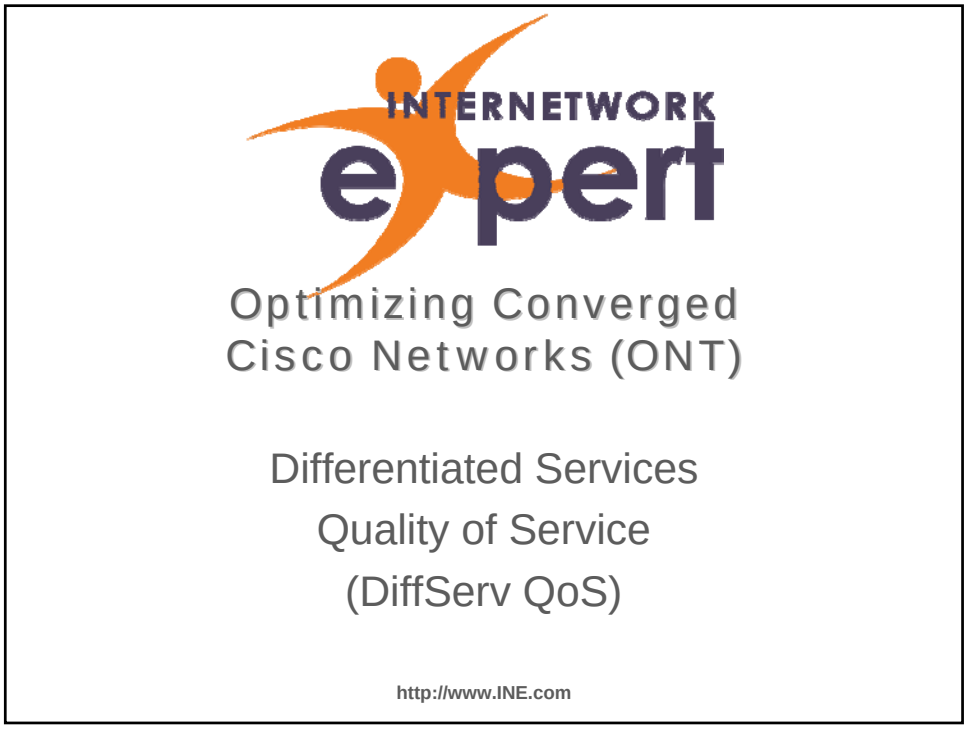



## DiffServ QoS Overview

- Differentiated Services goals
  - Classify traffic into flows
  - Apply QoS policy to flow
- Classification is also known as “marking”
- QoS policy is where features such as queuing, policing, rate limiting, etc. are applied

## Classification Methods

- Layer 2 header fields
- Layer 3 header fields
- Access Lists
- Application Level



## Layer 2 QoS Classification

- Ethernet Class of Service (CoS)
  - Exists in ISL or 802.1Q trunk headers only
  - 4 bit field called USER field in ISL header
  - 3 bit field called 802.1p priority in 802.1Q header
- Frame Relay
  - Discard Eligility (DE) bit
  - If set, more likely to get dropped when congestion occurs
- ATM
  - Cell Loss Priority (CLP)
  - Like DE, if set more likely to get dropped when congestion occurs
- MPLS
  - 3 bits called Experimental Bits (EXP)

## Layer 3 QoS Classification

- IP Type of Service (ToS)
  - 8 bit field in IP header
  - Differentiated Services Code Point (DSCP)
    - First 6 most significant bits
    - RFC defines the PHB's, but technically its up to you to implement
  - IP Precedence
    - First 3 most significant bits
    - Overlaps with DSCP
  - 2 least significant bits unused



## IP Precedence

- Higher value means higher priority
  - 0 (000) - routine
  - 1 (001) - priority
  - 2 (010) - immediate
  - 3 (011) - flash
  - 4 (100) - flash-override
  - 5 (101) - critical
  - 6 (110) - internet
  - 7 (111) - network
- Overlaps with DSCP

