Quality of Service on the Internet

C. Pham

DEA DIF lecture

These slides borrow material from various sources which are indicated below each slide when necessary

Slides mostly taken from Shivkumar Kalyanaraman which are mostly based on slides of Ion Stoica, Jim Kurose, Srini Seshan, Srini Keshav
Multimedia, real time on the Internet

- **Real-time applications**
  - Interactive applications are sensitive to packet delays (telephone)
  - Non-interactive applications can adapt to a wider range of packet delays (audio, video broadcasts)
  - Guarantee of maximum delay is useful
Time-constrained applications

- Elastic applications
  - Interactive data transfer (e.g. HTTP, FTP)
    - Sensitive to the average delay, not to the distribution tail
  - Bulk data transfer (e.g. mail and news delivery)
    - Delay insensitive
  - Best effort works well

Document is only useful when it is completely received. This means average packet delay is important, not maximum packet delay.
Discussion

What is the problem?

- Different applications have different delay, bandwidth, and jitter needs
- Some applications are very sensitive to changing network conditions: the packet arrival time distribution is important

Solutions

- Make applications adaptive
- Build more flexibility into network
Why Better-than-Best-Effort (QoS)?

- To support a wider range of applications
  - Real-time, Multimedia, etc

- To develop sustainable economic models and new private networking services
  - Current flat priced models, and best-effort services do not cut it for businesses
What do we have now?

Multimedia applications: network audio and video

Impairments:
- excessive delay: gaps in rendered audio, video
- excessive data loss
Quality of Service: What is it?

Multimedia applications: network audio and video

QoS

network provides application with *level of performance needed for application to function.*
What is QoS?

- “Better performance” as described by a set of parameters or measured by a set of metrics.

  **Generic parameters:**
  - Bandwidth
  - Delay, Delay-jitter
  - Packet loss rate (or loss probability)

  **Transport/Application-specific parameters:**
  - Timeouts
  - Percentage of “important” packets lost
What is QoS (contd)?

- These parameters can be measured at several granularities:
  - "micro" flow, aggregate flow, population.

- QoS considered "better" if
  - more parameters can be specified
  - QoS can be specified at a fine-granularity.

- QoS spectrum:

  [Diagram showing Best Effort and Leased Line with a spectrum scale]
QoS: why don’t we have it?

- QoS a concern since early 1980’s.
- Look at what’s happened since 1980:
  - Internet now a million times larger!
  - Applications: WWW, Napster, eBay, Gopher

- Why limited progress on QoS?
  - Lots of smart people working on it!

Why is the QoS problem so hard?

The Internet is a huge transformative success!
Why was the WWW so “easy”?

Implemented in hosts, servers at “network edge”
- “on top of” existing network
- “complexity at network edge”
- no changes to network core
Why is QoS more difficult?

- today’s Internet core provides “best effort” service
  - network congestion causes delays, loss
  - no timing guarantees
  - no loss guarantees
- multimedia requires loss, timing constraints met

“The different timing and reliability constraints of real-time communication require new protocols and architectures to be developed”

wet-behind-the-ears researcher, 1982.

New architecture needed for network core!
Improving QOS in IP Networks

- IETF groups are working on proposals to provide better QOS control in IP networks, i.e., going beyond best effort to provide some assurance for QOS.
- Work in Progress includes RSVP, Differentiated Services, and Integrated Services.
- Simple model for sharing and congestion studies:
Principles for QOS Guarantees

- Consider a phone application at 1Mbps and an FTP application sharing a 1.5 Mbps link.
  - bursts of FTP can congest the router and cause audio packets to be dropped.
  - want to give priority to audio over FTP

- PRINCIPLE 1: Marking of packets is needed for router to distinguish between different classes; and new router policy to treat packets accordingly
Principles for QOS Guarantees (more)

- Applications misbehave (audio sends packets at a rate higher than 1Mbps assumed above);
- PRINCIPLE 2: provide protection (isolation) for one class from other classes
- Require Policing Mechanisms to ensure sources adhere to bandwidth requirements; Marking and Policing need to be done at the edges:
Principles for QOS Guarantees (more)

- Alternative to Marking and Policing: allocate a set portion of bandwidth to each application flow; can lead to inefficient use of bandwidth if one of the flows does not use its allocation

- PRINCIPLE 3: While providing isolation, it is desirable to use resources as efficiently as possible
Principles for QOS Guarantees (more)

- Cannot support traffic beyond link capacity
- PRINCIPLE 4: Need a Call Admission Process; application flow declares its needs, network may block call if it cannot satisfy the needs
Summary

