in new neighborhoods and municipalities. In a new neighborhood, the fiber can be laid at the same time as the infrastructure. Municipalities have an interest in providing high-speed connectivity to their residents, and they can easily deploy APONs by passing the fiber through existing underground conduits that lead close to the homes. Also, power companies are a potential provider since they can deploy the fiber using the existing poles that support the electrical cables!

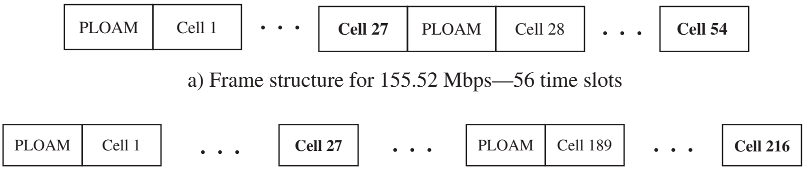
11.3.1 Frame Structures for Downstream and Upstream Transmission

The downstream transmission for both 155.52 Mbps and 622.08 Mbps consists of a continuous stream of time slots. Each time slot consists of 53 bytes, and it contains either an ATM cell or a 53-byte *physical layer OAM (PLOAM)* cell. PLOAM cells are only used every 28th time slot. Their function is explained in the following section. Groups of time slots are organized into *frames*.

The frame structure for 155.52 Mbps and 622.08 Mbps transmission rates is shown in Figure 11.22. For the 155.52-Mbps transmission rate, the frame consists of 56 time slots, two of which are PLOAM cells and the remaining 54 are ATM cells. As can be seen, the first time slot of the frame carries a PLOAM cell and the remaining 27 time slots contain ATM cells. This group of 28 time slots repeats once more within the frame. That is, the 29th time slot is a PLOAM cell, and the remaining 27 time slots carry ATM cells.

The frame for 622.08-Mbps transmission rate consists of 244 time slots, of which eight are PLOAM cells, and the remaining 216 are ATM cells. The first time slot of a frame contains a PLOAM cell and the next 27 time slots contain ATM cells. This group of 28 time slots repeats seven more times within the frame.

The frame structure for the upstream transmission is shown in Figure 11.23. The upstream frame consists of 53 time slots. Each slot consists of 56 bytes, of which the first 3 bytes are used for overheads and the remaining 53 bytes carry either an ATM cell



a) Frame structure for 155.52 Mbps—56 time slots

b) Frame structure for 622.08 Mbps—224 time slots

Figure 11.22 Frame structures for downstream transmission.

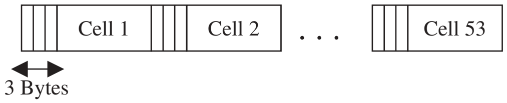


Figure 11.23 Frame structure for upstream transmission – 53 time slots.

or a PLOAM cell, or a *divided-slots* cell (the function and structure of this type of cell will be explained below in Section 11.3.3).

The overhead bytes carry the following information: a guard time, a preamble, and a delimiter. The guard time provides enough distance between two consecutive cells or mini-slots in a divided-slot cell to avoid collisions. The minimum guard time is 4 bits. The delimiter is a unique pattern used to indicate the start of an ATM or a PLOAM cell or the start of a mini-slot in a divided-slots. It is used to acquire bit synchronization.

The upstream and downstream frames are synchronized at the OLT. Figure 11.24 shows the frame alignment for the symmetric option (i.e. 155.52 Mbps downstream and 155.52 Mbps upstream). Upstream cells are aligned to the frame using the ranging procedure. Figure 11.25 shows the frame alignment for the asymmetric option (i.e. 622.08 Mbps downstream and 155.52 Mbps upstream).

11.3.2 The PLOAM Cell

PLOAM cells are used to convey physical layer OAM messages in the downstream and upstream direction. In addition PLOAM cells can carry grants which are used by the ONUs to transmit in the upstream direction.

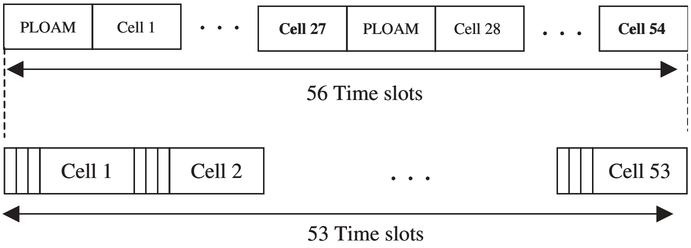


Figure 11.24 Frame alignment for the symmetric option.

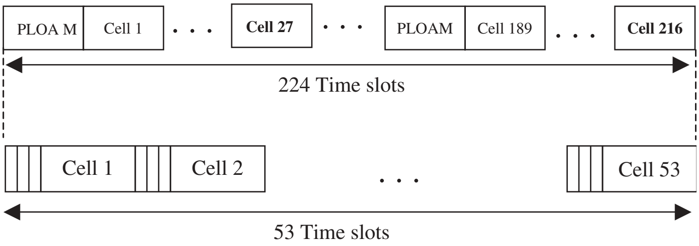


Figure 11.25 Frame alignment for the asymmetric option.

Table 11.3 The structure of the downstream PLOAM cell.

Byte	Description	Byte	Description
1	IDENT	25	GRANT 20
2	SYNC 1	26	GRANT 21
3	SYNC 2	27	CRC
4	GRANT 1	28	GRANT 22
5	GRANT 2	29	GRANT 23
6	GRANT 3	30	GRANT 24
7	GRANT 4	31	GRANT 25
8	GRANT 5	32	GRANT 26
9	GRANT 6	33	GRANT 27
10	GRANT 7	34	CRC
11	CRC	35	MESSAGE_PON_ID
12	GRANT 8	36	MESSAGE_ID
13	GRANT 9	37	MESSAGE_FIELD 1
14	GRANT 10	38	MESSAGE_FIELD 2
15	GRANT 11	39	MESSAGE_FIELD 3
16	GRANT 12	40	MESSAGE_FIELD 4
17	GRANT 13	41	MESSAGE_FIELD 5
18	GRANT 14	42	MESSAGE_FIELD 6
19	CRC	43	MESSAGE_FIELD 7
20	GRANT 15	44	MESSAGE_FIELD 8
21	GRANT 16	45	MESSAGE_FIELD 9
22	GRANT 17	46	MESSAGE_FIELD 10
23	GRANT 18	47	CRC
24	GRANT 19	48	BIP

The structure of the PLOAM cell in the downstream direction is shown in Table 11.3. The following 1-byte fields have been defined:

- *IDENT*: Bits 1 to 7 are set to 0. Bit 8 is set to 1 for the frame's first PLOAM cell, and to 0 for the all of the frame's subsequent PLOAM cells.
- *SYNC 1, SYNC 2*: A 1 KHz reference signal provided by the OLT is transported to the ONUs using these two bytes. It is used by the ONUs to synchronize to the downstream frame.
- *GRANT fields*: Provide grants to the ONUs for upstream transmission.
- *MESSAGE fields*: Transport alarms and threshold-crossing alerts.
- *Bit interleaving parity (BIP)*: Monitor the *bit-error rate (BER)* on the downstream link.
- *CRC*: Groups of seven grants are protected with a CRC with pattern: $x^8 + x^2 + x + 1$. No error correction is done.

The grants are permits issued by the OLT to the ONUs to transmit in the next upstream frame. Recall from Section 11.3.1 that a downstream frame is synchronized with an upstream frame. Grants transmitted in downstream frame i are used by the ONUs to transmit in the next upstream frame $i + 1$. The ONUs transmitting during the ith upstream frame received their grants in the $i - 1$ downstream frame.

Only 53 grants are required to be sent in a downstream frame, since an upstream frame has only 53 time slots. Each grant is a permission for a particular ONU to transmit in a specific time slot in the next frame. Each PLOAM cell contains only 27 grants (see Table 11.3). Therefore, the first two PLOAM cells in a frame (whether it is a 56 or 224 time-slot frame) suffice to carry all 53 grants. In fact, they carry a total of 54 grants, but the last one is an *idle grant*; that is, it is ignored by the ONUs. Idle grants are also carried in the remaining PLOAM cells if the frame has 224 time slots. The following types of grants have been defined:

- *Data grant*: During the ranging protocol (see Section 11.3.5), an ONU is assigned a PON-ID number between 0 and 63. A data grant contains the PON-ID number of an ONU. This is an indication to the ONU to send an ATM cell, or an idle cell if it has no data to send, in the next upstream frame in the time slot that corresponds to the position of the grant relative to the set of all of the grants in the frame.
- *PLOAM grant*: Indicates the PON-ID of an ONU. The ONU sends a PLOAM cell in the next upstream frame in the time slot that corresponds to the position of the grant relative to the set of all of the grants in the frame.
- *Divided-slots grant*: Indicates a group of ONUs. Each ONU in the group sends a mini-slot in the next frame (see Section 11.3.3).
- *Reserved grants*: Reserved for future grants.
- *Ranging grants*: Used in the ranging protocol.
- *Unassigned grants*: Indicates an unused upstream slot.
- *Idle grant*: Ignored by the ONUs.

The following fields have been defined for the message field:

- *MESSAGE_PON_ID*: Contains the PON-ID number of an ONU for which the message is addressed to. The field can also be set to 0×40 for broadcasting to all ONUs.
- *MESSAGE_ID*: Indicates the type of message.
- *MESSAGE_field*: Contains the message.
- *CRC*: The message fields are protected with a CRC with pattern: $x^8 + x^2 + x + 1$. No error recovery is done.

The PLOAM cell in the upstream direction is used to convey physical layer OAM messages. Its structure is shown in Table 11.4. The fields IDENT (it is set to 0), MESSAGE_PON_ID, MESSAGE_ID, MESSAGE_FIELD, CRC, and BIP are the same as those defined above in the downstream PLOAM cell. The LCF (laser control field) bytes and RXCF (receiver control field) bytes are used by the physical layer. A PLOAM cell is issued by an ONU in response to a PLOAM grant transmitted in a PLOAM cell in the downstream frame.

11.3.3 The Divided-slots Cell

Recall that the upstream frame carries 53 time slots. Each slot consists of 56 bytes, of which the first 3 bytes are used for overheads and the remaining 53 bytes carry either an ATM cell or a PLOAM cell, or a divided-slots cell. The structure of a divided-slots cell is shown in Figure 11.26. We see that it consists of a number of mini-slots, with each mini-slot consisting of a 3-byte overhead and a variable-length payload. The payload can

Table 11.4 The structure of the upstream PLOAM cell.

Byte	Description	Byte	Description
1	IDENT	25	LCF 11
2	MESSAGE_PON_ID	26	LCF 12
3	MESSAGE_ID	27	LCF 13
4	MESSAGE_FIELD 1	28	LCF 14
5	MESSAGE_FIELD 2	29	LCF 15
6	MESSAGE_FIELD 3	30	LCF 16
7	MESSAGE_FIELD 4	31	LCF 17
8	MESSAGE_FIELD 5	32	RXCF 1
9	MESSAGE_FIELD 6	33	RXCF 2
10	MESSAGE_FIELD 7	34	RXCF 3
11	MESSAGE_FIELD 8	35	RXCF 4
12	MESSAGE_FIELD 9	36	RXCF 5
13	MESSAGE_FIELD 10	37	RXCF 6
14	CRC	38	RXCF 7
15	LCF 1	39	RXCF 8
16	LCF 2	40	RXCF 9
17	LCF 3	41	RXCF 10
18	LCF 4	42	RXCF 11
19	LCF 5	43	RXCF 12
20	LCF 6	44	RXCF 13
21	LCF 7	45	RXCF 14
22	LCF 8	46	RXCF 15
23	LCF 9	47	RXCF 16
24	LCF 10	48	BIP

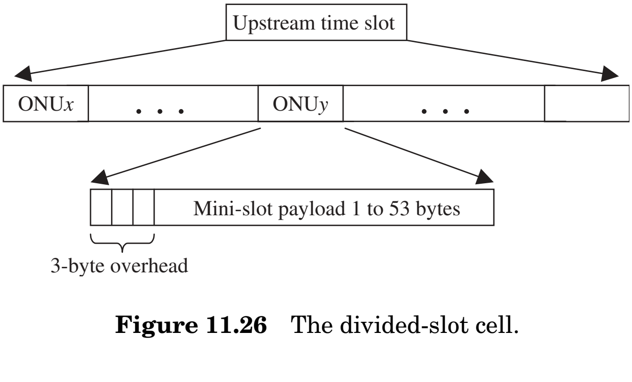


Figure 11.26 The divided-slot cell.

vary from 1 byte to 53 bytes. The three-byte overhead is the same as the one used with each cell in the upstream frame.