## DSCP

- Assured Forwarding
  - Low Drop
    - af11 - 001010
    - af21 - 010010
    - af31 - 011010 (Signaling)
    - af41 - 100010
  - Medium Drop
    - af12 - 001100
    - af22 - 010100
    - af32 - 011100
    - af42 - 100100
  - High Drop
    - af13 - 001110
    - af23 - 010110
    - af33 - 011110
    - af43 - 100110
- Class Selector
  - cs1 - 001000
  - cs2 - 010000
  - cs3 - 011000 (Signaling)
  - cs4 - 100000
  - cs5 - 101000
  - cs6 - 110000
  - cs7 - 111000
- Expedited Forwarding
  - EF - 101110 (VoIP Call)



## Advanced Classification

- Access Lists
  - Protocol
    - e.g. OSPF, ESP, GRE, etc.
  - Source/destination pairs
  - TCP/UDP ports
- Application
  - Network Based Application Recognition (NBAR)
  - **`match protocol`** class-map command



## Classification Trust Boundaries

- In order to implement proper DiffServ, the markings must be accurate
- "Trust boundary" defines connection points where marking should be modified or unmodified
  - e.g. at layer 2 access switch facing end host
- Some applications mark their own traffic (e.g. Cisco IP phone) but to be 100% sure the policy is usually manual



## QoS Pre-Classification

- In cases where tunneling is implemented, PHB is hidden inside tunnel
  - e.g. IPsec or GRE
- IOS QoS pre-classify feature changes the order of operations of QoS to check the PHB before tunnel encapsulation is added
- Needed for applications like VoIP inside of IPsec



## Queueing Overview

- Now that traffic is classified, what do we do with it?
- "Queueing" is the result of the QoS policy that defines our PHB
- Queueing typically means "output queueing", but network devices also have input queues
  - Input queueing can rarely be modified to fix QoS issues
- Other QoS mechanisms exist that are not queueing
  - i.e. all queueing methods are QoS methods, but not all QoS methods are queueing methods



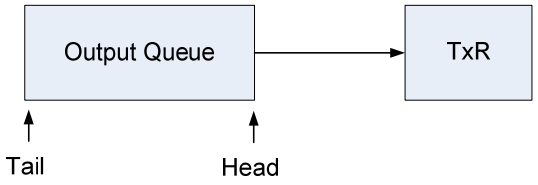
## Output Queueing Components

- Output queueing defines how traffic is scheduled to leave the interface
- Consists of…
  - Software queue
    - Commonly referred to as the output queue
    - Where our PHB is applied
  - Hardware queue
    - Called the Transmit Ring (TxR)
    - Physical interface driver that cannot be modified



## Output Queueing Workflow

- (Software) output queue consists of two portions
  - Tail - where traffic enters
  - Head - where traffic exits
- Traffic enters the tail of the queue, waits until it gets to the head, and then exits to the transmit ring
  - Exception is when output queue is empty
- Time that traffic waits in the output queue is the actual queueing delay






## Modifying the PHB

- To change the behavior of how traffic enters and exits the output queue, we modify the *queueing method*
- The queueing method is also known as *scheduling*, or if very complex, *fancy queueing*

## Queueing Methods

- Common queueing methods
  - First-In-First-Out (FIFO)
  - Weighted Fair Queueing (WFQ)
  - Custom Queueing (CQ)
  - Class Based Weighted Fair Queueing (CBWFQ)
    - AKA *Fancy Queueing*
  - Priority Queuing (PQ)
  - Low Latency Queueing (LLQ)
- Generalized as Congestion Management Techniques
  - If the link isn't congested why change the queueing?