QoS for networked applications

- Packet classification
- Isolation, scheduling, and policing
- High resource utilization
- Call admission
Fundamental Problems

- In a FIFO service discipline, the performance assigned to one flow is convoluted with the arrivals of packets from all other flows!
  - Can't get QoS with a “free-for-all”
  - Need to use new scheduling disciplines which provide “isolation” of performance from arrival rates of background traffic

**FIFO**

**Scheduling Discipline**
How to upgrade the Internet for QoS?

- **Approach**: de-couple end-system evolution from network evolution

- **End-to-end protocols**: RTP, H.323, etc to spur the growth of adaptive multimedia applications
  - Assume best-effort or better-than-best-effort clouds

- **Network protocols**: IntServ, DiffServ, RSVP, MPLS, COPS ...
  - To support better-than-best-effort capabilities at the network (IP) level
QOS SPECIFICATION, TRAFFIC, SERVICE CHARACTERIZATION, BASIC MECHANISMS
Service Specification

- **Loss**: probability that a flow's packet is lost
- **Delay**: time it takes a packet's flow to get from source to destination
- **Delay jitter**: maximum difference between the delays experienced by two packets of the flow
- **Bandwidth**: maximum rate at which the source can send traffic
- **QoS spectrum**:

  ![Diagram]

  - **Best Effort**
  - **Leased Line**
Hard Real Time: Guaranteed Services

- **Service contract**
  - Network to client: guarantee a deterministic upper bound on delay for each packet in a session
  - Client to network: the session does not send more than it specifies

- **Algorithm support**
  - Admission control based on worst-case analysis
  - Per flow classification/scheduling at routers
Soft Real Time: Controlled Load Service

- **Service contract:**
  - Network to client: similar performance as an unloaded best-effort network
  - Client to network: the session does not send more than it specifies

- **Algorithm Support**
  - Admission control based on measurement of aggregates
  - Scheduling for aggregate possible
Traffic and Service Characterization

- To quantify a service one has two know:
  - Flow’s traffic arrival
  - Service provided by the router, i.e., resources reserved at each router

- Examples:
  - Traffic characterization: token bucket
  - Service provided by router: fix rate and fix buffer space
    - Characterized by a service model (service curve framework)
Ex: Token Bucket

- Characterized by three parameters \((b, r, R)\)
  - \(b\) - token depth
  - \(r\) - average arrival rate
  - \(R\) - maximum arrival rate (e.g., R link capacity)

- A bit is transmitted only when there is an available token
  - When a bit is transmitted exactly one token is consumed
Token Bucket

Example

- $B = 4000$ bits, $R = 1$ Mbps, $C = 10$ Mbps
- Packet length = 1000 bits
- Assume the bucket is initially full and a “large” burst of packets arrives

istoica@cs.cmu.edu
Token Bucket

- **time = 0**
- **time = 0.1 ms**
- **time = 0.3 ms**
- **time = 1 ms**
- **time = 2 ms**
- **time = 3 ms**
Traffic Envelope (Arrival Curve)

- Maximum amount of service that a flow can send during an interval of time $t$

$\text{slope} = \text{max average rate}$

$b(t) = \text{Envelope}$

“Burstiness Constraint”

$slope = \text{peak rate}$
Arrival curve

\[ A(t) - \text{number of bits received up to time } t \]
Characterizing a Source by Token Bucket

- Arrival curve - maximum amount of bits transmitted by time $t$
- Use token bucket to bound the arrival curve
Per-hop Reservation with Token Bucket

- Given $b, r, R$ and per-hop delay $d$
- Allocate bandwidth $r_a$ and buffer space $B_a$ such that to guarantee $d$
What is a Service Model?

- The QoS measures (delay, throughput, loss, cost) depend on offered traffic, and possibly other external processes.

- A service model attempts to characterize the relationship between offered traffic, delivered traffic, and possibly other external processes.
Arrival and Departure Process

$R_{in}(t)$ = arrival process
$R_{in}(t)$ = amount of data arriving up to time $t$

$R_{out}(t)$ = departure process
$R_{out}(t)$ = amount of data departing up to time $t$
Delay and Buffer Bounds

\[ E(t) = \text{Envelope} \]

\[ S(t) = \text{service curve} \]

Maximum delay

Maximum buffer
SCHEDULING
Packet Scheduling

- Decide when and what packet to send on output link
- Usually implemented at output interface
Mechanisms: Queuing/Scheduling

- Use a few bits in header to indicate which queue (class) a packet goes into (also branded as CoS).
- High $$ users classified into high priority queues, which also may be less populated.
  - => lower delay and low likelihood of packet drop.
- Ideas: priority, round-robin, classification, aggregation, ...
Scheduling And Policing Mechanisms

