Introduction to IP QoS

Primer to IP Quality of Service Aspects
Queuing, Shaping, Classification

Agenda

- **IP QoS Introduction**
- **Queue Management**
- **Congestion Avoidance**
- **Traffic Rate Management**
- **Classification and Marking**

Why QoS is an Issue for IP

- **Originally, IP was designed for best effort service only**
  - simple network (IP), complex end-systems (TCP)
  - no Quality of Service (QoS)
    - Throughput not guaranteed
    - Transmission delay is not bounded
  - great for data but not appropriate for real-time applications
- **Multimedia applications**
  - need transmission of voice and video in real-time
  - in a traditional environment (circuit switching) require data flow at a stable rate and constant delay (synchronous behavior)

Why QoS is an Issue for IP

- **Multimedia applications such as video conferencing systems**
  - need a lot more bandwidth than applications that were used in the early days of the Internet
- **Traditional Internet applications**
  - such as WWW, FTP or TELNET, cannot tolerate packet loss
  - but are less sensitive to variable delays
- **Most real-time applications**
  - show just the opposite behavior
  - they can compensate for a reasonable amount of packet loss
  - but are very critical towards high variable delays
- **This means that without any bandwidth control**
  - the quality of these real-time streams depends on the bandwidth that is available
  - low or unstable bandwidth causes bad quality in real-time transmissions by leading to dropouts and hangs

Why QoS is an Issue for IP

- **Until recently the usual approach to satisfy real time constraints was**
  - Circuit switching paradigm (e.g. ISDN)
    - reservation of a circuit
  - Connection oriented cell-switching (e.g. ATM)
    - reservation of resources for a virtual circuit to achieve requested QoS
- **In order to make the Internet a real success**
  - delay- and bandwidth sensitive traffic should be also delivered over packet-switched, connectionless networks
- **But the Internet is inhomogeneous**
  - QoS is given by the weakest link in the chain between sender and receiver

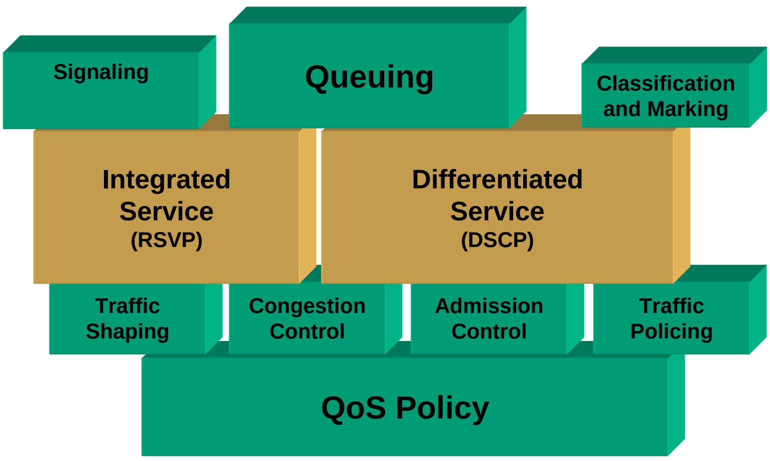
Why QoS is an Issue for IP

- **Today´s solution to reduce delay**
  - increase service rate (e.g. higher bandwidth)
    - long term process (note: experience shows that short term adaptation by load-dependent dynamic routing does not work well because of oscillation problems)
  - reduce the traffic load
    - for TCP streams e.g. done by slow start and congestion avoidance mechanism (van Jacobson)
- **But over-provisioning the network**
  - to achieve enough bandwidth and minimum delay for the worst case traffic scenario
  - cannot be economically justified

What is QoS?

- **QoS mechanisms manage the available bandwidth**
  - according to policies or to grant fairness
  - by distinguishing different traffic classes
- **Goal: Guaranteed lower bounds of quality**
  - minimum limits for bandwidth
  - maximum limits for delay
- **QoS does not create bandwidth but manages it**
  - so it is used more effectively

QoS Components



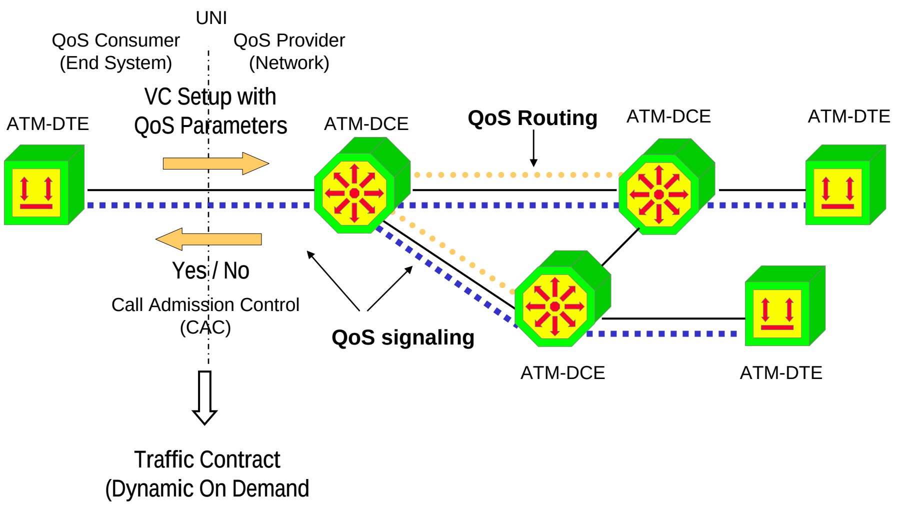
QoS Components

- **Signaling**
  - typically used for resource reservation
  - imitated by the end system
- **Queuing**
  - applies policy by prioritizing some packets over others
  - done on every intermediate store and forward system (packet switch) and maybe done at the sending end system
- **Classification**
  - detection of specific traffic classes or traffic flows
  - done at the network boundary
- **Marking**
  - packets belonging to some specific traffic class can be marked (= assigned a label) to request a special service
  - done at the network boundary