## FIFO Queueing

- Packets are sent in the exact order that they arrive
- Simplest form of scheduling
- TxR is always a FIFO queue

## Weighted Fair Queueing

- Traffic is “weighted” based on IP flow information
  - Higher ToS values get higher weights
  - Lowest bandwidth flows get higher weights than higher bandwidth flows
- Goal is to give fairness to low bandwidth and high bandwidth flows, while at the same time prioritizing important flows
- Good for low speed links



## Custom Queueing

- Manually defined weights control how flows are serviced
- Legacy way of preferring one traffic class over another
- Limited application compared to MQC
  - More on this later…



## Class Based Weighted Fair Queueing

- Like WFQ, but flows are manually defined based on user parameters
- AKA Modular Quality of Service Command Line Interface (MQC)
- MQC used to only support WFQ, hence CBWFQ
- Current technique of offering bandwidth guarantee to mission critical flows



## Priority Queueing

- Special priority queueing gets infinite weight
  - Means that it is always serviced before anything else
- Legacy way of offering low delay
- Limited application because of traffic starvation



## Low Latency Queueing

- Like PQ, but implemented inside MQC
- Gives infinite weight to a traffic class, but also implements a policer to ensure that starvation does not occur
- Current technique of offering low delay service



## Congestion Management Problems

- Congestion *management* techniques wait until congestion event occurs, then deal with it
- Results in a situation known as *tail drop*
  - If output queue is full, traffic is dropped as it tried to get in at the tail
- Tail drop leads to a problem known as Global TCP Synchronization



## Global TCP Synchronization

- TCP uses a built in congestion management technique called *sliding windowing* and *slow start*
- "Window" controls how much traffic can be sent before needing an ACK
  - More reliable and higher bandwidth, larger window
  - As traffic is not dropped, window increases
- When ACK is not received (loss occurred) TCP goes into slow start
  - Drop the window size down
  - Try to build up the window size again
- When one packet is dropped, usually lots are dropped
- Slow start "synchronizes" between flows
  - Periods of high utilization followed by lows



## Congestion Avoidance

- Congestion *avoidance* techniques try to fix this problem before it happens
- Selectively admits or drops packets in the output queue based on weighted thresholds
- Implemented as Weighted Random Early Detection (WRED)

## How WRED Works

- Traffic weights are assigned queue limits
- As depth increases, *drop probability* increases
- If limit is exceeded, probability becomes 100%
- Result is that flows are *selectively* dropped, with higher weighted flows less likely to be dropped
- Individual flows forced to go into slow start one at a time



## Traffic Shaping

- Slows down the output rate to the TxR
- Delays excess traffic for later transmission
- Used in cases where the input speed exceeds the output speed, or in VC based environments
  - Frame Relay DS3 with 20Mbps PVC
  - Metro Ethernet with 5Mbps guarantee



## Traffic Policing

- Like shaping, enforces a limit on bandwidth
  - Unlike shaping, does not queue excess traffic
- Implies that traffic can be policed as *input* or *output*
  - All other QoS mechanisms seen are output only
- AKA *rate limiting*
- Applicable both for QoS and Security



## Control Plane Policing (CoPP)

- Rate limiting applied to the router's control plane itself
  - e.g. the CPU
- Used to protect routers and switches from DoS attacks against themselves
- Goal is to maintain packet forwarding even in the case of attack or heavy processing load

## Serialization Delay Issues

- Queueing methods change the PHB while waiting in the output queue
- Serialization is a physical function of the link
- What happens if a large packet is currently on the TxR being serialized when my priority VoIP packet arrives?



## Serialization Delay Example

- 256Kbps Frame Relay link
  - Serializes at 256000bps
- 1500 bytes serialized at 256Kbps
  - 1500 * 8 / 256000 = 0.046sec = 46ms
- Even if VoIP is in LLQ, worst case delay is 46ms



## Fragmentation and Serialization

- Serialization delay on low speed links can be fixed by limiting the size of the largest packet
  - e.g. the MTU
- Implemented at layer 2 with features such as FRF.12 and Multilink PPP LFI
- Goal is to ensure that largest packet takes no longer than 10ms to serialize
- Result is that worst case VoIP serialization delay is 10ms per link



## Other Link Efficiency Methods

- Header compression
  - TCP header compression
  - cRTP
- Payload compression
  - Predictor
    - Tries to guess next payload contents based on previous
  - Stacker
    - Lemple Ziv based algorithm
- IOS supports both software and hardware based compression

# DiffServ QoS Q&A