Each divided slots cell is associated with a group of ONUs. The OLT assigns one divided-slots cell to group of ONUs using the grant field in the PLOAM cell. Each mini-slot contains information from one ONU belonging to the group, such as the size of its queue of packets awaiting transmission to the OLT. This information can be used by the OLT in a dynamic manner to determine the transmission schedule of the ONUs.

The round robin scheme gives each ONU the same opportunity to transmit, and it does not know whether an ONU has traffic to send up or not. As a result, it is possible that some of the upstream cells can be wasted if some of the ONUs transmit idle cells. The round robin algorithm can be modified so that it hands out grant permits only to those ONUs that have an ATM cell to transmit. Another modification will be to grant a different number of data grants to the ONUs depending upon how many ATM cells are waiting in their queues for upstream transmission. In order to implement these modifications, the OLT will require to have some knowledge about the queue size in each ONU. This information can be delivered to the OLT using the mini-slots cell. The actual MAC protocol that decides how much bandwidth should be allocated to each ONU based on information regarding the ONU's queue size is beyond the scope of the G.983.1 standard.

11.3.4 Churning

As we have seen so far, the downstream transmission is shared by all of the ONUs in the APON. ATM cells transmitted to an ONU are in fact received by all ONUs. An ONU selects its own ATM cells based on their VPI/VCI value, and ignores all of the other ATM cells. A similar situation arises in the DOCSIS cable modems. The security of data transmission is critical in protecting the privacy of users and the confidentiality of their communications. In view of this, the APON provides a scrambling technique, known as *churning*. Each ONU sends a churning key to the OLT using the new_churn_key message. The churning keys are used by the OLT to scramble the data. Each ONU updates its churning key once every second. Additional security can be provided using encryption at a higher layer.

11.3.5 Ranging

In an APON, the ONUs can be at different distances from the OLT, ranging from 0 to 20 km. Because of this, cells transmitted by different ONUs might partially overlap, and consequently collide, in the upstream transmission. To avoid collisions, each ONU is placed at the same virtual distance from the OLT. The process that measures the distance between the OLT and each ONU, and places each ONU in the same virtual distance is known as *ranging*. Each ONU is given an *equalization delay*, which is used by the ONU to adjust its time of transmission.

PROBLEMS

1. You have to download a 20-MB file to your computer at home.
   a) How long does it take to download it assuming that you are connected with a 56K modem, that gives a throughput of 49 Kbps?
   b) How long does it take to download the same file assuming that your computer is connected to an ADSL modem, which provides a throughput of 1 Mbps?
2. What is the difference between the fast path and the interleaved path in an ADSL modem?
3. What is the advantage of using the network access server (NAS)?
4. In L2TP, why are control messages transmitted over a reliable channel and not data messages?

5. Consider the pointer_field in an MPEG packet used in DOCSIS. What is its value for the following cases?
   a) The first DOCSIS MAC frame starts immediately after the pointer_field.
   b) The first *M* bytes after the pointer_field contain the tail of a DOCSIS MAC frame. Then, there are *B* stuff-bytes, after which a new DOCSIS MAC frame begins.
   c) A new DOCSIS MAC frame begins immediately after the pointer_field, followed by *B* stuff-bytes, followed by another DOCSIS MAC frame.
6. Explain the difference between FTTH and FTTB/C.
7. Explain how ranging is used in an APON to place each ONU at the same virtual distance.
8. In an APON, explain how the round robin algorithm can be modified so that grant permits are handed out only to those ONUs that have an ATM cell to transmit.

12

Voice Over ATM and MPLS

Voice over packet solutions have been developed for the IP network, ATM, frame relay, and MPLS. In this chapter, we explore the topic of voice over ATM and voice over MPLS. Both ATM and MPLS are suitable technologies for voice over packet because they can provide QoS, a necessary requirement for real-time traffic such as voice.

There are several reasons why it is beneficial to migrate voice from the TDM network to a packet-switching network. For instance, national and international operators typically have an extensive *public switched telephone network (PSTN);* they also operate data networks, such as IP with or without MPLS, frame relay, and ATM. Migrating voice to their data network permits them to maintain a single group of engineers and managers to run the network, rather than two separate groups, one for the PSTN and one for the data network.

Voice over packet technologies can also used by carriers, known as a *competitive local exchange carrier (CLEC)*, and value added network suppliers, to provide communication services in competition with pre-existing operators, known as the *incumbent local exchange carrier (ILEC)*. Typically, CLECs and value added network suppliers do not own the transmission infrastructure. In view of this, they either buy bandwidth from an ILEC, or form joint ventures with companies who have the right-of-way or have already deployed a network. In the case of a CLEC, therefore, integration of voice and data service makes sense due to cost and limited bandwidth availability. For instance, within the local loop, voice over packet permits a CLEC to provide voice and data services over cable modems, ADSL and APON, in addition to the voice and data services provided by the ILEC.

Voice over packet also provides a cost-performance solution for cellular operators who have to interconnect their cell sites and *message switching centers (MSC)*. Finally, in enterprise (private) networks a significant portion of the traffic is voice. Since a company buys bandwidth at commercial rates, integrating voice and data is a cost-effective solution.

The ATM Forum has defined several specifications for transporting voice over ATM. These standards can be organized into two groups. The first group of specifications, referred to as *ATM trunking for voice*, deals with the transport of voice over ATM between two telephone networks. The second group of specifications deals with how to provide voice over ATM to a user at a desktop or to a user over ADSL. In this chapter, we describe two of the ATM trunking for voice specifications: *circuit emulation services (CES)* and *ATM trunking using AAL 2 for narrowband services*. Circuit emulation services emulate a TDM link, such as a T1 or E1 link, over an ATM network using AAL 1. The basic

Connection-oriented Networks Harry Perros
ISBN: 0-470-02163-2

operation of circuit emulation services and AAL 1 is described in Section 3.7.1. The ATM trunking using AAL 2 for narrowband services specification is used to transport voice traffic between two distant private or public telephone networks

The MPLS and Frame Relay Alliance has so far defined two different specification for voice over MPLS. These two specifications use ATM's AAL 1 and AAL 2 protocols. The first specification deals with circuit emulation services over MPLS, and it makes use of AAL 1. The second specification deals with the transport of voice over MPLS and it uses AAL 2. Both specifications are described in this chapter.

Below, basic telephony concepts and signaling protocols are reviewed. Subsequently, we describe two ATM specifications: *circuit emulation services (CES)* and *ATM trunking using AAL 2 for narrowband services*. The latter specification is based on two specially designed AAL 2 service-specific convergence sublayers *(AAL 2 SSCS for trunking* and *segmentation and reassembly SSCS [SEG-SSCS])*. Because these two SSCS specifications play a very important role in AAL 2 trunking, they are also described in detail. The chapter concludes with the two specifications proposed by the MPLS and Frame Relay Alliance.

12.1 BACKGROUND

To understand the voice over ATM and MPLS solutions, the reader has to be familiar with basic telephony terms and signaling protocols that are used in the existing telephone system. In this section, we first review some of the basic terms used in telephony and describe the basic signaling steps involved in placing a call. Subsequently, we briefly present the *channel-associated signaling (CAS) scheme*, the *signaling system no. 7 (SS7), narrowband ISDN (N-ISDN)*, and the *digital subscriber signaling system no. 1 (DDS1).*

12.1.1 Basic Concepts

Figure 12.1 gives an example of part of a telephone network. Telephone switches (indicated in Figure 12.1 by A and B) are known as *exchanges*. Exchanges A and B serve a number of subscribers (indicated in Figure 12.1 by circles). C is also an exchange, but it does not have any subscribers. A and B are known interchangeably as *local exchanges,*

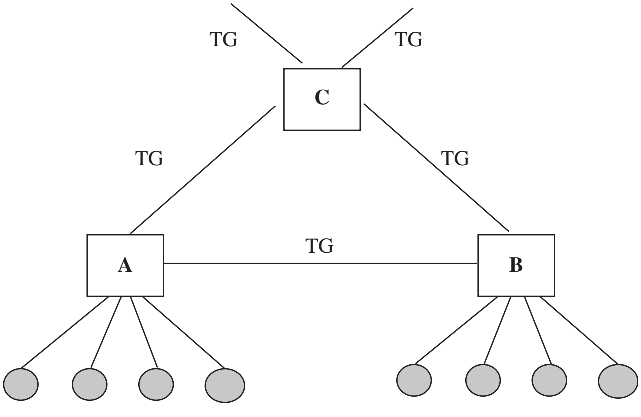


Figure 12.1 An example of a telephone network.

central offices, or *end offices*; C is known interchangeably as an *intermediate, tandem, toll*, or *transit* exchange.

A *trunk* is a circuit between two exchanges, and a group of trunks is known as a *trunk group (TG)*. A trunk is nothing else but a channel associated with a time slot in a T1/E1 link or in a SONET/SDH link, which carries a single voice call. A *subscriber*, often referred to as a *customer* or *user*, is connected to its local exchange via a subscriber line which is commonly referred to also as the *local loop*.

A private branch exchange (PBX) is an exchange owned by an organization, such as a business, a university, and a government agency. It is located in a building that belongs to the organization and is connected to a nearby local exchange. A PBX enables the employees in the organization to call each other and also place and receive calls to/from outside the organization. Typically, a PBX is connected to the telephone network via a T1 or E1 link.

The main steps involved in placing a call from one subscriber to another are shown in Figure 12.2. For simplicity, assume that both subscribers, S1 and S2, are attached to the same local exchange. S1 lifts the handset of the telephone from its cradle. This is known as *off-hook*, which is interpreted by the local exchange as a request for service. The local exchange generates a dial tone, which indicates to S1 that it is ready to receive the digits. S1 punches the digits, which are transferred one at a time to the local exchange.

Earlier telephones used *dial-pulse* to generate the digits. The dial is rotated by the subscriber, and when it is released it spins back to its rest position producing a string of breaks in the dc path. Each string of breaks represent a different number. For instance, one break

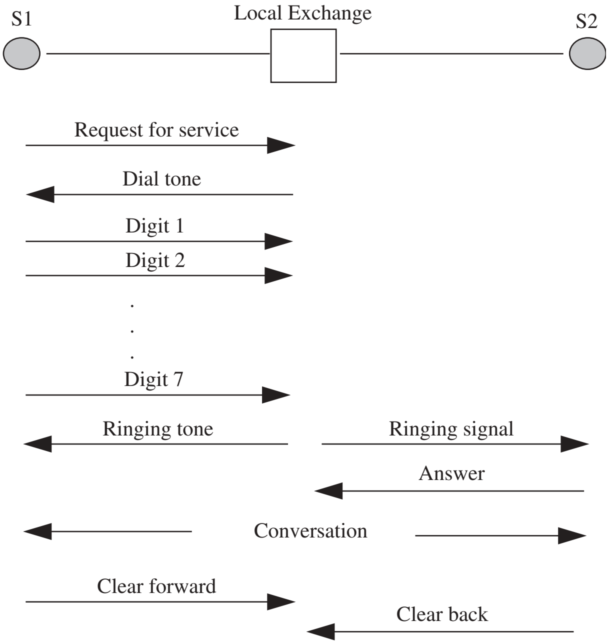


Figure 12.2 Basic signaling.

represents 1, two breaks represent 2, etc. The dial-pulse system was replaced by the *dual-tone multi-frequency (DTMF)* system. When a subscriber presses a key on the keypad, an oscillator inside the telephone generates two simultaneous tones. Each digit is represented by a particular combination of two frequencies, one selected from a low group of frequencies (697, 770, 852, and 941 Hz) and the other selected from a high group of frequencies (1209, 1336, 1477, and 1622 Hz). This allows for sixteen digit values, of which only twelve are used for digits 1, 2, . . ., 9, 0, and special values # and *. The DTMF frequency combinations are distinct from naturally occurring sounds in the vicinity of the calling subscriber and also from voice and voiceband data (i.e. modem traffic and facsimile).

After the digits have been transferred to the local exchange, the local exchange checks to see if the called party's phone is free. If it is busy, it sends a busy signal to S1. If it is free, it sends a ringing signal to alert S2, and it also informs S1 that it is in the process of alerting S2 by sending a ringing tone. When S2 answers the phone, the local exchange sets up a path through its switch to connect the two subscribers, and at that moment the conversation can begin. Finally, the two subscribers go *on-hook*, i.e., they hang up, which generates a clear forward message from the calling subscriber and a clear back message from the called subscriber. (The terms *on-hook* and *off-hook* come from the days when the receiver of the telephone was resting on a hook!)