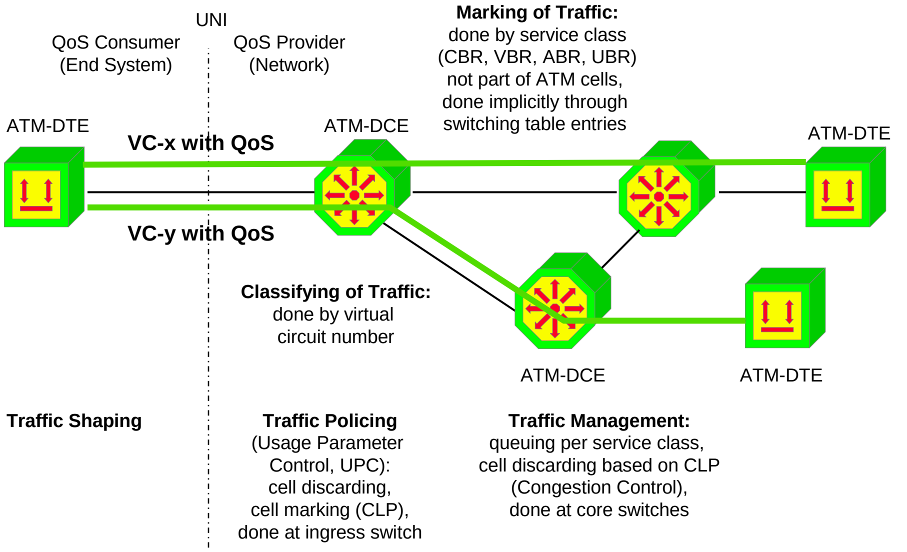
QoS Components

- **Traffic shaping**
  - smoothens bursty traffic by introducing delay
  - done by the end system or at the network boundary
- **Congestion control**
  - reduces packet rate when network congestion occurs
- **Admission control**
  - provides QoS features only to dedicated users
- **QoS policy**
  - fundamental QoS agreements specifying detail how to handle traffic, traffic classes, signaling, etc.
  - part of a traffic contract
- **Two fundamental IP QoS realization approaches**
  - Integrated Service versus Differentiated Services

ATM – QoS Review 1



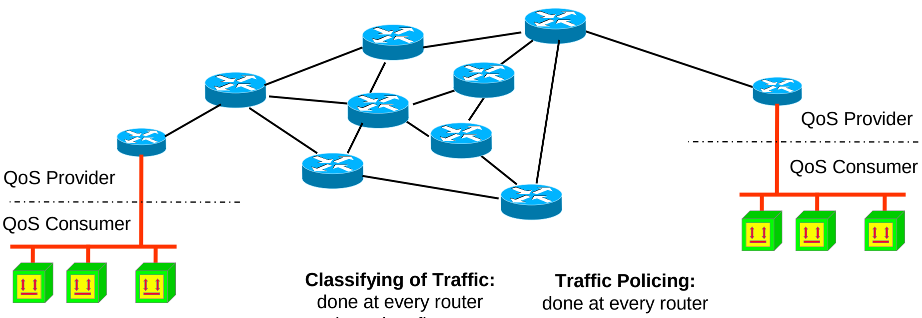
ATM – QoS Review 2



IP QoS Models

- **Integrated Services Model**
  - end-to-end guaranteed QoS for individual flows
    - flow is a stream of packets of the same session
  - connection-oriented approach
  - dynamic QoS mechanism -> requires a signaling protocol
    - RSVP (Resource ReSerVation Protocol, RFC 2205)
  - best suited for private networks
- **Differentiated Service**
  - QoS handling according to traffic classes
    - identified by DSCP (Differentiated Services Code Point)
    - based on per hop behavior (PHB)
  - connection-less approach
  - static QoS mechanism -> requires a fixed traffic contract
  - best suited for ISP in order to offer QoS services to customers

IP QoS Scenario: Integrated Services



Classifying of Traffic: done at every router based on flows

Traffic Policing: done at every router

Marking of Traffic: done by end system by specifying flows

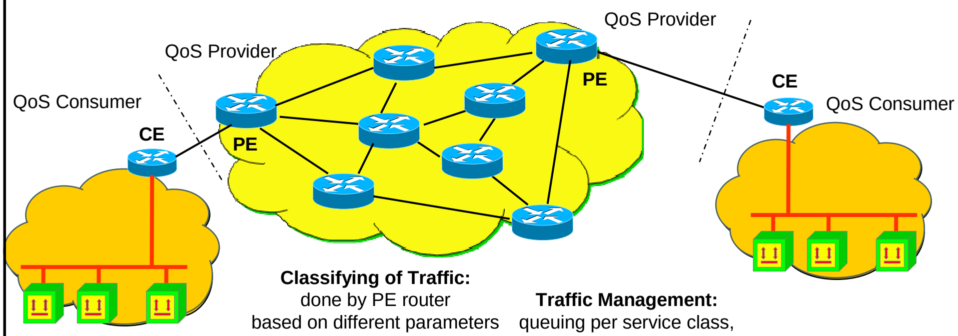
Traffic Management: queuing per flow, done by every router

Traffic Shaping: done by end system

Signaling: initiated by end system, passed on by routers, done by RSVP

Call Admission Control: done at every router

IP QoS Scenario: Differentiated Services



Classifying of Traffic:
done by PE router
based on different parameters
(e.g. interface, IP, TCP header)

Traffic Management:
queuing per service class,
done by every core router

Traffic Shaping:
done by CE router

Marking of Traffic:
done by PE router
by specifying DSCP
(service class)

Call Admission Control:
done by provider by
provisioning network
resources for service classes

Signaling:
not necessary because
of static approach

Traffic Policing:
done at PE router by CAR
(Committed Access Rate)

Final Remark

- Simplicity let IP succeed over other technologies
- With the demand for QoS the Internet will lose its simplicity!

Performance Measure

- **Performance measures used to characterize a connections performance**
  - Bandwidth
    - rated throughput capacity of a connection
    - describes the “size of the pipe” required for the appliaction to communicate over the network
  - Delay
    - serialization delay
    - propagation delay
    - switching / queuing delay
  - Packet Loss
    - packet drops because of congestion
    - corrupted packets because of transmission errors

 IP QoS Primer, v4.3 17

Agenda

- **IP QoS Introduction**
- **<u>Queue Management</u>**
- **Congestion Avoidance**
- **Traffic Rate Management**
- **Classification and Marking**

 IP QoS Primer, v4.3 18

What's the Purpose of Queuing ?

- **To avoid packet loss when temporary traffic bursts occur**
- **“Queue depth”**
  - number of packets waiting in the queue
  - small queue depth means low delay !
- **Congestion occurs when the queue overflows**
  - subsequent packets are dropped !
  - for the dropped packets the delay increase to infinity !