- Scheduling: choosing the next packet for transmission on a link can be done following a number of policies;
- FIFO: in order of arrival to the queue; packets that arrive to a full buffer are either discarded, or a discard policy is used to determine which packet to discard among the arrival and those already in the queue.
Priority Queueing

- Priority Queuing: classes have different priorities; class may depend on explicit marking or other header info, e.g., IP source or destination, TCP Port numbers, etc.
- Transmit a packet from the highest priority class with a non-empty queue
- Preemptive and non-preemptive versions
Round Robin (RR)

- Round Robin: scan class queues serving one from each class that has a non-empty queue

![Diagram of Round Robin scheduling]

In one round, the system serves one packet from each non-empty queue.
Weighted Round Robin (WRR)

- Assign a weight to each connection and serve a connection in proportion to its weight.
- Ex:
  - Connection A, B and C with same packet size and weight 0.5, 0.75 and 1. How many packets from each connection should a round-robin server serve in each round?
  - Answer: Normalize each weight so that they are all integers: we get 2, 3 and 4. Then in each round of service, the server serves 2 packets from A, 3 from B and 4 from C.
(Weighted) Round-Robin Discussion

- **Advantages:** protection among flows
  - Misbehaving flows will not affect the performance of well-behaving flows
    - Misbehaving flow - a flow that does not implement any congestion control
  - FIFO does not have such a property
- **Disadvantages:**
  - More complex than FIFO: per flow queue/state
  - Biased toward large packets (not ATM) - a flow receives service proportional to the number of packets
- **If packet size are different, we normalize the weight by the packet size**
  - ex: 50, 500 & 1500 bytes with weight 0.5, 0.75 & 1.0
**Generalized Processor Sharing (GPS)**

- Assume a fluid model of traffic
  - Visit each non-empty queue in turn (like RR)
  - Serve infinitesimal from each
  - Leads to “max-min” fairness

- GPS is un-implementable!
  - We cannot serve infinitesimals, only packets

---

**max-min fairness**

Soit un ensemble de sources 1,...,n demandant des ressources $x_1,..,x_n$ avec $x_1<x_2..<x_n$ par exemple. Le serveur a une capacité $C$.

On donne alors $C/n$ à la source 1. Si $C/n>x_1$, on donne $C/n+(C/n-x_1)/(n-1)$ aux (n-1) sources restantes. Si cela est supérieur à $x_2$, on recommence.

(Existe en version max-min weighted faire share)
Packet Approximation of Fluid System

- **GPS un-implementable**
- **Standard techniques of approximating fluid GPS**
  - Select packet that finishes first in GPS assuming that there are no future arrivals (emulate GPS on the side)
- **Important properties of GPS**
  - Finishing order of packets currently in system independent of future arrivals
- **Implementation based on virtual time**
  - Assign virtual finish time to each packet upon arrival
  - Packets served in increasing order of virtual times
**Fair Queuing (FQ)**

- Idea: serve packets in the order in which they would have finished transmission in the fluid flow system
- Mapping bit-by-bit schedule onto packet transmission schedule
- Transmit packet with the lowest finish time at any given time
**FQ Simple Example**

Cannot preempt packet currently being transmitted
Round Number and Finish Number

- Single flow: *clock ticks when a bit is transmitted*. For packet \(k\):
  - \(P_k =\) length, \(A_k =\) arrival time, \(S_i =\) begin transmit time, \(F_k =\) finish transmit time
  - \(F_k = S_k + P_k = \max (F_{k-1}, A_k) + P_k\)

- Multiple flows: *clock ticks when a bit from all active flows is transmitted* → round number
  - Can calculate \(F_k\) for each packet if number of flows is known at all times
    - \(F_k =\) current round number + size of packet \(k\), inactive case
    - \(F_k =\) largest \(F_k\) in the queue + size of packet \(k\), active case
  - \(F_{i,k,t} = \max (F_{i,k-1,t}, R_t) + P_{i,k,t}\)
  - In packet approximation, finish number indicate a relative order (service tag) in which a packet is to be served. finish time ≠ finish number
Example

- The round number increases at a rate inversely proportional to the number of active connections
  - Thus is only used for computing finish numbers

- Largest finish number in a connection's queue is the connection's finish number

Example

- Suppose packets of size 1, 2 and 2 units arrive at a FQ scheduler at time for connection A, B and C. Also, assume that a packet of size 2 arrive for connection A at time 4. The link service rate is 1 unit/s. Compute the finish number of all packets.
FQ Advantages