12.1.2 Channel-Associated Signaling (CAS)

Let us consider two subscribers (i.e. S1 and S2) that are interconnected via exchanges A, B, and C (see Figure 12.3). Exchanges A and B, and B and C are connected by a PDH link (e.g. T1/E1 or T3/E3). Establishing a call between S1 and S2 requires establishing a two-way circuit between S1 and S2. The circuit from S1 to S2 consists of: S1's subscriber line to local exchange A; a time slot (say time slot i) in the frame transmitted from A to B; a time slot (say time slot j) in the frame transmitted from B and C; and S2's subscriber line. In the direction from S2 to S1, the circuit consists of S2's subscriber line to local exchange C, time slot j in the frame transmitted from C to B, time slot i in the frame transmitted from B and A, and S1's subscriber line. These physical resources are entirely dedicated to the call between S1 and S2 and cannot be shared by other calls. These resources are freed when the call is terminated which happens when either side hangs up.

The *channel-associated signaling (CAS)* is used to establish and release calls, and it has been in existence since the beginning of automatic telephony. It was the only signaling system used until the late 1970s, when the *common channel signaling (CCS)* was developed. CAS is still being used, but it is gradually been replaced by CCS. As will be explained below, CAS uses various signals to establish and release a call. In addition, it uses a number of supervisory bits, which are known as the *ABCD signaling bits*. The signals, the digits coded in DTMF, and the ABCD bits are all transmitted through the same circuit, i.e. the same time slots between adjacent exchanges, that are used for the transmission of the voice traffic of the call. In view of this, CAS is an *in-band* signaling protocol.

Note that the detailed signaling between S1 and its local exchange A was discussed in the previous section, and is not shown in Figure 12.3. After exchange A receives the last digit from S1, it seizes a free trunk between A and B which will be used for this new voice call. The trunk is an unused time slot in the frame traveling from A to B.

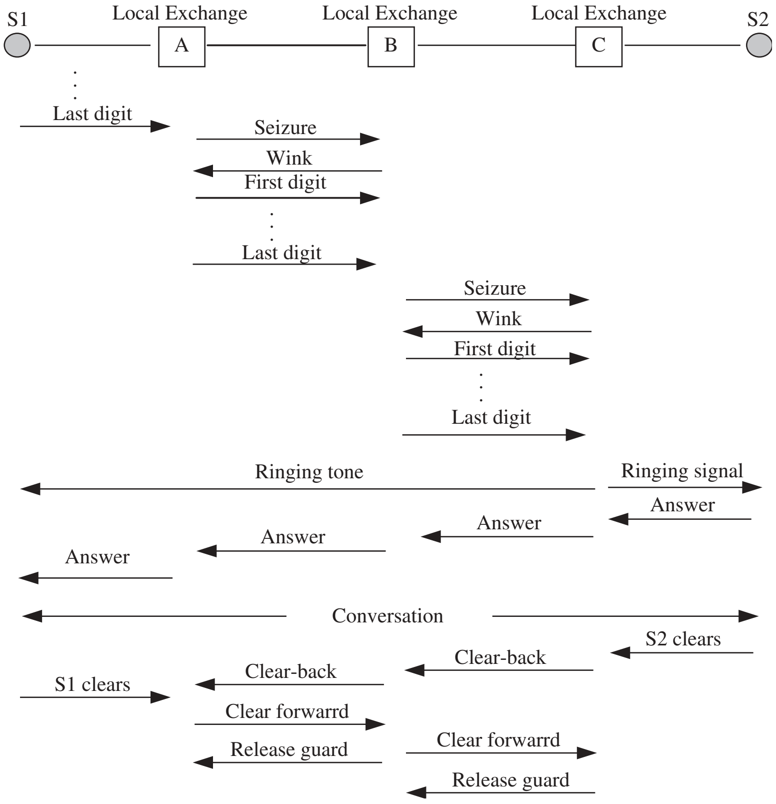


Figure 12.3 CAS signaling.

Exchange A sends a special *seizure signal* to B on this time slot. Exchange B responds with a *proceed-to-send signal* (also called a *wink signal*). This signal is sent on the same time slot number in the frame traveling in the direction from B to A, to indicate that it is ready to receive the digits of the called party.

Exchange A sends the digits to B, by transmitting one digit in the time slot allocated to this call over a number of successive frames. Exchange B repeats the same procedure as A. That is, once it has received the entire number of the called party, it seizes an outgoing trunk, i.e., a time slot in the frame traveling from B to C, and sends a seizure signal to exchange C. Exchange C responds with a wink signal, after which exchange B sends the digits. Exchange C checks to see if the receiver's phone is idle. If it is idle, it generates a ringing signal and also sends a ringing tone towards S1. When S2 answers, exchange C sends an answer signal to exchange B, which exchange B forwards to exchange A. The conversation between S1 and S2 can now begin.

In the example in Figure 12.3, S2 hangs up first. This gives rise to the *clear-back message* sent from exchange C backwards towards exchange A. When exchange A receives the clear-back message, it stops charging for the call and sets up a timer. Exchange A releases the connection when it receives a clear message from S2 or the timer expires. This is done by sending a *clear-forward signal* to exchange B, which in turn sends it to

Table 12.1 ABCD signaling scheme.

Frame number	Transmitted bit
6	A
12	B
18	C
24	D
30	A
36	B
42	C
48	D

exchange C (see Figure 12.3). The *release guard signal* is used to indicate to the receiving exchange that it can use the same trunk for another call.

In addition to the signals mentioned above (e.g. seizure, wink, answer, clear-forward, clear-back, and release guard, and the dialed digits which are transported using the DTMF scheme), a number of *supervisory bits* are also used to indicate supervisory line states (e.g. on-hook, off-hook, idle, and ringing). Supervisory bits are known as the *ABCD signaling bits*. The ABCD bits are transferred by robbing the 8th bit of the time slot associated with the voice call every six frames. The robbed bit is used to transmit the A, B, C, and D bits, respectively (see Table 12.1). Let us consider a time slot associated with a particular voice call. Then, the A bit will be transmitted by robbing the 8th bit of the slot in the 6th frame, the B bit will be transmitted by robbing the 8th bit of the time slot in the 12th frame, the C bit will be transmitted by robbing the 8th bit of the time slot in the 18th frame, the D bit will be transmitted by robbing the 8th bit of the time slot in the 24th frame, and so on. The four bits provide for a 16-state signaling scheme for each voice channel.

12.1.3 Signaling System No. 7 (SS7)

As mentioned in the previous section, CAS is an in-band signaling protocol. In the *common-channel signaling (CCS)*, all signaling information for establishing and releasing a phone call is carried in messages over a separate packet-switching network. This packet-switching network is known as the *signaling network*. It consists of *signaling points (SP)* and *signaling transfer points (STP)*, which are interconnected by *signaling links (SL)*. An SP originates, receives, and processes signaling messages. It can be part of either a telephone exchange or a database, which is accessed via CCS messages. An STP is an SP that simply switches messages from an incoming SL to an outgoing SL. That is, it does not originate or process signaling messages.

The first generation common-channel signaling protocol was *signaling system no. 6 (SS6)*, which was introduced in the 1970s. SS6 was followed by *signaling system no. 7 (SS7)* about ten years later. SS7 consists of several layers (see Figure 12.4). The *message transfer part (MTP)* provides a reliable transfer service to the protocols running above it. It is divided into three parts: MTP1, MTP2, and MTP3. These three parts occupy the

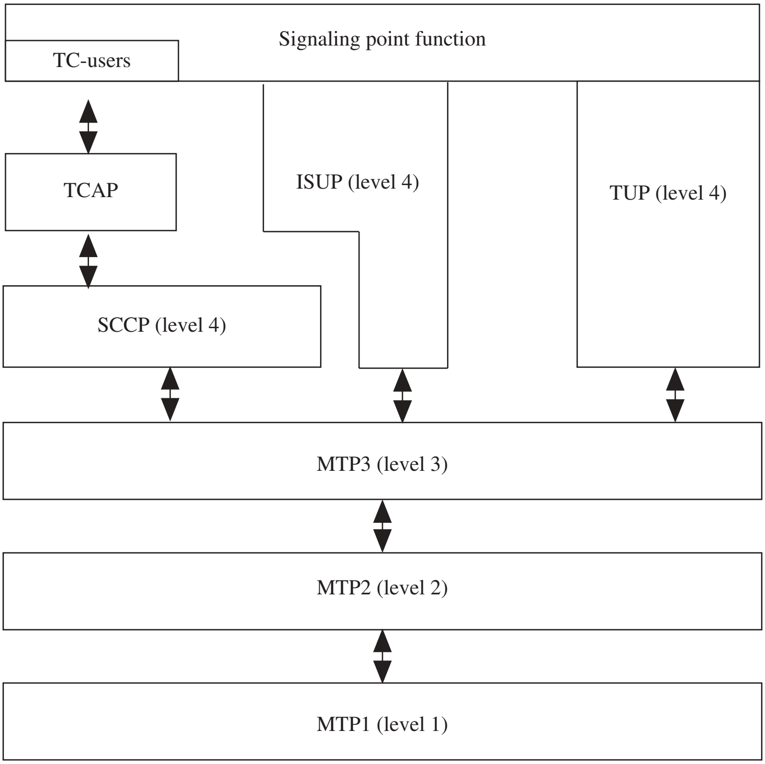


Figure 12.4 The SS7 stack.

first three levels of the SS7 hierarchy. The *telephone user part (TUP)* is a protocol for telephony call control and for trunk maintenance. The *integrated service user part (ISUP)* is a protocol for call control and trunk maintenance for both the telephone network and N-ISDN (see following Section 12.1.4). It supports signaling for calls in which either one or both parties are N-ISDN users. The *signaling connection control part (SCCP)* is a protocol that provides functions for the transfer of messages that are not trunk related. The combination of MTP and SCCP corresponds to the OSI Layers 1, 2, and 3. The *transaction capabilities application part (TCAP)* provides for operations that are not related to individual trunks and involve two signaling points. TCAP provides an interface to TC-users.

12.1.4 Narrowband ISDN (N-ISDN)

The *integrated service data network (ISDN)* was a concept of a future network put forward in the 1980s. This future network will be capable of providing a wide range of services for voice and non-voice applications, and all of these services will be provided by one network, rather than different networks. ISDN was seen as evolving progressively from the digital telephone network to include additional functions and network features, such as circuit-switching and packet-switching for data. The fundamental building block of ISDN was the 64-Kbps connection.

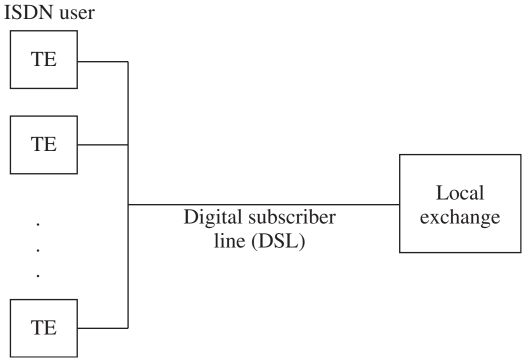


Figure 12.5 An N-ISDN user.

The first generation of ISDN is referred to as the *narrowband ISDN (N-ISDN)*. In this network, ISDN users can communicate with each other in circuit-switching mode and packet-switching mode. As shown in Figure 12.5, an ISDN user can operate multiple 64-Kbps digital *terminal equipment (TE)* of differing types, such as *digital telephone (PCM)*, high-speed facsimile terminal, and high-speed computer modem. TEs are connected to the local exchange via a *digital subscriber line (DSL)*. The DSL is a two-wire or four-wire line that allows simultaneous transmissions in both directions. The transmission rate in one direction is 144 Kbps.

The 144-Kbps bit stream is divided into two 64-Kbps B-channels and one 16-Kbps D-channel. The B-channels are used for circuit-mode communications. They can also be used to carry digital information, such as digitized voice, computer data, and video. Both B-channels can be used at the same time, permitting the user to operate two phones or a phone and a computer modem at the same time. The two B-channels can also be linked together to provide an aggregate 128-Kbps data channel. The D-channel is used for signaling between the user and the local exchange. The *digital subscriber signaling system no. 1 (DSS1)* is used (see Section 12.1.5). In addition, the D-channel can be used for low speed packet-switching. The B-channels and the D-channel are full-duplex.

This basic access scheme is known as the *basic rate* and it was standardized by ANSI in 1988. Because it makes use of two B-channels and a D-channel, it is also known as 2B + D. The basic rate was intended for residential users and very small offices.

As in the TDM schemes (see Section 2.1), the basic rate access transmission is organized into fixed-length frames that repeat continuously. Each frame is 48 bits long and it repeats every 250 μsec. The frame structure from the N-ISDN user to the local exchange is shown in Figure 12.6. Fields B1 and B2 contain data from the two B-channels. They are 8-bit long, and they repeat twice in the frame. The remaining fields (D, F, and L) are 1-bit long. D repeats four times within the frame, and is used to carry data from the D-channel. F and L carry a framing bit and a dc balancing bit, respectively. The F-L combination is used to synchronize the receiver at the beginning of the frame. Each frame contains 16 bits from each B-channel, and four bits from the D-channel. The bit rate is 144 Kbps, but due to the additional overheads bits, the total bit rate is 192 Kbps.