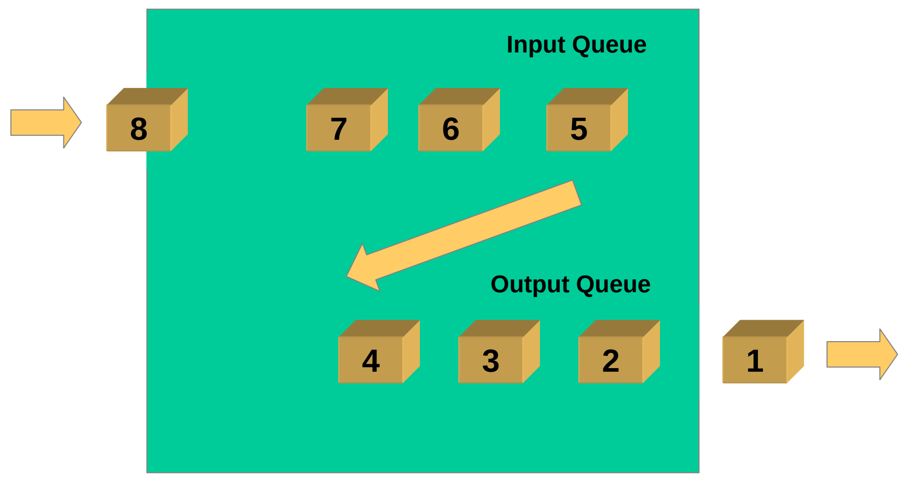
What is Queue-Management ?

- **Queue management means**
  - implementing a queuing strategy which packet should be transmitted next if packets are waiting in a queue
- **Several methods were developed**
  - FIFO
  - Priority Queuing
  - Class-based or Custom Queuing
  - Fair Queuing (FQ)
  - Weighted Fair Queuing (WFQ)
    - Flow-based
    - Class-based

FIFO Queuing

- **FIFO queuing is the fastest, simplest, and hence most common solution**
- **Only <u>one queue</u> for IP packet switching available**
- **Queuing theory shows**
  - if traffic increases the delay becomes longer and more variable
  - if load exceeds the service rate of outgoing interface the queue can increase infinitely and hence the router will drop packets caused by lack of buffer memory
- **In case of congestion: No predictable behaviour!**
  - usually, complete service degeneration

Principle of FIFO Queuing



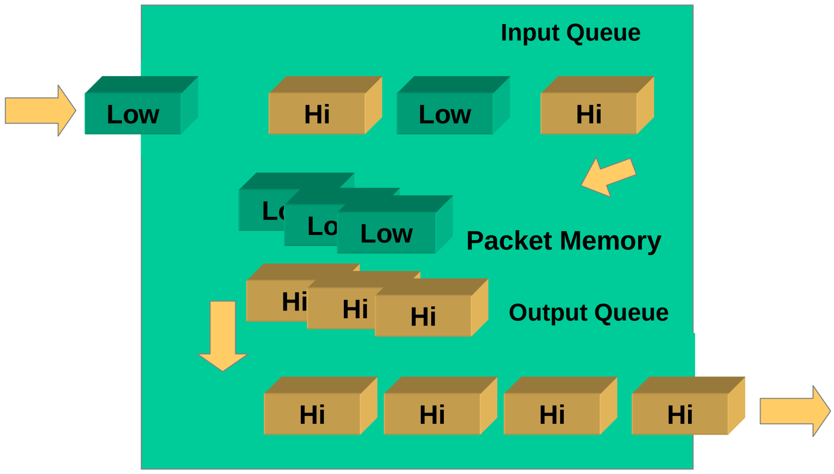
FIFO Benefits and Drawbacks

- **Benefits**
  - + simple and fast
  - + supported on all router platforms
- **Drawbacks**
  - - unfair allocation of bandwidth among multiple flows
  - - causes starvation (aggressive flows can monopolize links)
  - - causes jitter (bursts or packet trains temporarily fill the queue)

Priority Queuing

- **High priority packets are placed in the output queue before normal packets**
- **Several levels of priority possible**
  - prioritising certain protocols and services
    - E.g. IP before SNA, TCP before UDP, Telnet before FTP
  - requires additional (output) queues

Principle of Priority Queuing



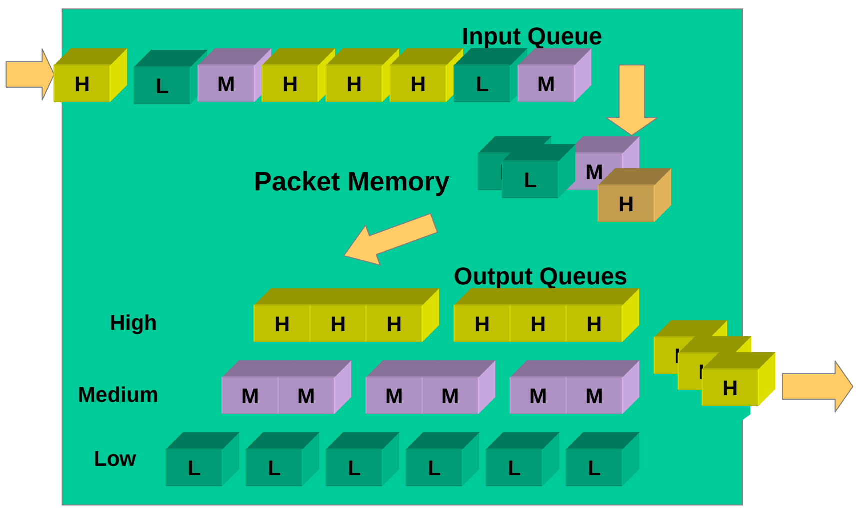
Priority Queuing Benefits and Drawbacks

- **Benefits**
  - \+ provides low-delay propagation to high priority packets
  - \+ supported on most router platforms
- **Drawbacks**
  - \- all drawbacks of FIFO queuing within a single class
  - \- packet identification is a time consuming process
  - \- large amounts of high priority traffic may lead to starvation of queues with lower priority
    - Lower priority traffic will be dropped
    - High delay for packets waiting in lower priority queues
  - \- manual configuration of classification on every hop

Class-Based Queuing

- Improvement of priority queuing
- Packets are sent from several priority queues in a round-robin manner
  - hereby avoiding service starvation of lower priority queues
- At each turn, depending on the priority of the queue, the router may only send a certain amount of data
- So, high priority queues are drained much faster than low priority queues
- But also the traffic with the lowest priority is served periodically

Principle of Class-Based Queuing



Class-Based Queuing

- **The router offers a different service quality for each class of traffic**
  - hence “class-based” queuing
- **Time consuming process**
  - appropriate on slow links only
- **Class-based queuing (CBQ) is also known as “Custom Queuing”**

