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)

- 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

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 Policy
- Traffic Shaping
- Congestion Control
- Admission Control
- Traffic Policing
- Signaling
- Queuing
- Classification and Marking
- Integrated Service (RSVP)
- Differentiated Service (DSCP)
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**
  - smoothes 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

- **Traffic Contract** (Dynamic On Demand)
  - VC Setup with QoS Parameters
  - ATM-DCE ATM-DTE

Traffic Shaping

- Traffic Policing
  - Usage Parameter Control (UPC)
  - cell discarding based on CLP
  - done at ingress switch

Traffic Management

- Queuing per service class
- cell discarding based on CLP
- done at core switches

Marking of Traffic:
- done by service classes (CBR, VBR, ABR, UBR)
- not part of ATM cells
- done implicitly through switching table entries

Classifying of Traffic:
- done by virtual circuit number

Marking:
- done by service class (CBR, VBR, ABR, UBR)
- not part of ATM cells
- done implicitly through switching table entries
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

- Traffic Policing:
  - done at every router

- Traffic Management:
  - queuing per flow, done by every router

- Classifying of Traffic:
  - done at every router

- Marking of Traffic:
  - done by end system

- Call Admission Control:
  - done at every router

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

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 connection's performance
  - Bandwidth
    - rated throughput capacity of a connection
    - describes the "size of the pipe" required for the application 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

Agenda

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

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 one queue 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

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

- **Input Queue**
- **Output Queue**
- **Packet Memory**

#### 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

- **Input Queue**
- **Packet Memory**
- **Output Queues**
  - **High**
  - **Medium**
  - **Low**

© 2005, D.I. Manfred Lindner
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”

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

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 = \( \frac{N}{R} \)

**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

<table>
<thead>
<tr>
<th>Weighted Fair queuing</th>
<th>Priority queuing</th>
<th>Custom queuing</th>
</tr>
</thead>
<tbody>
<tr>
<td>No queue lists</td>
<td>4 queues</td>
<td>16 queues</td>
</tr>
<tr>
<td>Low-volume traffic</td>
<td>High-priority queue serviced first</td>
<td>Round-robin service</td>
</tr>
<tr>
<td>given priority</td>
<td>Packet-by-packet dispatching</td>
<td>Threshold dispatching</td>
</tr>
<tr>
<td>Conversation dispatching</td>
<td>Interactive traffic gets priority</td>
<td>Works well on speeds Up to 2 Mbps</td>
</tr>
<tr>
<td></td>
<td>Designed for Low-bandwidth links</td>
<td>Designed for Medium speed links</td>
</tr>
<tr>
<td>Enabled by default</td>
<td>Must be configured</td>
<td>Must be configured</td>
</tr>
</tbody>
</table>

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

Agenda

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

- Slow start with congestion avoidance is a sender-imposed flow control
  - Congestion Avoidance requires TCP to maintain a variable called "slow start threshold" (ssthresh)
  - Initially, ssthresh is set to TCPs maximum possible MSS (i.e. 65,535 octets)

- On detection of congestion, ssthresh is set to half the current window size
  - 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

- 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
  - exponential growth of cwnd

- cwnd > ssthresh:
  - congestion avoidance, cwnd is incremented by MSS × MSS / cwnd every time an ack is received
  - linear growth of cwnd
Slow Start and Congestion Avoidance

- High Congestion: Every segment get lost after a certain time
- Low Congestion: only single segment get lost

Round-Trip Times

- Ack missing
- Duplicate Ack

Congestion Situations

- Congestion indicated by duplicate acknowledgements
  - lets TCP oscillate between ssthresh 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!

Long Term View of TCP Throughput

- Relative Throughput Rate
- Slow start
- Congestion avoidance
- Congestion avoidance
- ssthresh

Congestion with Timeouts

- Relative Throughput
- Time
- Slow start
- Max. achievable throughput
- Congestion avoidance
- Timeout
- Ssthresh
Avoiding Timeouts

- While duplicate acks only indicate the possibility of congestion, timeouts can be regarded as a clear congestion alert.
- Treeouts 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.

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

<table>
<thead>
<tr>
<th>Traditional Congestion</th>
<th>RED</th>
</tr>
</thead>
<tbody>
<tr>
<td>data rate</td>
<td>data rate</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>average rate</td>
<td>average rate</td>
</tr>
<tr>
<td>drop probability</td>
<td>drop probability</td>
</tr>
<tr>
<td>queue depth max</td>
<td>queue depth max</td>
</tr>
</tbody>
</table>

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

1) Congested router marks outgoing packets with the CE bit
2) sets ECT flag in all IP headers
3) IP Packets with CE bit set
4) Acknowledgements with TCP ECN-Echo flag

Explicit Congestion Notification (ECN) 2

5) Congested router sets CWR flag in all TCP headers
6) Sender recognizes that the sender has reduced cwnd
7) Receiver

Explicit Congestion Notification (ECN) 3

8) Congested router marks outgoing packets with the CE bit
9) sets CWR flag in all TCP headers
10) non congested router
11) Recognizes that router is not congested anymore
12) Receives no ECN-echo anymore
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

<table>
<thead>
<tr>
<th>Policing Function (CAR)</th>
<th>Shaping Function (TS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sends conforming traffic up to the line rate and allows bursts when tokens are exhausted, it can drop packets</td>
<td>Smooths traffic and send it out at a constant rate when token exhausted, it buffers packets and sends them out later, when tokens are available</td>
</tr>
<tr>
<td>Works for both input and output traffic</td>
<td>Implemented for output traffic only</td>
</tr>
<tr>
<td>When a packet drops -&gt;TCP lowering its window size</td>
<td>TCP can detect and adapt its retransmission timer accordingly, it is more TCP-friendly</td>
</tr>
</tbody>
</table>

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 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 a control mechanism for a FIFO queue
  - If token bucket is full, all following tokens are dropped

**Token Bucket Principle**

- 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!

**Agenda**

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

<table>
<thead>
<tr>
<th>Attributes</th>
<th>IP Precedence</th>
<th>DSCP</th>
<th>QoS Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope of the classification</td>
<td>Entire network</td>
<td>Entire network</td>
<td>Internal to the router only</td>
</tr>
<tr>
<td>Number of Classes</td>
<td>8 classes (0-7)</td>
<td>64 classes (0-63)</td>
<td>100 classes (0-99)</td>
</tr>
</tbody>
</table>