In addition to the basic rate, the *primary rate* was defined for users with greater requirements for bandwidth, such an organization with a digital PBX or a local network.

F	L	B1	L	D	L	F	L	B2	L	D	L	B1	L	D	L	B2	L	D	L

Figure 12.6 The frame structure from the N-ISDN user to the network.

F	Time slot 1	Time slot 2	Time slot 3	. . .	Time slot 24

(a) Frame structure for the 1.544 Mbps interfcae

Time slot 0	Time slot 1	Time slot 2	. . .	Time slot 31

(a) Frame structure for the 2.048 Mbps interfcae

Figure 12.7 Primary rate frame structures.

In the US, Canada, and Japan, the primary rate is 1.544 Mbps (the same rate as in DS1). In Europe, the primary rate is 2.048 (the same as in E1). The frame structure for the 1.544-Mbps interface and the 2.048-Mbps interface are shown in Figure 12.7.

The frame structure for the 1.544-Mbps interface consists of twenty-four 8-bit time slots and a framing bit (bit F). The frame consists of 193 bits, and it repeats every 125 μsec. Of the 24 time slots, 23 are used for 64-Kbps B-channels and one time slot is used for a 64-Kbps D-channel. This transmission structure is referred to as 23B + D.

The frame structure for the 2.048-Mbps interface consists of thirty-two 8-bit time slots. The frame consists of 256 bits, and it repeats every 125 μsec. The first 8-bit time slot is used for framing and synchronization. Of the remaining thirty-one 8-bit time slots, thirty are used for 64-Kbps B-channels, and one 8-bit time slot is used for a 64-Kbps D-channel. This transmission structure is referred to as 30B + D.

It is possible for a customer with a lower bandwidth requirement to employ fewer B-channels than those provided by the primary rate interface.

The interface for the primary rate can also support channels with a transmission rate higher than the B-channel. These higher rate channels are: the 384-Kbps H0-channel, the 1536-Kbps H11-channel, and the 1920-Kbps H12-channel. The primary rate interface H0 channel structure can support the combinations 3H0 + D and 4H0 for the 1.544-Mbps interface, and 5H0 + D for the 2.048-Mbps interface. The primary rate interface H1 channel structure can support the H11 channel structure consisting of one 1536-Kbps H11-channel, and the H12 channel structure consisting of one 1920-Kbps H12-channel and one D channel.

Primary rate interface structures have also been defined that permit 0 or 1 D-channel plus any combination of B-channels and H0-channels up to the capacity of the physical interface, such as 3H0 + 5B + D and 3H0 + 6B.

Frame relay

Frame relay is a networking architecture that was defined originally to interconnect N-ISDN users. As it turned out, it is possible to use frame relay as a stand-alone protocol for transporting data and more recently voice over a wide area network. Currently, frame relay is a popular networking solution, and is typically offered by public network operators.

Broadband ISDN (B-ISDN)

N-ISDN was the first step towards realizing the ISDN vision of providing a wide range of services for voice and non-voice applications over the same network. The second generation of ISDN that provides for very high speeds was referred to as the *broadband ISDN (B-ISDN)*. The *asynchronous transfer mode (ATM)* is the preferred network architecture for B-ISDN (see Chapters 3 – 5).

12.1.5 Digital Subscriber Signaling System No. 1 (DSS1)

The *Digital Subscriber Signaling System No. 1 (DSS1)* is used for signaling between a N-ISDN user and its local exchange. DSS1 is message oriented and many of DSS1 concepts are similar to SS7. DSS1 messages are transported over the D-channel. It is divided into the data link layer and the network layer. The data link layer, also known as the LAP-D link access protocol, is concerned with the reliable transfer of frames between the *terminal equipment (TE)* and its local exchange.

The network layer protocol is usually referred to by the ITU-T recommendation Q. 931, in which it was specified. The ATM signaling protocol Q. 2931 (described in Chapter 5) is based on Q.931, and is also referred to as the *digital subscriber signaling system no. 2 (DSS2)*. Q. 931 includes the following messages: setup (SETUP), setup acknowledgment (SETACK), call proceeding (CALPRC), progress message (PROG), alerting message (ALERT), connect (CONN), connect acknowledgment (CONACK), disconnect (DISC), release (RLSE), release complete (RLCOM), and information (INFO). The signaling takes place between the TE of an N-ISDN user and the local exchange to which the user is attached. The signaling messages are carried over the D-channel.

The message format consists of a protocol discriminator, call reference value, length, message type, and information elements, such as bearer capability, called party number, calling party number, called subaddress, cause, channel ID, high-layer capability, keypad, low-layer capability, progress indicator signal, transit network selection, and user-to-user information.

A typical scenario of a signaling sequence for setting up and tearing down a call is shown in Figure 12.8. In this example, we assume that the calling and called TEs are attached to the same local exchange. The calling TE sends a SETUP message to its local exchange to request the establishment of a call. The complete called number is included in the SETUP message. The local exchange forwards the SETUP message to the called TE, and it also sends a CALLPRC message to the calling TE to indicate that the call setup has started. Let us assume that the called TE accepts the request to set up a call. It alerts the called user with a visible or audible tone, and returns an ALERT message to the local exchange which forwards it to the calling TE. If the requested call is a voice call, the local exchange also connects a ringing tone to the calling TE. When the called user answers, the called TE sends a CONN message to the local exchange which forwards it to the calling TE to indicate that the call has been answered. CONNACK messages are sent from the calling TE to the exchange, and from the exchange to the called TE. At this point the two N-ISDN users can begin to talk or send data.

The call clearing procedure is also shown in Figure 12.8. A call can be cleared by either party. In this example, it is the called user that clears the call first. This is done by sending a DISC message to the local exchange, which forwards it to the calling TE. The B-channels are released using the RLSE and RLCOM messages.

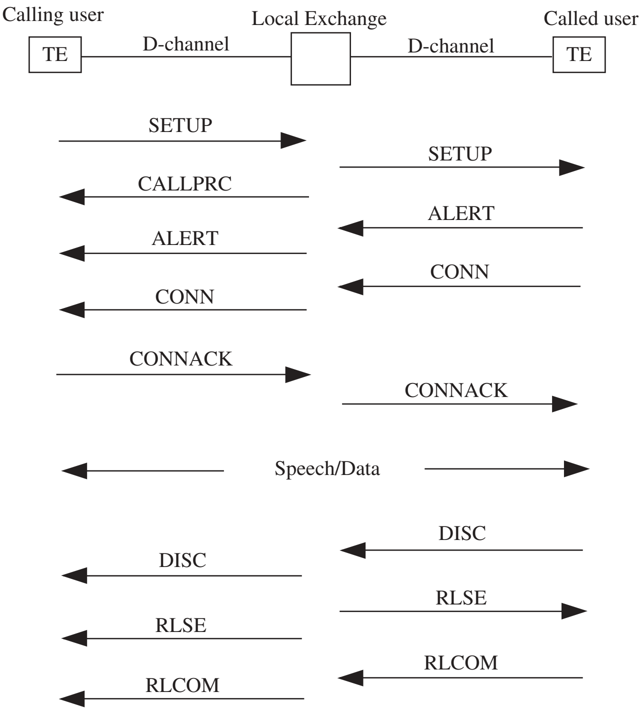


Figure 12.8 An example of DSS1 signaling.

12.2 VOICE OVER ATM SPECIFICATIONS

ATM is a technology well-suited for voice over packet, since it can guarantee the end-to-end delay, a necessary QoS parameter for the delivery of voice, and is also well-established in the backbone and in access networks, such as ADSL-based access networks.

The key to implementing voice over ATM is a device known in ATM as the *interworking function (IWF)*. In IP networks, an IWF is known as a *media gateway*. An IWF implements the functionality defined in a voice over ATM specification. It can be a standalone device or part of a bigger device. An IWF can offer a variety of services, such as the transport of PCM voice (i.e., 64-Kbps voice), encoded voice, facsimile, telephone signaling, and circuit mode data (i.e., T1/E1 and fractional T1/E1).

The ATM Forum has defined a number of different specifications for voice over ATM. These specifications can be classified into two groups. The first group, referred to as *ATM trunking for voice*, deals with the transport of voice between two telephone networks. The second group of specifications deals with how to provide voice to a user at a desktop or to a user over ADSL.

The following four specifications have been defined for ATM trunking for voice:

- *ATM trunking using AAL 1 for narrowband services*
- *Circuit emulation service*s *(CES)*
- *Dynamic bandwidth circuit emulation service*s *(DBCES)*
- *ATM trunking using AAL 2 for narrowband service*s

The first three specifications are based on AAL 1, whereas the fourth one is based on AAL 2. The first specification *ATM trunking using AAL 1 for narrowband services* addresses the interconnection between two PBXs over a private or public ATM network. The *circuit emulation services (CES)*, as shown in Section 3.7.1, emulates a point-to-point TDM circuit, and is used to connect TDM interfaces such as such as T1, T3, E1, E3, and J2 (the J system is used in Japan), over an ATM network. The *dynamic bandwidth circuit emulations service*s *(DBCES)* specification provides circuit emulation services for fractional T1/E1, as in the CES specification, but in addition it permits the bandwidth allocated to the ATM connection that carries the fractional T1/E1 signal to be dynamically changed. Finally, the *ATM trunking using AAL 2 for narrowband service*s specification was designed so that it can interconnect two distant public or private telephone networks over an ATM network. It can be used, for instance, to connect distant PBXs to a central office over an ATM network, or to interconnect two distant PBXs. This specification is based on two AAL 2 service-specific convergence sublayers: the *AAL 2 SSCS for trunking* and the *segmentation and reassembly service-specific convergence sublayer for AAL 2 (SEG-SSCS)*. The *AAL 2 SSCS for trunking* specification was designed to provide a variety of services, such as audio service, circuit-mode service, and frame-mode data service. The SEG-SSCS specification was designed to transport packets whose size is bigger than the maximum length of 45 bytes permitted in the payload of the CPS packet.

The following specifications have been defined for the second group of specifications, which deals with how to provide voice to a user at a desktop or to a user over ADSL:

- *Voice and Telephony over ATM to the Desktop*
- *Loop Emulation Servic*e *(LES) using AAL 2*

The first specification, as its name implies, deals with providing voice to the desktop, and the second specification deals with providing narrowband services to a user over ADSL, *hybrid fiber coaxial (HFC)*, or a wireless connection.

Below, we first review briefly the circuit emulation services specification and then we describe in detail the ATM trunking using AAL 2 for narrowband services specification. (The reader is assumed to be familiar with the material on ATM adaptation layers 1, 2, and 5; if needed, review Section 3.7.)

12.3 THE CIRCUIT EMULATION SERVICES (CES) SPECIFICATION

Circuit emulation services (CES) emulates a point-to-point TDM circuit. CES is used to connect TDM interfaces such as such as T1, T3, E1, E3, and J2, over an ATM network. CES is based on AAL 1 and it uses the CBR service category in order to guarantee the end-to-end delay. Both the unstructured and structured data transfer protocols of AAL 1 CS, described in Section 3.7.1, are used. The following services have been defined:

- *Structured DS1/E1 N × 64 Kbps (fractional DS1/E1)*
- *Unstructured DS1/E1 (1.544 Mbps/2.048 Mbps)*
- *Unstructured DS3/E3 (44.736 Mbps/34.368 Mbps)*
- *Structured J2 N × 64 Kbps (fractional J2)*
- *Unstructured J2 (6.312 Mbps)*

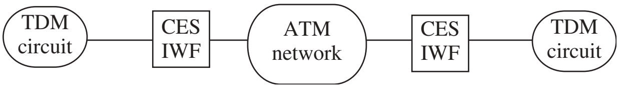


Figure 12.9 Reference model for CES.

These services are named according to whether they use the unstructured or structured data transfer protocol of the AAL 1 CS. Only the first two services are described in this section.

The reference model is shown in Figure 12.9. The two CES IWFs are connected by an ATM connection using the CBR service category. Each CES IWF is connected to a TDM circuit such as T1, T3, E1, E3, and J2. The two IWFs extend transparently the TDM circuit across the ATM network.

12.3.1 Structured DS1/E1/J2 *N* x 64 Kbps Service

This service is intended to emulate point-to-point fractional DS1, E1 and J2 circuits, where N takes the values: $1 \leq N \leq 24$ for DS1; $1 \leq N \leq 31$ for E1; and $1 \leq N \leq 96$ for J2.