 IP QoS Primer, v4.3 29

Lack of Fairness in CBQ

- **Class-Based Queuing is not fair !**
- **CBQ assigns protocols and services to different service classes but ignores session information**
  - for example, a misbehaving TCP session could push away other sessions within the same priority level if it produces a huge volume of traffic

 IP QoS Primer, v4.3 30

CBQ Benefits and Drawbacks

- **Benefits**
  + guarantees throughput to traffic classes (prevents starvation between traffic classes)
  + supported on most platforms

- **Drawbacks**
  - all drawbacks of FIFO queuing within a single class
  - manual configuration of classification on every hop
  - inaccurate bandwidth allocation
  - high jitter due to implementation of scheduling

Fair Queuing

- **Traditional datagram switching switch packets independently**
  – no memory about the past history
- **Fair queuing**
  – introduces state information by separating incoming traffic into well identified “flows”
    - Packets belonging to the same session build a flow
    - Flow could be recognized by same context
      – e.g. same source and destination IP addresses, same UDP/TCP ports and same TOS / DSCP in IPv4
      – e.g. same source IP address and same Flow-ID in IPv6
  – guarantees each flow an equal share of the transmission capacity

Fair Queuing

- **Theoretical principle**
  - determine the number of active flows (conversations)
  - store packets of every conversation in separate queues
  - serve queues round-robin bit-by-bit
    - Time division multiplexing with equal time-slots
- **In praxis**
  - packet cannot transmitted one bit at a time
  - but if you know the amount of current conversations you can calculate how many bit/s per conversation should be transmitted on average
    - number of conversations N, Bitrate of link R in 1/seconds
    - time to transmit a bit = N/R

Fair Queuing

- **method**
  - if you mark the time of packet arrival (Ta) you can calculate a virtual time Tv
    - Tv = Ta + (packet length in bit) * (N/R)
  - virtual time represents the time when this packet will be completely transmitted using the average rate for that queue
  - hence this time (Tv) can be used for the sending order of the packets
    - queues with longer packets will be served less often than queues with smaller packets
  - the only difference with perfectly fair queuing bit-by-bit
    - the queue that just transmitted a packet is slightly in advance by at most one packet of data

Weighted Fair Queuing (WFB)

- **Fair queuing gives every flow an equal portion of the link capacity**
- **Weighted fair queuing gives certain flows a larger portion of the link capacity**
  - certain flows have more “weight”
  - time division multiplexing with unequal time slots
- **Weighting the traffic can be based on**
  - the IP precedence bits in the TOS (Type of Service) field of the IP packet header

Weighted Fair Queuing (WFQ)

- **2 Types of WFQ**
  - Flow-based WFQ
    - active flows with the same IP precedence traffic get the same amount of interface bandwidth
    - active flows with high precedence traffic get a larger amount of interface bandwidth than active flows with lower-precedence traffic
  - Class-based WFQ
    - combination of class-based and WFQ

WFQ Benefits and Drawbacks

- **Benefits**
  - + simple configuration
  - + guarantees throughput to all flows
  - + drops packets of most aggressive flows
  - + supported on most platforms
- **Drawbacks**
  - - all drawbacks of FIFO queuing within a single queue
  - - multiple flows can end up in one queue
  - - does not support the configuration of classification
  - - can not provide fixed bandwidth guarantees
  - - performance limitations due to complex classification and scheduling mechanism

Queue Comparison

Weighted Fair queuing	Priority queuing	Custom queuing
No queue lists	4 queues	16 queues
Low-volume traffic given priority	High-priority queue serviced first	Round-robin service
Conversation dispatching	Packet-by-packet dispatching	Threshold dispatching
Interactive traffic gets priority	Critical traffic gets through	Proportional allocation of bandwidth
Works well on speeds Up to 2 Mbps	Designed for Low-bandwidth links	Designed for Medium speed links
Enabled by default	Must be configured	Must be configured

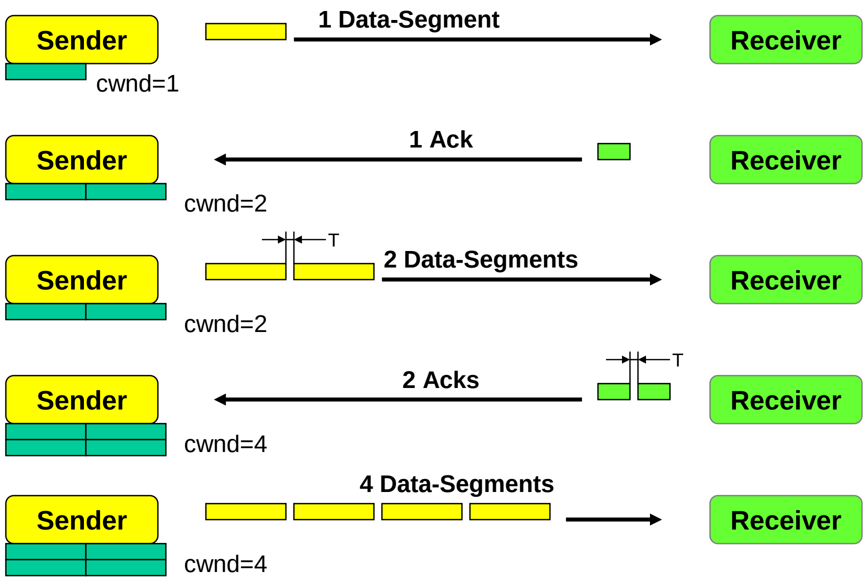
Agenda

- IP QoS Introduction
- Queue Management
- Congestion Avoidance
- Traffic Rate Management
- Classification and Marking

TCP's Slow Start and Congestion Avoidance

- TCP (Transmission Control Protocol) is currently the dominant transport protocol used on the Internet
- TCP uses "Slow Start" and "Congestion Avoidance"
  - in order to find the point of maximal data throughput before congestion occurs,
  - both algorithms have been invented by Van Jacobson
  - mandatory in today's TCP implementations

Slow Start 3



Congestion

- **Slow start encounters congestion, when**
  - TCP segment(s) is (are) dropped by a router
- **Congestion can be detected by the sender**
  - through timeouts

    or
  - duplicate acknowledgements
- **TCP reduces its sending rate**
  - with Congestion Avoidance"

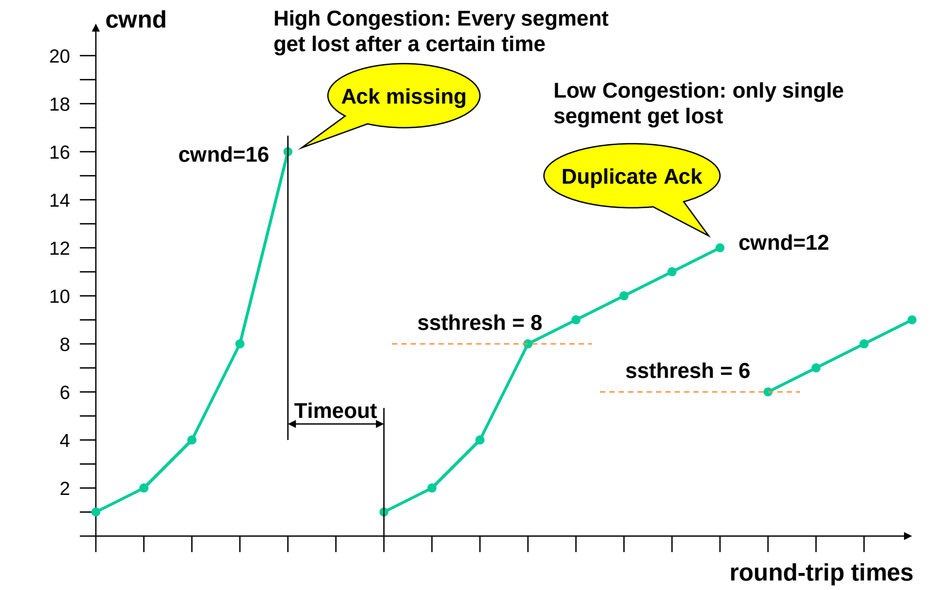
Congestion Avoidance 1

- **Slow start with congestion avoidance is a <u>sender-imposed flow control</u>**
  - Congestion Avoidance requires TCP to maintain a variable called <u>"slow start threshold"</u> (ssthresh)
  - Initially, ssthresh is set to TCPs maximum possible MSS (i.e. 65,535 octets)
- **On detection of congestion, ssthresh is set to <u>half the current window size</u>**
  - here, window size means: minimum of advertised window and congestion window (but at least 2 segments)
  - Note: ssthresh marks a safe window size because congestion occurred at a window size of 2 x ssthresh

Congestion Avoidance 2

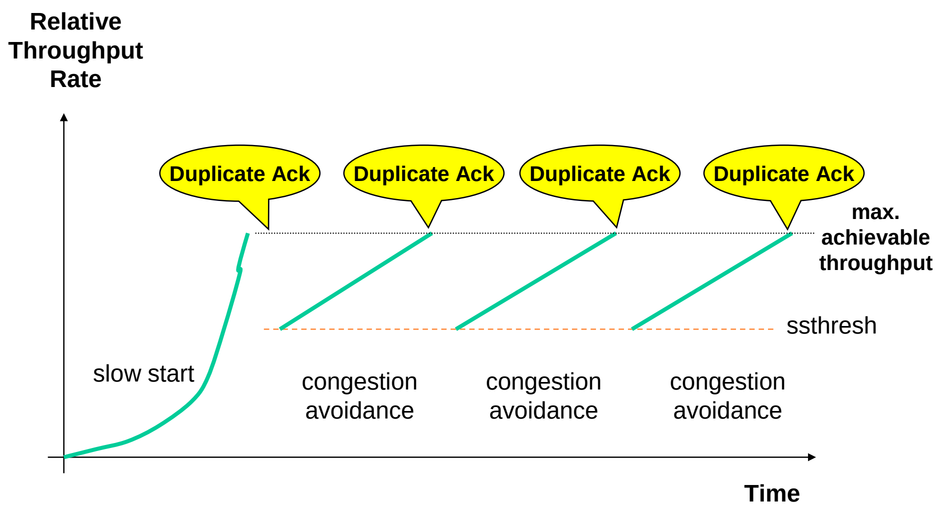
- **If the congestion is indicated by**
  - a timeout:
    - cwnd is set to 1 -> forcing slow start again
  - a duplicate ack:
    - cwnd is set to ssthresh (= 1/2 current window size)
- **cwnd ≤ ssthresh:**
  - slow start, doubling cwnd every round-trip time
  - <u>exponential</u> growth of cwnd
- **cwnd > ssthresh:**
  - congestion avoidance, cwnd is incremented by <u>MSS × MSS / cwnd</u> every time an ack is received
  - <u>linear</u> growth of cwnd

Slow Start and Congestion Avoidance



 IP QoS Primer, v4.3 45

Long Term View of TCP Throughput

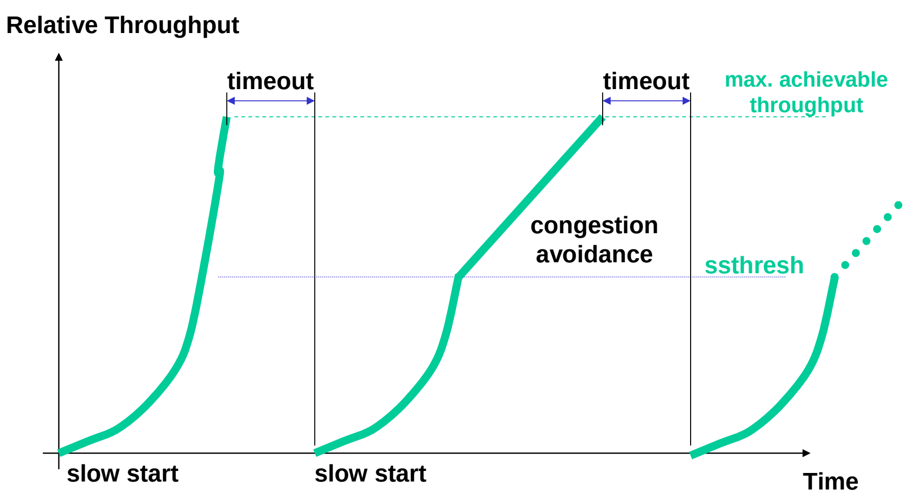


 IP QoS Primer, v4.3 46

Congestion Situations

- **Congestion indicated by duplicate acknowledgements**
  - lets TCP oscillate between sstresh and the point of congestion
  - no delays at the point of congestion because of immediate duplicate acks sent by the receiver
- **Congestion indicated by timeout:**
  - High delay and slow start necessary !