CES must maintain 125-μsec frame integrity across the ATM connection. For example, given a 2 × 64 Kbps circuit, the 2 bytes that are sent to the input of the IWF in each frame have to be delivered at the output of the egress IWF in one frame and in the same order. The structured DS1/E1/J2 $N \times 64$ Kbps service also requires synchronous circuit timing. The IWF service interface provides 1.544 MHz timing to external DS1 equipment, 2.048 MHz timing to external E1 equipment, and 6.312 MHz timing to external J2 equipment.

The structured DS1/E1/J2 $N \times 64$ Kbps service can support signaling in two modes: with CAS and without CAS. With the former, the CES IWF has to be able to recognize and transport to the egress IWF the ABCD signaling bits. In the non-CAS mode, also known as *basic mode*, there is no support for CAS. This basic mode can be used for applications not requiring signaling or those that provide signaling using CCS (as used in N-ISDN).

The AAL 1 CS structured data transfer protocol is used to transport fractional DS1/E1/J2. This protocol is described in detail in Section 3.7.1. A structured block is created by collecting N bytes, one from each of the N time slots, and grouping them in sequence. The SDT pointer points to the beginning of such a block. A significant source of delay is due to the *packetization delay* – that is, the amount of time it takes to collect enough data to fill a cell. This can be reduced by sending partially filled cells. This reduction in the delay is at the expense of higher cell rate. Partial fill is an optional feature of the CES IWF. The number of bytes that are sent in each cell can be set at the time the connection is established. When padding is used, the SDT pointer applies to both payload and padding.

If a cell loss is detected at an egress IWF, a dummy cell is inserted. The content of the inserted cell is implementation dependent. If too many cells have been lost, the AAL 1 receiver will locate the next AAL 1 STD pointer to re-acquire framing.

12.3.2 DS1/E1/J2 Unstructured Service

A large number of applications use DS1/E1/J2 interfaces that make use of the entire bandwidth of the TDM circuit. The unstructured service emulates a point-to-point DS1/E1/J2 circuit across an ATM network. The incoming bits from the DS1 circuit are simply placed sequentially into the payload of the AAL 1, without regard to framing, using the unstructured data transfer protocol (see Section 3.7.1).

12.4 THE ATM TRUNKING USING AAL 2 FOR NARROWBAND SERVICES SPECIFICATION

The ATM trunking using AAL 2 for narrowband services specification was designed to interconnect two distant public or private telephone networks over an ATM network. It can be used, for instance, to interconnect a distant PBX and a central office over an ATM network, such as PBX A and the central office in Figure 12.10. It can be also used to connect two distant PBXs over an ATM network, such as PBX B and C in Figure 12.10. A PBX or a central office is connected to an IWF over a T1/E1 link. This specification is used in cellular telephony to transport multiple voice calls.

The protocol stack of an IWF that supports the *ATM trunking using AAL 2 for narrowband service* specification is shown in Figure 12.11. As can be seen, an IWF can transport

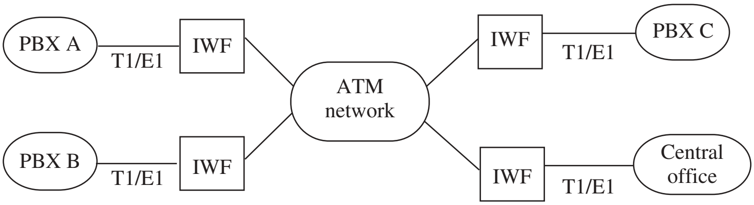


Figure 12.10 ATM trunking using AAL 2 for narrowband services.

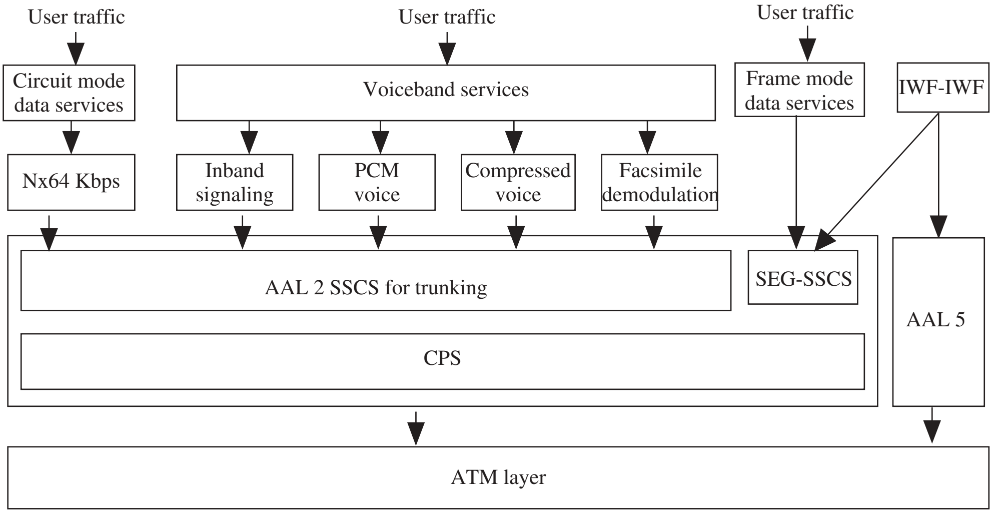


Figure 12.11 The protocol stack of an IWF.

PCM voice (i.e., 64-Kbps voice), compressed voice, facsimile, in-band signaling, and circuit mode data (i.e., fractional T1/E1). In addition, it can transport frame mode data and signaling between two IWFs.

The *ATM trunking using AAL 2 for narrowband service* specification uses the services of two AAL 2 service-specific convergence sublayers developed explicitly for voice trunking: the *AAL 2 SSCS for trunking* and the *segmentation and reassembly service-specific convergence sublayer for AAL 2 (SEG-SSCS).* The AAL SSCS for trunking is described in detail below in Section 12.5. The purpose of this sublayer is to convey telephone voice calls, voiceband data, such as facsimile and data transmitted over a modem, and fractional T1/E1 data. SEG-SSCS for AAL 2 is described in detail in Section 12.6. Using this sublayer, it is possible to transport a packet with a size bigger than the maximum length of 45 bytes permitted in the payload of the CPS packet.

12.4.1 Switched and Non-Switched Trunking

The *ATM trunking using AAL 2 for narrowband services* specification covers both *switched trunking* and *non-switched trunking*. Switched trunking involves analysis of the signaling that accompanies an incoming narrowband call and routing of the user data to an AAL 2 channel that runs over an ATM connection. Similar analysis is required for the incoming calls from an ATM network. There is no permanent correspondence between a TDM time slot to an AAL 2 channel and ATM connection. A new call on the same TDM time slot can be switched to a different AAL 2 connection and ATM connection.

In non-switched trunking, the information stream of a narrowband channel is always carried on the same AAL 2 channel within the same ATM connection. That is, there is a permanent correspondence between a narrowband call and an AAL 2 channel over a specific ATM connection. Non-switched trunking involves no termination of signaling and no routing of narrowband calls in the IWFs.

12.4.2 IWF Functionality for Switched Trunking

The narrowband signaling associated with the individual 64-Kbps channels, whether CAS or CCS, is extracted and forwarded to the *signaling termination and call handling* function of the IWF. The destination egress IWF is determined based on the calling party address. In CAS the calling party address is signaled using the *dual-tone multi-frequency (DTMF)* system. In CCS, it is signaled in the setup message. CCS information is forwarded to the destination IWF using the SAAL, or using *AAL 2 SEG-SSCS (SSADT)* described below in Section 12.6. CAS bits are forwarded to the destination IWF using the special CAS bit packet defined in Section 12.5.

The signaling termination and call handling function also communicates with the CID management function of AAL 2 to assign a CID value for a new call. This CID value represents an AAL 2 channel over some ATM connection. If there is not enough bandwidth over the existing ATM connection(s) to the destination IWF, a new ATM connection is created using the ATM signaling procedures.

12.4.3 IWF Functionality for Non-switched Trunking

The IWF does not interpret, or respond, or process incoming signals. Instead, signaling are passed transparently between the narrowband side and the ATM side. Time slots on the incoming T1/E1 link are permanently mapped to AAL 2 channels. CCS information is carried either over SEG-SSCS (SSTED) or AAL 5. ABCD CAS bits and DTMF information are mapped into the same AAL 2 channel as the user information using the special CAS bit packet defined in the following section.

12.5 THE AAL 2 SERVICE-SPECIFIC CONVERGENCE SUBLAYER (SSCS) FOR TRUNKING

The purpose of this convergence sublayer is to convey voice calls, voiceband data, such as facsimile and data transmitted over a modem, and fractional T1/E1 data.

The reference model of the AAL 2 SSCS for trunking is shown in Figure 12.12. This model depicts a single connection. On either side of the connection, there is a transmitting and a receiving SSCS. For each transmitting and receiving SSCS, there is a signal processing device, which passes and receives information to and from the SSCS. This device is called a *User* and is indicated by a capital *U* to distinguish it from a user, i.e., a customer, who generates the traffic carried over this AAL 2 connection.

At the transmitting side, the User provides a number of functions, such as encoding of voice signals, extraction of dialed digits from multi-frequency tones, and extraction of the ABCD CAS bits. On the receiving side, it decodes voice, generates multi-frequency tones from the received dialed digits, and regenerates the ABCD CAS bits.

The SSCS runs on top of CPS. At the transmitting side, SSCS uses a number of different CPS-packet formats to transport the data received from its User. The CPS-packets are passed on to CPS and are eventually delivered to the receiving SSCS, from where the data is extracted and passed on to its User.

12.5.1 User Functions

The following are some of the functions provided by the User:

Audio

At the transmitter's side, it encodes audio samples using one of several audio algorithms. The transmitting User also detects silence periods and sends silence insertion descriptors. At the receiver's side, it decodes audio bits into a sequence of audio samples, including comfort noise generation as directed by silence insertion descriptors.

Various encoding algorithms can be used. Each algorithm creates encodings, which are grouped together into a packet referred to as the *encoding data unit (EDU)*. Bigger packets can be formed by concatenating several EDUs. The User passes the EDUs to the SSCS, which transmits them to the destination SSCS, which in turn forwards them

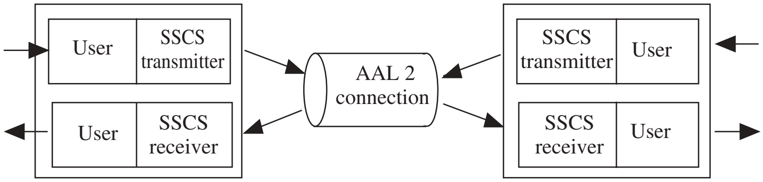


Figure 12.12 The reference model for the AAL 2 SSCS for trunking.

to the destination User, which is responsible for decoding them into a sequence of audio samples. The following are some of the ITU-T audio algorithms.

- *G.711 pulse code modulation (PCM)*: Produces one 8-bit value every 125 μsec, representing the sign and amplitude of an audio sample. Two encoding laws – the A-law and the μ-law – can be used. It normally transmits at 64 Kbps, but it can also transmit at 56 Kbps and 48 Kbps. The G.711 output is accumulated over 1 msec to form an EDU of 8 encoded values.
- *G.722 sub-band adaptive pulse code modulation (SB-ADPCM)*: Produces one 8-bit value every 125 μsec, and represents audio samples with higher fidelity than G.711 PCM. The EDU consists of eight encoded values (i.e. values collected over 1 msec).
- *G.723.1*: Operates at either 5.3 or 6.4 Kbps. Both rates are a mandatory part of the encoder and decoder. Every 30 ms, it emits either 160 or 192 bits, respectively; this characterizes a voice sample. It is possible to switch between the two rates at any 30 msec boundary.
- *G.726 adaptive pulse code modulation (ADPCM)*: Supports bit rates of 40, 32, 24, and 16 Kbps. Every 125 μsec, the encoding produces 5, 4, 3, or 2 bits, respectively.
- *G.722 embedded adaptive pulse code modulation (EADPCM)*: This is a family of variable bit rate coding algorithms with the capability of bit dropping outside the encoder and decoder blocks. It produces code words which contain *enhancement bits* and *core bits*. The enhancement bits can be discarded during network congestion. The number of code bits must remain the same to avoid mistracking of the adaptation state between transmitter and receiver. Algorithms of the G.727 family are referred to by the pair (x, y), where *x* is the number of core plus enhancement bits and *y* is the number of core bits. Recommendation G.727 provides coding rates of 40, 32, 24, and 16 Kbps, with core rates of 16, 24, and 32 Kbps. This corresponds to the following (x, y) pairs: (5, 2), (4, 2), (3, 2), (2, 2), (5, 3), (4, 3), (3, 3), (5, 4), and (4, 4). The data unit format requires that G.727 outputs be accumulated over an interval of 1 msec to yield a sequence of eight encoded values.
- *G.728 low delay code excited linear prediction (LD-CELP)*: This coder produces a group of four codewords every 2.5 msec. Each group of codewords is referred to as an *adaptation cycle* or *frame*.
- *G.729*: Runs at 8 Kbps. Every 10 msec, it emits 80 bits that encode a voice frame.

Multi-frequency tones and CAS bits

At the transmitter's side, it detects and extracts dialed digits codes from multi-frequency tones, such as DTMF. It also extracts the ABCD CAS bits. At the receiver's side, it regenerates the multi-frequency tones from the received dialed digit codes and regenerates the ABCD CAS bits.

Facsimile

At the transmitter's side, when it detects a facsimile, it demodulates the facsimile signal and sends the demodulated original image data and associated control signals to the transmitting SSCS. At the receiver's side, it receives the image data and control signals from the receiving SSCS, it remodulates them into voiceband for transmission to the peer facsimile terminal. This demodulation/remodulation procedure provides a higher-fidelity transfer.

Circuit mode data

At the receiver's side, it passes through circuit mode data, such as $N \times 64$ Kbps fractional T1/E1, and at the receiver's side it regenerates the circuit mode data.

Data frames

At the transmitter's side, it extracts payloads from data frames, and removes flags, bit stuffing and CRC. At the receiver's side, it regenerates data frames and restores flags, bit stuffing, and CRC.

12.5.2 The Service-Specific Convergence Sublayer

Voice is real-time traffic, that has to be delivered to the destination with minimum jitter. Annoying distortions can result due to jitter variability and brief silence periods can be shortened or lengthened. Modem traffic also has to be delivered with minimum jitter, because abnormal phase shifts can be sensed if the delay varies. An SSCS transmitter passes information from its User to CPS with no delay variation. However, cell delay variation can be introduced at the CPS transmitter during periods of time when voice from too many AAL 2 connections is directed simultaneously onto the same ATM connection. Cell delay variation can be controlled by CAC and by requesting the user to switch to an algorithm with greater compression. As in the case of the AAL 1 convergence sublayer, the receiving SSCS introduces a delay before it delivers the information to the receiving User in order to cancel any delay variations incurred by the network.

Type 1 and Type 3 packets

A transmitting SSCS passes data to CPS in the form of a packet known as CPS-packet (see Section 3.7.2). The structure of the CPS-packet is shown in Figure 3.20. It consists of a 3-byte header and a payload which has a maximum length of 45 bytes. In the AAL 2 SSCS for trunking, the CPS-packet payload has been further defined. Specifically, it can be either a Type 1 packet (unprotected) or a Type 3 packet (fully protected). Type 2 packets are to be defined.

In Type 1 packet, the CPS-packet payload is simply made up of data without any additional information that can be used for error detection, such as CRC or parity check. The maximum payload is 45 bytes.

The format of the Type 3 packet is shown in Figure 12.13. The maximum payload is 43 bytes, and the remaining 2 bytes are used for the fields: message type and CRC. The message type is a 6-bit field and it contains a code to indicate the contents of the payload. Message type codes have been defined for dialed digits, ABCD CAS bits, facsimile demodulation control data, alarms, and user state control operations. The CRC-10 is a 10-bit field that contains the FCS computed using the polynomial $x^{10} + x^9 + x^5 + x^4 + x + 1$.

payload	Message type	CRC-10

Figure 12.13 Type 3 packet (fully protected).

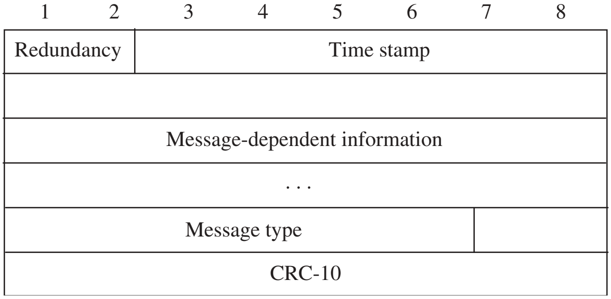


Figure 12.14 Common facilities for Type 3 packets.

In certain cases, SSCS makes use of a more detailed format of the Type 3 packet, referred to as the *common facilities* for Type 3 packets (see Figure 12.14). It contains information which depends upon the type of the message and the fields: redundancy, time stamp, message type and CRC-10. The message type and the CRC-10 fields are the same as in the Type 3 packet. The common facilities for Type 3 packet is triple redundant. That is, it is transmitted three times. The interval between two successive transmissions depends on the type of message. It is 5 msec for dialed digits and ABCD CAS bits; 20 msec for facsimile demodulation control packets and user state control packets. Each copy of the redundant packet contains the same content, except in the 2-bit redundancy field which is populated with the values 1, 2, and 3 corresponding to the packet's first, second, and third retransmission. The time stamp field is a 14-bit field and it contains a time stamp.

SSCS packets

A variety of different packets have been defined to transport the different types of data streams supported by the AAL 2 SSCS for trunking. Below, we examine some of these packets:

- *Audio packet*: The Type 1 packet is used, and the payload contains one or more EDUs. For instance, when the G.711 PCM 64-Kbps algorithm is used, five EDUs are transmitted in a payload, making a total of 40 bytes. Also, an audio packet is accompanied by a sequence number that is carried in the UUI field of the CPS-packet header.
- *Generic silence insertion description (SID) packet*: G.711, G.722, G.726, G.727, G.728 do not contain provisions for voice activity detection, discontinuous transmission, and comfort noise generation tailored to a specific algorithm. In view of this, a generic SID has been developed. Specifically, a generic SID packet is sent immediately after the last active voice packet of a talkspurt. It marks the beginning of the silence and alerts the receiver to expect an absence of active voice packets. The SID can also be sent at arbitrary times during a silence period. The silence period is terminated when the receiver receives an active voice packet. The SID packet is carried in a Type 1 packet.
- *Circuit-mode data at $N \times 64$ Kbps packet*: The Type 1 packet is used. It consists of $M \times N$ bytes, where M is the number of multiples of N bytes that will be packed

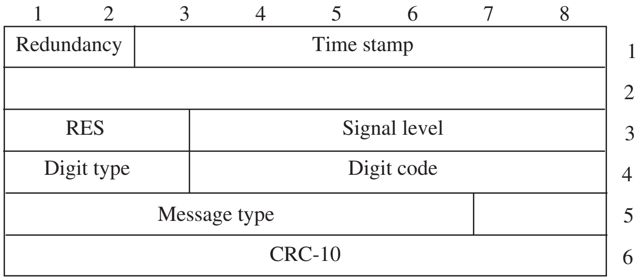


Figure 12.15 Dialed digits packet.

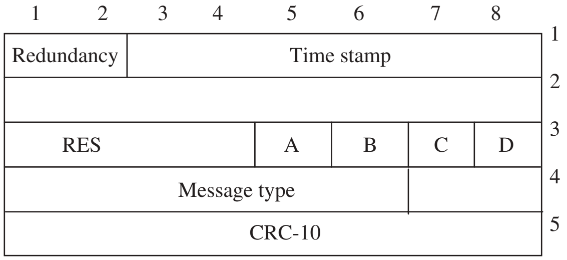


Figure 12.16 CAS bits packet.

together into the same payload, and N is the number of time slots in the $N \times 64$ Kbps stream. Sequence numbers are also used for circuit mode data.

- *Dialed digits packet*: The common facilities for Type 3 packet is used to transport multi-frequency tones for the DTMF signaling system, and is also used for the *signaling systems R1 (MF-R1)* and *R2 (NF-R2)* across an AAL 2 connection. Dialed digits can be used to convey the destination address, either during the call setup or in the middle of a call. The format of the dialed digits packet is shown in Figure 12.15. The message-dependent information consists of a 5-bit signal level field, a 3-bit digit type field, a 5-bit digit code field, and a 3-bit *reserved field (RES)*. The signal level field gives the total power level. The digit type field indicates the type of multi-frequency tone used, such as DTMF, MF-R1, and MF-R2. Finally, the digit code indicates the actual character transmitted (such as a 0 to 9 digit) and special characters.
- *CAS bits packet*: The common facility for Type 3 packets is used to transport the ABCD CAS bits. The packet format is given in Figure 12.16.

12.6 THE SEGMENTATION AND REASSEMBLY SSCS FOR AAL 2 (SEG-SSCS)

This standard specifies a convergence sublayer for the transport of packets with a size bigger than the maximum length of 45 bytes permitted in the payload of the CPS packet.

SEG-SSCS is subdivided into the following sublayers: *service-specific segmentation and reassembly (SSSAR), service-specific transmission error detection (SSTED)*, and *service-specific assured data transfer (SSADT)*. These sublayers are shown in Figure 12.17.

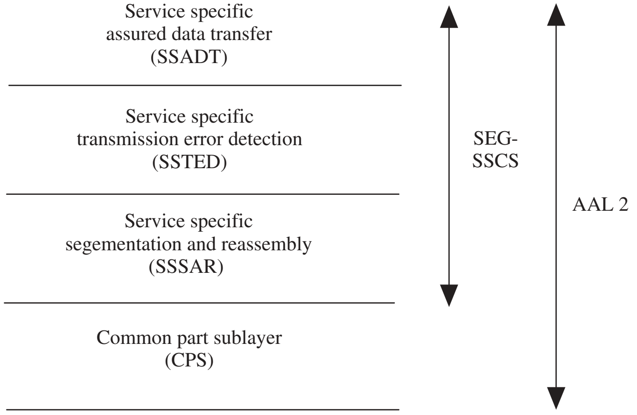


Figure 12.17 The SEG-SSCSC layers.

SSSAR is concerned with the transfer of PDUs which are larger than the maximum 45-byte payload permitted in the CPS-packet. SSTED is used to detect corrupted PDUs, and SSADT is used to recover corrupted PDUs by retransmission. The minimum service provided by SEG-SSCS is the SSSAR function. The other two sublayers are optional.

12.6.1 SSSAR

It transfers PDUs over CPS which can be up to 65,568 bytes. In this sublayer, there is no mechanism to detect corrupted PDUs or to recover corrupted PDUs by retransmission; that is, the transfers are *non-assured*. The receiving SSSAR can detect some errors during the reassembly of a PDU, such as when the received PDU exceeds a pre-defined maximum length. PDUs discovered to be in error are discarded. SSSAR cannot detect partial or complete loss of a PDU, bit errors, and data merged together from separate PDUs. Hence, it might deliver corrupted data. Also, SSSAR does not have any mechanism to correct lost or corrupted PDUs by retransmission.

SSSAR uses the CPS-packet format shown in Figure 3.20. When the SSAR sublayer receives a PDU, it segments it into a number of segments and submits them to CPS. At the SSAR receiver, the CPS-packet payloads are reassembled into the original PDU. SSSAR uses the UUI field in the CPS-packet header to implement “more data.” Specifically, UUI = 27 means more data is required to complete the reassembly of a PDU at the receiving SSSAR. Any other value between 0 and 26 indicates the receipt of the final data for a PDU.

12.6.2 SSTED

SSTED detects corrupted PDUs which it discards. It uses an AAL 5 type of trailer (see Figure 12.18). The trailer consists of the following fields.

- *SSTED-UUI*: A 1-byte field used to transfer transparently user-to-user information.
- *Reserved*: A 6-bit reserved field.

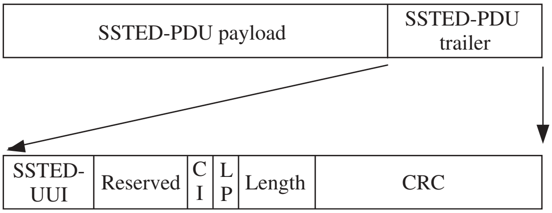


Figure 12.18 The SSTED-PDU trailer.

- *Congestion indicator (CI)*: A 1-bit field provided for compatibility with the service of the CPCS of the AAL 5. It is transparently transported from user to user of the transmitter to the user of the receiver.
- *Loss priority (LP)*: A 1-bit field provided for compatibility with the service of the CPCS of the AAL 5. It is transparently transported from user to user of the transmitter to the user of the receiver.
- *Length*: A 2-byte field that gives the length of the SSTED-PDU payload field.
- *CRC*: The following polynomial is used:
  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
  If the CRC fails, then the management network is informed and the PDU is discarded.

12.6.3 SSADT

SSADT uses sequence numbers to detect missing PDUs that have been either lost or discarded by a lower SEG-SSCS layer because they were corrupted. Missing PDUs are recovered by selective retransmission. A flow control mechanism allows an SSADT receiver to control the rate at which the peer SSADT transmitter sends information.