Congestion with Timeouts



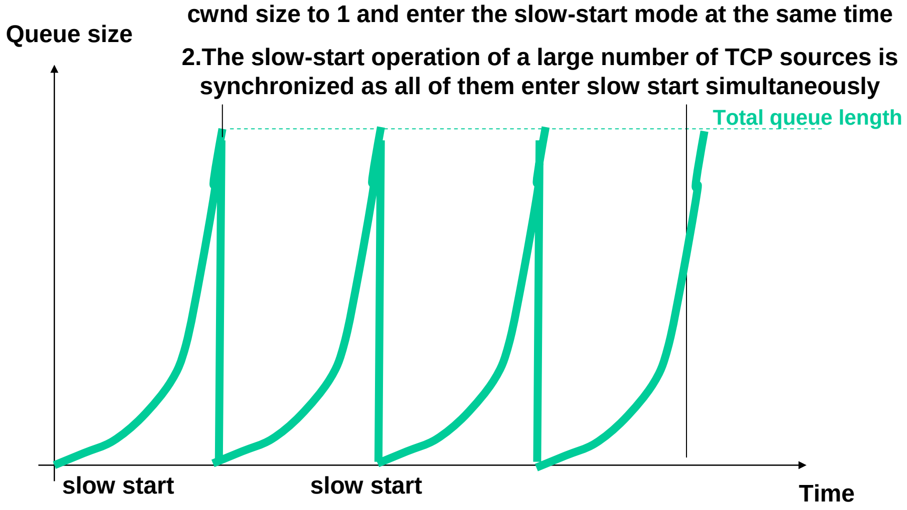
Avoiding Timeouts

- **While duplicate acks only indicate the possibility of congestion, timeouts can be regarded as a clear congestion alert**
- **Timeouts let TCP degenerate significantly**
  - especially when a lot of TCP streams get synchronized because of a Tail-Drop situation
- **Tail-Drop**
  - means that from a certain point on (when the FIFO queue is completely full) all TCP streams will recognize timeouts
- **Remedy: Proactive Queue Management**
  - enables routers to detect congestion before queues overflow
  - RED, ECN

Tail-Drop Scenario or Wave effect

1.Tail-drop causes a large number of TCP sources to change the cwnd size to 1 and enter the slow-start mode at the same time

2.The slow-start operation of a large number of TCP sources is synchronized as all of them enter slow start simultaneously



Base Idea for Random Early Detection (RED)

- **"Wave Effect"**
  - causes TCP's throughput oscillation
  - causes inefficient use of network bandwidth
- **What is the problem?**
  - routers accept every packet until the moment of congestion then data is lost and timeouts occur
    - error rate before congestion: ≈ 0%
error rate after congestion: ≈ 100%
- **Why not exploit the signaling effect of duplicate acknowledgments to reduce the TCP load before timeouts occur ?**

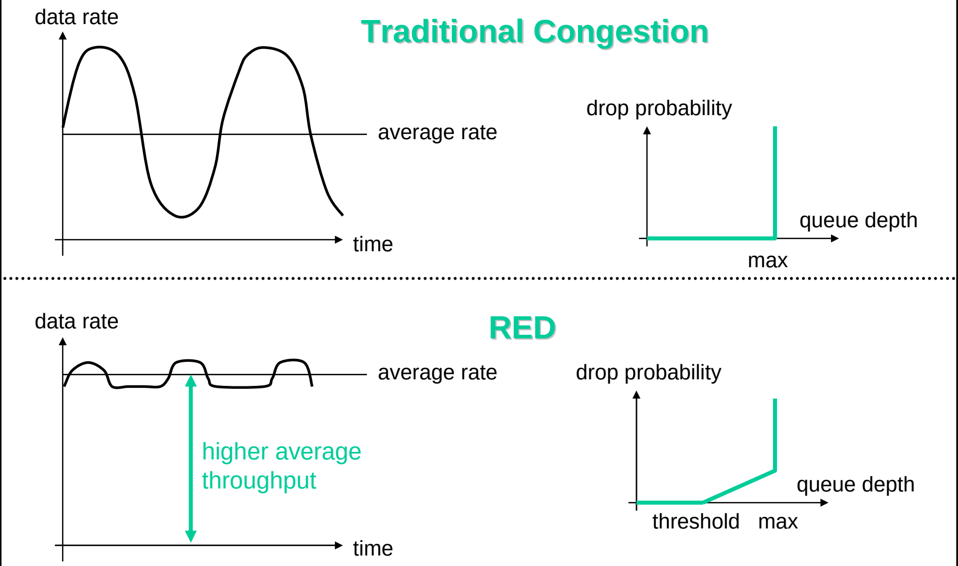
Random Early Detection (RED)

- **A router anticipates congestion by observing its queue depth**
- **If the queue depth exceeds a certain threshold, the router discards randomly chosen packets**
- **This causes the TCP receiver(s) to issue duplicate acks to their TCP sender(s)**
- **So the TCP sender(s) will immediately reduce their sending window**

Random Early Detection (RED)

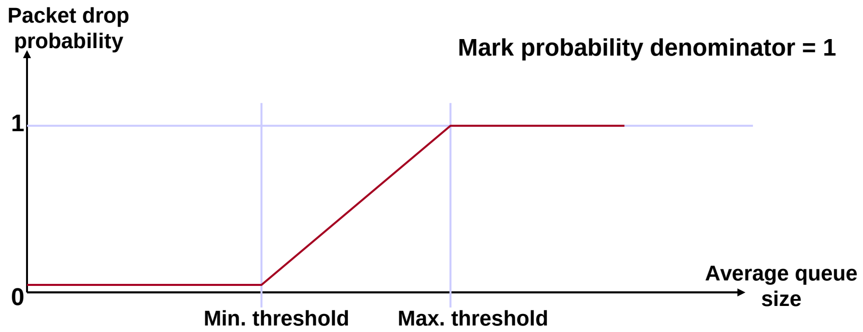
- **The trick is to increase the probability of packet drop after exceeding the threshold**
- **So theoretically, congestion will not occur anymore**
  - by dropping almost every packet at the end
- **Note: Without RED**
  - every single TCP connection experiences the wave effect
  - all waves get synchronized because the network alternates between congestion and non-congestion states
- **RED scrambles these waves**
  - providing a higher average TCP throughput

Wave Effect and RED



RED Packet Drop Probability

- When the average queue depth is above the minimum threshold, RED starts dropping packets
- The packet drop rate increases linearly as the average queue size increases, until the average queue size reaches the maximum threshold
- When the average queue size is above the maximum threshold, all packets are dropped



WRED (Weighted Random Early Detection)

- WRED allows selective RED parameters based on IP precedence
- WRED drops more aggressively for low-precedence-level packets and less aggressively for high-precedence-level packets

Flow WRED (I)

- Adaptive TCP flows respond to a congestion signal and reduce their load
- Non-adaptive UDP flows do not respond to congestion signals and don't slow down
- So non-adaptive flows(UDP) can send packets which higher rate than adaptive flows (TCP) at time of congestion
- Greedy, non-adaptive flows (UDP) tend to use a higher queue resource than the adaptive flows (TCP)

Flow WRED (II)

- To provide fairness, WRED classifies all arriving packets into the queue based on their flow and precedence
- WRED used state information to determine the resources for each flow
- Flows which taking more than their fair share would be penalized

The Limits of Interpreting Symptoms Only

- **Slow start and congestion avoidance try to maximize the traffic throughput without inclusion of network information**
  - host-based congestion control
  - original IP idea: "Keep the network simple !"
  - slow start and congestion avoidance suspects congestion only by observing symptoms of the network
- **Further improvements require an active inclusion of the intermediate network**
- **Led to the introduction of an Explicit Congestion Notification, which requires the help from routers that are expecting congestion**

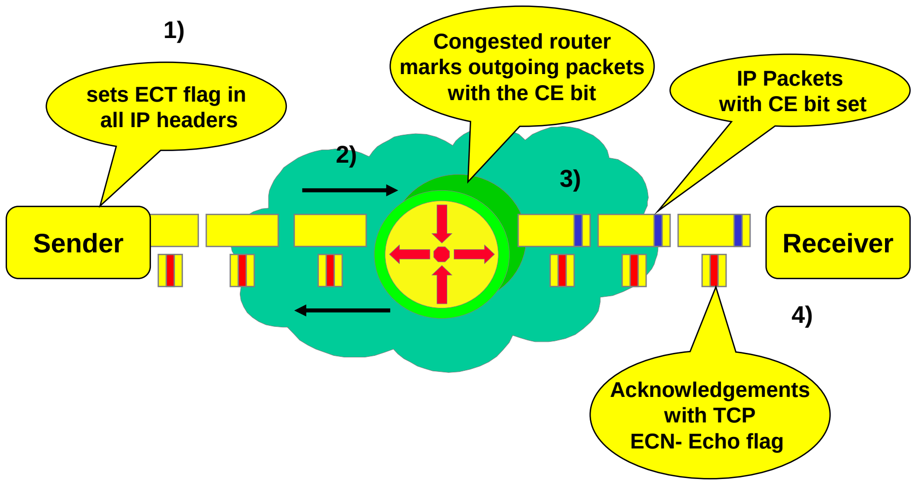
Explicit Congestion Notification (ECN)

- **During TCP connection establishment, the ECN capability is negotiated**
  - ECN utilizes bit 6 and 7 of the IPv4 TOS field
    - ECT (Explicit Congestion Notification Transport System)
    - CE (Congestion Experienced)
  - additionally ECN requires the two TCP options
    - "ECN-Echo" and "Congestion Window Reduced" (CWR)
- **Then the sender**
  - sets the ECT bit in the IP header of all datagram it sends
- **When routers experience congestion**
  - they may mark the IP header of such packets with an explicit CE bit flag

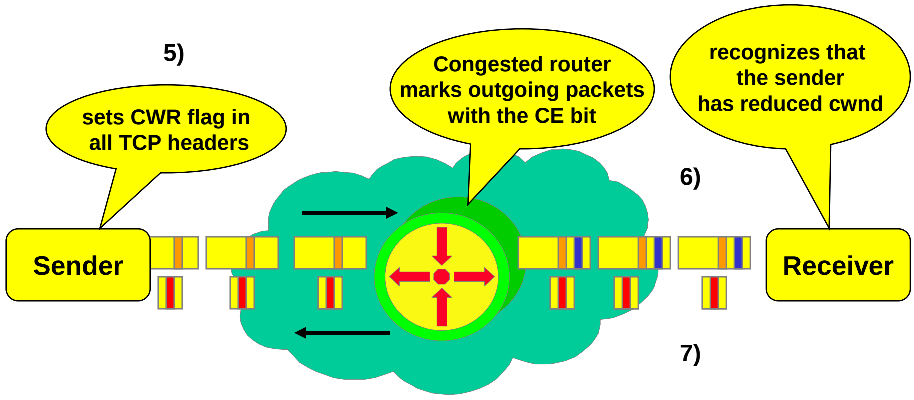
Explicit Congestion Notification (ECN)

- **The receiver detects the CE flag**
  - and sets the TCP ECN-Echo flag in its acknowledgement segment
- **If the sender receives this acknowledgement segment with the ECN-echo flag set,**
  - the sender reduces its congestion window (-> congestion avoidance)
  - the sender sets the TCP CWR flag in its next segment in order to notify the receiver that the sender has reacted upon the congestion
- **Main advantage:**
  - the sender does not have to wait for three duplicate acks to detect the congestion

Explicit Congestion Notification (ECN) 1

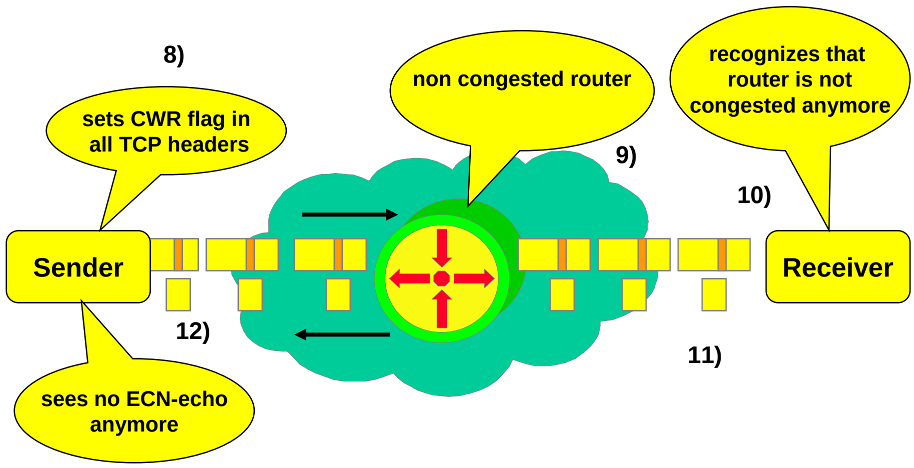


Explicit Congestion Notification (ECN) 2



 IP QoS Primer, v4.3 63

Explicit Congestion Notification (ECN) 3



 IP QoS Primer, v4.3 64

Agenda

- IP QoS Introduction
- Queue Management
- Congestion Avoidance
- Traffic Rate Management
- Classification and Marking