12.7 VOICE OVER MPLS (VoMPLS)

MPLS can provide QoS on a per-connection basis (as in ATM), and therefore is a suitable technology for voice over packet. The MPLS and Frame Relay Alliance have defined so far the following two specifications, known as *implementation agreements*:

- *TDM Transport over MPLS using AAL 1*
- *I.366.2 Voice Trunking Format over MPLS*

The first implementation agreement defines a service which emulates a point-to-point TDM circuit, such as fractional DS1/E1 ($n \times 64$ Kbps), T1, E1, T3, and E3, over MPLS. It assumes that the TDM traffic to be carried over MPLS is already encapsulated in AAL 1 SAR-PDUs (see Section 3.7.1), and it simply provides an efficient transport of the SAR-PDUs over an LSP.

The second implementation agreement was defined to convey voice calls, voiceband data, such as facsimile and data transmitted over a modem, and fractional T1/E1 circuit-mode data, and is similar to the voice over ATM specification described in Section 12.4. It assumes that the information to be carried over MPLS is the output of the AAL Type 2 SSCS for trunking, described above in Section 12.5. This convergence sublayer was defined in ITU-T I.366.2 recommendation, which explains the name of this implementation agreement. The voice trunking information formatted per the ITU-T I.366.2 recommendation, is subsequently encapsulated in CPS-packets, described in Section 3.7.2. The CPS-packets are not transported over CPS-PDUs, but they are transported directly over MPLS. This implementation agreement describes only how the CPS-packets are transported over MPLS, and in view of this, it is commonly referred to as *AAL 2 over MPLS (A2oMPLS)*.

One method for implementing VoIP is to transport the IP packets containing the voice samples over MPLS. In this case, the voice samples are first encapsulated in RTP, UDP, and IP and then in MPLS. Compressed headers are used in some implementations. The encapsulated information is then conveyed by an MPLS transport arrangement, such as frame relay, ATM, PoS, and Ethernet. A2oMPLS by-passes the RTP/UDP/IP encapsulation, and therefore it provides a more efficient mechanism for voice over MPLS.

We now proceed to describe the two implementation agreements in detail.

12.8 TDM TRANSPORT OVER MPLS USING AAL 1

The reference architecture for the TDM transport over MPLS using AAL 1, or *TDM-MPLS* for sort, is shown in Figure 12.19. Each TDM device is connected to a *provider edge (PE)* device over a PDH TDM link, such as T1, E1, T3, or E3. The PE is equivalent to the CES IWF (see Section 12.3). It is connected to the destination PE over an MPLS network via a point-to-point bidirectional LSP, which has been created either by manual provisioning or by using an MPLS signaling protocol, such as CR-LDP or RSVP-TE.

The PE provides multiple functions, including:

- Transport of fractional T1/E1 (i.e. $n \times 64$ Kbps) or of an entire signal (i.e. of a T1, E1, T3, or E3) over an LSP.
- End-to-end preservation of the order of the TDM frames.
- Transparent transfer of CAS bits.
- A mechanism for the reconstruction of the TDM clocks.
- Transport of standard alarms between the two TDM devices.

The TDM traffic transmitted to a PE from the TDM device is first encapsulated using AAL 1. Recall from Section 3.7.1 that AAL 1 consists of a SAR sublayer and a *convergence sublayer (CS)*. The CS performs a variety of functions, such as handling of the cell delay variation, processing of the sequence count, structured and unstructured data transfers, and transfer of timing information. The SAR sublayer is responsible for the bit error detection, and possibly correction, of blocks of data received from CS. It accepts

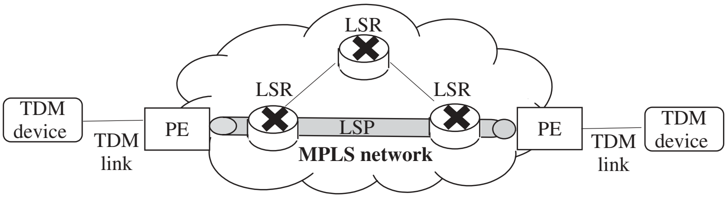


Figure 12.19 The TDM-MPLS reference architecture.

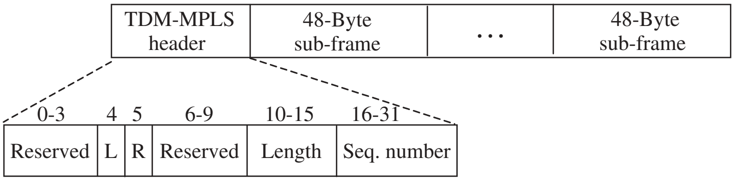


Figure 12.20 The TDM-MPLS frame.

blocks of 47 bytes from the convergence sublayer, and adds a 1-byte header to form the SAR-PDU. The resulting 48-byte SAR-PDUs are then transported over MPLS to the destination PE, which extracts the TDM traffic and transmits it to its TDM device over the TDM link. The same occurs in the opposite direction.

Unlike in ATM where each SAR-PDU is transported in a single ATM cell, many SAR-PDUs can be transported together over MPLS in a single frame. The format of this frame, referred to as the *TDM-MPLS* frame, is shown in Figure 12.20. As we can see, it consists of a TDM-MPLS header and multiple SAR-PDUs referred to as *48-byte subframes*. The TDM-MPLS frame is further encapsulated with a label stack if the underlying network of the MPLS is packet over SONET (PoS) or Ethernet.

The following fields have been defined in the TDM-MPLS header:

- *L bit*: This is the fourth bit in the header and is used to indicate physical layer loss of signal.
- *R bit*: Occupies the fifth bit in the header and is used to indicate that the source is not receiving packets at its TDM-MPLS receive port.
- *Length*: This is a 6-bit field that indicates the length of the TDM-MPLS frame (header and payload) in case padding is employed to meet minimum transmission unit requirements of layer 2 of the MPLS network. It must be used if the TDM-MPLS frame length plus the layer 2 overhead is less than 64 bytes, and it must be 0 if this length exceeds 64 bytes.
- *Sequence number*: The 16 bits sequence number is used to guarantee ordered frame delivery.

The payload of a TDM-MPLS frame consists of one to thirty subframes. As mentioned above, each subframe is a 48-byte SAR-PDU. The number of subframes in the payload can be inferred by the receiving side from the length indicated in the TDM-MPLS header. It is pre-configured and is typically chosen to trade-off the delay to fill in a TDM-MPLS frame against overheads. For instance, using one subframe per TDM-MPLS frame reduces delay to minimum, but incurs the highest overhead. Using eight subframes per TDM-MPLS frame reduces the overhead, but increases the delay by a factor of eight.

Each subframe is a 48-byte SAR-PDU, of which 47 bytes comes from the AAL 1 convergence sublayer. The structure of these 47 bytes follows the AAL 1 definition. That is, it could be an unstructured data transfer or a structured data transfer (see Section 3.7.1). It could also be a *structured data transfer with CAS*, which is a structured data transfer with additional provisioning to carry the CAS bits.

TDM-MPLS assumes that the QoS guarantee is provided by the MPLS network. Specifically, it is assumed that sufficient bandwidth has been allocated to the LSP carrying the

TDM-MPLS frames, so that to provide a low end-to-end transfer delay and a low packet loss probability.

12.9 I.366.2 VOICE TRUNKING FORMAT OVER MPLS

As we have seen, AAL 2 can be used to multiplex many voice calls over the same ATM connection. To that effect, the AAL 2 SSCS for trunking described in Section 12.5 is needed in order to convert the voice traffic and signals into packets at the transmitter, and extract the voice traffic and signals from the packets at the receiver. These packets are in the form of CPS-packets, which are transmitted over ATM. This implementation agreement, assumes the presence of an AAL 2 service-specific convergence sublayer for trunking, but instead of carrying the CPS-packets over ATM, they are carried over MPLS. The implementation agreement, therefore, is only concerned with the transport of AAL 2 CPS-packets over MPLS, and in view of this, it is commonly referred to as *AAL 2 over MPLS (A2oMPLS)*.

The reference architecture for this implementation agreement is shown in Figure 12.21. The A2oMPLS functionality is implemented in a *gateway (GW)*, which can be a line card in a device that implements the AAL 2 SSCS for trunking. The device is attached to one or more LSRs, and the gateways are interconnected over the MPLS network via bidirectional point-to-point LSPs.

In the AAL 2 CPS, the CPS-packets are packed into CPS-PDUs, and each CPS-PDU is carried in a separate ATM cell. In the A2oMPLS architecture, multiple CPS-packets can be placed onto the same frame, known as the A2oMPLS frame, and transported over an LSP. The structure of the A2oMPLS frame is shown in Figure 12.22. The following fields have been defined:

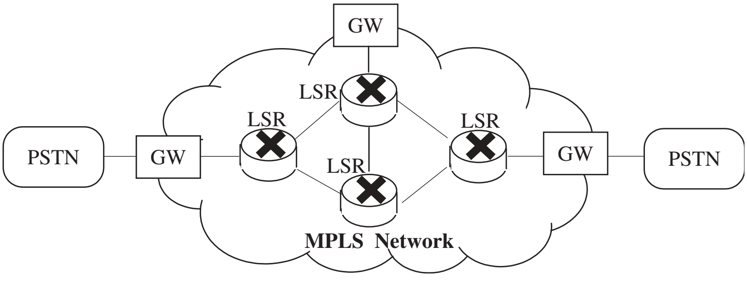


Figure 12.21 The A2oMPLS reference architecture.

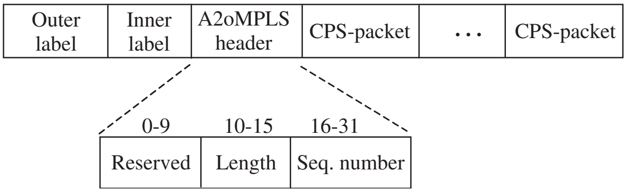


Figure 12.22 The A2oMPLS frame structure.

- *Outer label*: This is the MPLS encapsulation that is required if the underlying layer 2 network is PoS or Ethernet.
- *Inner label*: This is an optional MPLS label encapsulation that is used to increase the number of multiplexed voice calls onto the same LSP. This label is only meaningful to the transmitting and receiving GWs, and is not used by the LSRs in the MPLS network.
- *A2oMPLS header*: The header consists of a *reserved* field (bits 0 to 9), a *length* field (bits 10 to 15), and a *sequence number* field (bits 16–31). The length field is used in the same way as in the TDM-MPLS header described above. It is used to indicate the length of the A2oMPLS frame (header and payload) in case padding is employed to meet minimum transmission unit requirements of layer 2 of the MPLS network. It must be used if the A2oMPLS frame length plus the layer 2 overhead is less than 64 bytes, and it must be 0 if this length exceeds 64 bytes. The 16 bits sequence number can be used to guarantee ordered frame delivery.
- *CPS-packets*: These have the same structure as shown in Figure 3.20.

In A2oMPLS up to 248 different voice calls can be established over the same LSP. The optional inner label can be used in order to increase the total multiplexed voice calls over the same LSP. Recall from Section 3.7.2 that each voice call transported over an AAL 2 connection is associated with a *channel identifier (CID)*. The CID value is carried in each CPS-packet so that it can be associated with the correct voice call. CID values are allocated as in AAL 2 (see Section 3.7.2).

The CPS-packets are carried in a A2oMPLS frame in any arbitrary order. The procedure of populating an A2oMPLS frame is similar to the one in Section 3.7.2 used to populate a CPS-PDU. That is, at the transmitter, CPS-packets are placed in the payload of an A2oMPLS frame until either the frame reaches a maximum size or a timer expires. The frame is then transmitted over the LSP to the receiver A2oMPLS, where the CPS-packets are retrieved from the frame's payload and delivered to the appropriate user.

PROBLEMS

1. What are the main sources for jitter and what is its effect on the QoS for voice over packet?
2. Explain how the QoS of a phone call over a connection-oriented network, such as ATM and MPLS, can be guaranteed.
3. Explain, how an ISP provides QoS for a phone call over an IP network.
4. Explain the difference between switched mode and non-switched mode described in the *ATM trunking using AAL 2 for narrowband services* specification.
5. The AAL 2 SSCS for trunking provides silence removal. Describe how this is done. (Hint: describe the relevant User function and the SSCS SID packet.)
6. Describe the demodulation/remodulation function of the AAL 2 SSCS for trunking used for the transport of facsimile.
7. A T1 signal is transported using the *TDM transport over MPLS using AAL 1* implementation agreement. Calculate the delay to fill in a TDM-MPLS frame using the unstructured data transfer scheme and the percent overhead assuming that n subframes are transported in each TDM-MPLS frame, for $n = 1, 2, \ldots, 31$. The overhead consists of the TDM-MPLS overhead and a 30-bit MPLS label encapsulation. The percent overhead is the total number of transmitted overhead bytes divided by the entire length of the TDM-MPLS frame plus the MPLS label encapsulation.

Bibliography

Most of the material in this book, except for Chapter 10 which deals with the new OBS architecture, comes from existing standards. In this section, we give a bibliography of relevant standards and other references organized per topic. Browsing the Internet for further information is also encouraged.

SONET/SDH AND GFP (CHAPTER 2)

The SONET/SDH specifications can be found in References 1 and 2. References 3 and 4 describe how PPP frames can be transmitted over SONET/SDH. Finally, References 5 and 6 describe the generic framing procedure and the link capacity assignment scheme.

1. *Synchronous Optical Network (SONET): Physical Interface Specifications*, ANSI T1.105.06, 2000.
2. *Network Node Interface for the Synchronous Digital Hierarchy (SDH)*, ITU-T Recommendation G.707, October 2000.
3. *PPP over SONET/SDH,* IETF RFC 2615, June 1999.
4. *PPP in HDLC-Like Framing*, RFC 1662, July 1994.
5. *The Generic Framing Procedure (GFP)*, ITU-T Recommendation G.7041, October 2001.
6. *Link Capacity Adjustment Scheme (LCAS) for Virtual Concatenated Signals*, ITU-T Recommendation G.7042, October 2001.

ATM NETWORKS (CHAPTERS 3, 4, AND 5)

The specifications for the ATM architecture and its adaptation layers are described in References 7 to 11. The classical IP and ARP over ATM specification is described in Reference 12. References 13 and 14 describe how multicasting can be implemented over ATM. This topic is not covered in this book, but it can be found in my earlier book *An Introduction to ATM*. References 15 and 16 give the specifications for congestion control in an ATM network, and References 17 and 18 describe the signaling protocol for setting up a point-to-point connection over the UNI. Finally, References 19 and 20 describe the signaling protocol for setting up a point-to-multipoint connection over the UNI and the

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2

PNNI, respectively. The last two topics are not covered in this book, but they can be found in my above-mentioned book on ATM networks.

7. *B-ISDN General Network Aspects*, ITU-T Recommendation I.311, March 1993.
8. *B-ISDN ATM Layer Specification*, ITU-T Recommendation I.361, February 1999.
9. *Broadband ISDN – ATM Adaptation Layer for Constant Bit Rate Services Functionality and Specification*, ANSI, T1/S1 92–605, November 1992.
10. *B-ISDN ATM Adaptation Layer Type 2 Specification*, ITU-T Recommendation I.362.2, November 1996.
11. *B-ISDN ATM Adaptation Layer (AAL) Specification*, ITU-T Recommendation I.363, March 1993.
12. *Classical IP and ARP Over ATM*, IETF RFC 2225, April 1998.
13. *Support for Multicast Over UNI 3.0/3.1 Based ATM Networks*, IETF, RFC 2022.
14. *Multicast Server Architectures for MARS-Based ATM Multicasting*, IETF, RFC 2149.
15. *Traffic Management Specification Version 4.1*, ATM Forum, March 1999.
16. *Addendum to Traffic Management V4.1 for an Optional Minimum Desired Cell Rate Indication for UBR*, ATM Forum, July 2000.
17. *ATM User-Network Interface (UNI) Signalling Specification, Version 4.0*, ATM Forum, July 1996.
18. *Broadband Integrated Services Digital Network (B-ISDN) – Digital Subscriber Signalling System No 2 (DSS 2) – User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control*, ITU-T Recommendation Q.2931, 1995.
19. *Broadband Integrated Services Digital Network (B-ISDN) – Digital Subscriber Signalling System No 2 (DSS 2) – User-Network Interface (UNI) Layer 3 Specification for Point-to-Multipoint Call/Connection Control*, ITU-T Recommendation Q.2971, October 1995.
20. *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, ATM Forum, March 1996

MULTI-PROTOCOL LABEL SWITCHING (CHAPTERS 6 AND 7)

The specifications of the MPLS architecture can be found in References 21 to 25. References 26 to 28 give the specifications for LDP and CR-LDP, and References 29 to 31 give the specifications of RSVP and RSVP-TE. Several other MPLS-related Internet Drafts and RFCs are available in the MPLS Working group of IETF.

21. *Requirements for Traffic Engineering over MPLS*, IETF RFC 2702.
22. *Multiprotocol Label Switching Architecture*, IETF, RFC 3031.
23. *VCID Notification over ATM Link for LDP*, IETF RFC 3038.
24. *Use of Label Switching on Frame Relay Networks*, IETF RFC 3034.
25. *MPLS Support of Differentiated Services*, IETF RFC 3270.
26. *LDP Specification*, IETF, IETF RFC 3036.
27. *Applicability Statement for CR-LDP*, IETF RFC 3213.
28. *Constraint-Based LSP Setup Using LDP*, IETF RFC 3212.
29. *Resource ReSerVation Protocol (RSVP)*, IETF RFC 2205.
30. *Applicability Statement for Extensions to RSVP for LSP-Tunnels*, IETF RFC 3210.
31. *RSVP-TE: Extensions to RSVP for LSP Tunnels*, IETF RFC 3209.

WAVELENGTH ROUTING OPTICAL NETWORKS (CHAPTERS 8 AND 9)

The two books given in References 32 and 33 contain many good articles on optical fibers and components. Reference 34 describes ITU-T's optical transport network, and Reference 35 describe the G.709 transport standard. Reference 36 describes several different control plane architectures for transporting IP traffic over a wavelength routing network, and References 37 to 39 describe the GMPLS architecture. References 40 and 41 describe the CR-LDP and RSVP-TE extensions, respectively. Reference 42 describes the OIF UNI specification. Several other MPLS related Internet Drafts and RFCs can be found in the IETF MPLS and CCAMP Working groups.

32. *Optical Fiber Telecommunications IVA: Systems and Impairments*, Kaminow and Li (Editors), Academic Press 2002.
33. *Optical Fiber Telecommunications IVB: Components*, Kaminow and Li (Editors), Academic Press 2002.
34. *Architecture of the Optical Transport Network*, ITU-T G.872.
35. *Network Node Interfaces for the Optical Transport Network (OTN),* ITU-T G.709.
36. *IP over Optical Networks: A Framework*, IETF RFC 3717.
37. *Generalized Multi-Protocol Label Switching (GMPLS) Architecture*, draft-ietf-ccamp-gmpls-architecture-07.txt
38. *GMPL Extensions for SONET and SDH Control*, draft-ietf-ccamp-gmpls-sonet-sdh-08.txt
39. *Generalized MPLS-Signaling Functional Description*, IETF RFC 3471.
40. *Generalized MPLS-Signaling – CR-LDP Extensions*, IETF RFC 3472.
41. *Generalized MPLS-Signaling – RSVP-TE Extensions*, IETF RFC 3473.
42. *User Network Interface (UNI) 1.0 Signaling Specification*, Optical Internetworking Forum.

OPTICAL BURST SWITCHING (CHAPTER 10)

Reference 43 gives a review of the optical packet switching technique and describes different switch architectures. Reference 44 summarizes the main features of OBS, and References 45 and 46 describe parts of the JumpStart project. OBS is an evolving technology; it is important to keep current with the literature.

43. *A Survey of Optical Packet Switching and Optical Burst Switching Techniques*, L. Xu *et al.*, IEEE Magazine on Communications, Jan 2001, pp 136–142.
44. *An Introduction to Optical Burst Switching*, T. Battistilli and H. Perros, IEEE Optical Communications (part of the IEEE Communications Magazine), August 2003, pp S10–S15.
45. J*umpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks*, I. Baldine *et al.*, IEEE Magazine on Communications, Feb 2002. pp 82–89.
46. *Signalling Support for Multicast and QoS Within the JumpStart WDM Burst Switching Architecture*, I. Baldine *et al.*, Optical Networks Magazine, Vol 4, 2003, pp 68–80.

ACCESS NETWORKS (CHAPTER 11)

References 47 to 51 give the ITU-T specifications of ADSL, ADSL2, and ADSL2+. The DSL Forum documents listed in References 52 to 54 are also interesting to read, as well as other documents that can be found in the DSL Forum's Web site. References 55 to 58 provide background information for the schemes for accessing network service providers (see Section 11.1.4). The data-over cable service interface specifications (DOCSIS), which is used in cable-based access networks, is given in Reference 59. Finally, Reference 60 gives the ITU-T specification for PONs. Access networks are continuously evolving; scan the Internet for the latest developments.

47. *Asymmetric Digital Subscriber Line (ADSL) Transceivers Series G: Transmission Systems and Media, Digital Systems and Networks Digital Sections and Digital Line System – Access Network*, ITU-T Recommendation G.992.1.
48. *Series G: Transmission Systems and Media, Digital Systems and Networks Digital Sections and Digital Line System – Access Networks Splitterless Asymmetric Digital Subscriber Line (ADSL) Transceivers*, ITU-T Recommendation G.992.2.
49. *Series G: Transmission Systems and Media, Digital Systems and Networks; Digital Sections and Digital Line System – Access Networks; Asymmetric Digital Subscriber Line Transceivers 2 (ADSL 2)*, ITU-T Recommendation G.992.3.
50. *Series G: Transmission Systems and Media, Digital Systems and Networks – Digital Sections and Digital Line System – Access Networks – Splitterless Asymmetric Digital Subscriber Line Transceivers 2 (Splitterless ADSL2)*, ITU-T Recommendation G.992.4.
51. *Asymmetric Digital Subscriber Line (ADSL) Transceivers – Extended Bandwidth ADSL2 (ADSL2+) Series G: Transmission Systems and Media, Digital Systems and Networks Digital Sections and Digital Line System – Access Networks*, ITU-T Recommendation G.992.5.
52. *Broadband Service Architecture for Access to Legacy Data Networks over ADSL ("PPP over ATM")*, DSL Forum TR-012, June 1998.
53. *ATM Over ADSL Recommendation*, DSL Forum TR-017, March 1999.
54. *References and Requirements for CPE Architectures for Data Access*, DSL Forum TR-18, May 1999.
55. *PPP Over AAL5*, IETF RFC 2364, July 1998.
56. *Layer Two Tunneling Protocol "L2TP"*, IETF, Internet-Draft, November 2000.
57. *Remote Authentication Dial in User Service (RADIUS)*, IETF, RFC 2865, June 2000.
58. *A Method for Transmitting PPP Over Ethernet (PPPoE)*, IETF, RFC 2516, February 1999.
59. *Data-Over Cable Service Interface Specifications – Radio Frequency Interface Specification*, Cable Television Laboratories, 1999.
60. *Broadband Optical Access Systems Based on Passive Optical Networks (PON)*, ITU-T Recommendation G.983.1, October 1998.

VOICE OVER ATM AND MPLS (CHAPTER 12)

To understand voice over packet, some knowledge of the telephone network is needed. The book given in Reference 61 provides a good description of various signaling protocols used in telephony. Also, the book given in Reference 62 gives a good overview of the various solutions for voice over packet. References 63 to 65 provide the ATM Forum specifications for *circuit emulation services (CES),* and References 66 to 68 provide the ATM Forum specifications for voice over AAL 2. Finally, References 69 to 71 give the

MPLS/Frame Relay Alliance specifications for voice over MPLS. This is an area of active development; check the MPLS/Frame Relay Alliance Web site for the latest specifications.

61. *Signaling in Telecommunication Networks*, J. van Bosse, Wiley 1998.
62. *Voice over Packet Networks*, D.J. Wright, Wiley 2001.
63. *Voice and Telephony over ATM – ATM Trunking Using AAL 1 for Narrowband Services, Version 1.0*, ATM Forum, July 1997.
64. *Circuit Emulation Service Interoperability Specification, Version 2.0*, ATM Forum, January 1997.
65. *Specification of (DBCES) Dynamic Bandwidth Utilization – In 64 Kbps Time Slot Trunking Over ATM – Using CES*, ATM Forum, July 1997.
66. *ATM Trunking Using AAL 2 for Narrowband Services*, ATM Forum, February 1999.
67. *AAL Type 2 Service-Specific Convergence Sublayer for Trunking*, ITU-T Recommendation I.366.2.
68. *Segmentation and Reassembly Service-Specific Convergence Sublayer for the AAL Type 2*, ITU-T Recommendation I.366.1.
69. *Voice over MPLS – Bearer Transport Implementation Agreement*, MPLS Forum 1.0, July 2001.
70. *TDM Transport over MPLS Using AAL 1*, MPLS/Frame Relay Alliance 4.0, June 2003.
71. *I.366.2 Voice Trunking Format over MPLS Implementation Agreement*, MPLS/Frame Relay Alliance 5.0.0, August 2003.

Index

Connection-oriented Networks Harry Perros
© 2005 John Wiley & Sons, Ltd ISBN: 0-470-02163-2