Traffic Rate Management

- **To offer QoS in a network, traffic entering the service provider network needs to be policed on the network boundary routers to make sure the traffic rate stays within the service limit**
  - CAR (Committed Access Rate)
- **To ensure that traffic conforms to a traffic contract the customer needs traffic shaping**
  - TS (Traffic Shaping)
- **Both use the token bucket scheme**
  - to report whether a packet is compliant or noncompliant with the rate parameters configured for it
  - Token bucket is simply a measuring tool

Traffic Policing versus Traffic Shaping

Policing Function (CAR)	Shaping Function (TS)
Sends conforming traffic up to the line rate and allows bursts	Smoothes traffics and send it out at a constant rate
When tokens are exhausted, it can drop packets	When token exhausted, it buffers packets and sends them out later, when tokens are available
Works for both input and output traffic	Implemented for output traffic only
When a packet drop ->TCP lowering its window size	TCP can detect and adapt its retransmission timer accordingly. It is more TCP-friendly

Bucket Models

Two traffic shaping models are common:

- **“Leaky Bucket”**
- **“Token Bucket”**

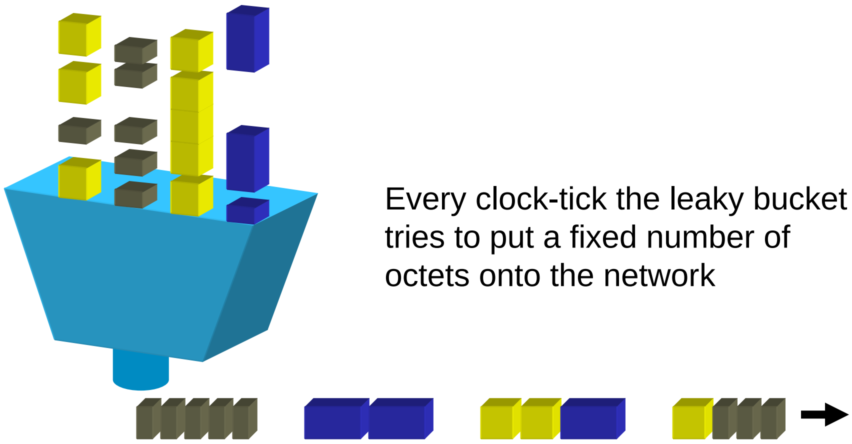
Leaky Bucket

- **Realized as a finite FIFO queue**
  - when queue is full, successive packets are discarded
- **Every clock-tick, a fixed number of octets is put onto the network**
  - e.g. one packet with 65536 octets or two packets with 32768 octets etc.
  - of course, a 35000 and a 45000 octet packet cannot be send at once

Leaky Bucket

- **Shapes bursty traffic into steady traffic only**
  - easy to implement; only bucket depth and leak-rate as parameters
- **Cannot fully utilize the network since it prevents bursts even if there are enough network resources**
- **Originated in the ATM world to control the cell rate**

Leaky Bucket Principle



Every clock-tick the leaky bucket tries to put a fixed number of octets onto the network

Drawbacks of Constant Rate

- **When large bursts arrive, it would be better to increase the leaky bucket output rate**
  - how long does this work before congestion occurs?
  - has the network enough resources at the moment?
- **Solution: Token Bucket**

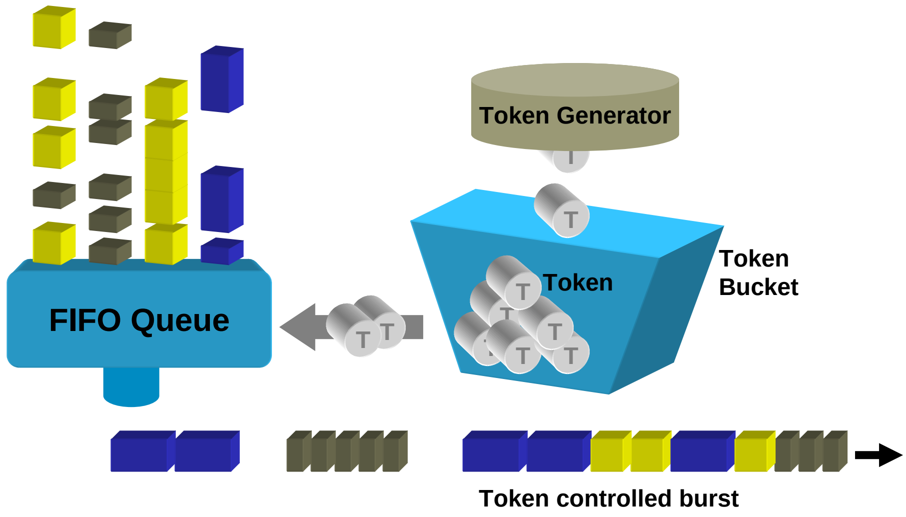
Token Bucket

- Token bucket does not receive traffic but tokens
- Tokens are produced at a constant rate by a token generator
- Token bucket is control mechanism for a FIFO queue
  - if token bucket is full then all following tokens are dropped

Token Bucket

- Each token represents a certain amount of octets
- The FIFO queue may send this amount of octets by consuming one token
- The FIFO queue may send even more at a time, as long there are tokens in the bucket
  = Controlled bursts !

Token Bucket Principle



 IP QoS Primer, v4.3 75

Agenda

- IP QoS Introduction
- Queue Management
- Congestion Avoidance
- Traffic Rate Management
- Classification and Marking

 IP QoS Primer, v4.3 76

Packet Classifier

- **Packet classification means identification**
- **Packets can be identified**
  - by parameter: source IP-address, destination IP-address, IP-protocol field, source and destination ports
    - flow concept
  - by IP-Precedence or DSCP-field (Differentiated Service Code Point)
    - class concept
  - by flow-id and source IP address (IPv6)

Packet Marking

- **A marker function is used to color the classified traffic by setting either the IP-precedence or the DSCP field**
  - done at the network boundary
  - class concept
- **Within the network core**
  - a per-hop behavior (PHB) to the packets can be applied based on either the IP-precedence or the DSCP field marked in the packet header

QoS-Group – Internal Marker

- Is used to mark packets matching certain user-specified classification criteria
- Is an internal label to the router and is not part of the IP packet header

Attributes	IP Precedence	DSCP	QoS Groups
Scope of the classification	Entire network	Entire network	Internal to the router only
Number of Classes	8 classes (0-7)	64 classes (0-63)	100 classes (0-99)