- FQ protect well-behaved flows from ill-behaved flows
- Example: 1 UDP (10 Mbps) and 31 TCP’s sharing a 10 Mbps link
Weighted Fair Queueing

- Variation of FQ: Weighted Fair Queueing (WFQ)
- Weighted Fair Queueing: is a generalized Round Robin in which an attempt is made to provide a class with a differentiated amount of service over a given period of time.
Implementing WFQ

- WFQ needs per-connection (or per-aggregate) scheduler state – implementation complexity.
  - complex iterated deletion algorithm
  - complex sorting at the output queue on the service tag
- WFQ needs to know the weight assigned for each queue – manual configuration, signalling.
- WFQ is not perfect...
- Router manufacturers have implemented as early as 1996 WFQ in their products
  - from CISCO 1600 series
  - Fore System ATM switches
Approximating GPS with WFQ

- Fluid GPS system service order
- Weighted Fair Queueing
  - select the first packet that finishes in GPS
Big Picture

- FQ does not eliminate congestion → it just manages the congestion
- You need both end-host congestion control and router support for congestion control
  - end-host congestion control to adapt
  - router congestion control to protect/isolate
- Don’t forget buffer management: you still need to drop in case of congestion. Which packet’s would you drop in FQ?
  - one possibility: packet from the longest queue
Further readings

- See http://www.cnaf.infn.it/~ferrari/ispn.html for Quality of Service list of papers
- See http://www.cnaf.infn.it/~ferrari/sched.html for scheduling list of papers
QoS ARCHITECTURES
Stateless vs. Stateful QoS Solutions

- **Stateless** solutions - routers maintain no fine grained state about traffic
  - scalable, robust
  - weak services

- **Stateful** solutions - routers maintain per-flow state
  - powerful services
    - guaranteed services + high resource utilization
    - fine grained differentiation
    - protection
  - much less scalable and robust
# Existing Solutions

<table>
<thead>
<tr>
<th>QoS</th>
<th>Stateful</th>
<th>Stateless</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tenet [Ferrari &amp; Verma ’89]</td>
<td>DiffServ</td>
</tr>
<tr>
<td></td>
<td>IntServ [Clark et al ’91]</td>
<td>- [Clark &amp; Wroclawski ’97]</td>
</tr>
<tr>
<td></td>
<td>ATM [late ’80s]</td>
<td>- [Nichols et al ‘97]</td>
</tr>
</tbody>
</table>

|                          | Round Robin [Nagle ’85]                      | DecBit [Ramkrishnan & Jain ’88]              |
|                          | Fair Queueing [Demers et al ’89]            | Random Early Detection (RED) [Floyd & Jacobson ’93] |
| Network support for      | Flow Random Early Drop (FRED) [Lin & Morris ’97] | BLUE [Feng et al ’99]                        |
| congestion control       |                                              | REM [Low et al ’00]                          |
Integrated Services (IntServ)

- An architecture for providing QOS guarantees in IP networks for individual application sessions
- Relies on resource reservation, and routers need to maintain state information of allocated resources (eg: g) and respond to new Call setup requests
Integrated Services Model

- Flow specification
  - Leaky Bucket, Token Bucket
- Routing
- Admission control
- Policy control
- Resource reservation
  - RSVP
- Packet scheduling
  - WFQ, CBQ, RED
Integrated Services: Classes

- **Guaranteed QOS**: this class is provided with **firm bounds** on queuing delay at a router; envisioned for hard real-time applications that are highly sensitive to end-to-end delay expectation and variance.

- **Controlled Load**: this class is provided a QOS **closely approximating** that provided by an unloaded router; envisioned for today’s IP network real-time applications which perform well in an unloaded network.
Signaling semantics

- Classic scheme: sender initiated
- SETUP, SETUP_ACK, SETUP_RESPONSE
- Admission control
- Tentative resource reservation and confirmation
- Simplex and duplex setup; no multicast support
RSVP for the IntServ approach

- Resource reSerVation Protocol
- What is RSVP?
  - Method for application to specify desired QoS to net
  - Switch state establishment protocol (signaling)
  - Multicast friendly, receiver-oriented
  - Simplex reservations (single direction)
- Why run RSVP?
  - Allows precise allocation of network resources
  - Guarantees on quality of service
  - Heterogeneous bandwidth support for multicast
  - Scalable (?)

source Gordon Schaffee
Resource Reservation

- Senders advertise using PATH message
- Receivers reserve using RESV message
  - Flowspec + filterspec + policy data
  - Travels upstream in reverse direction of Path message
- Merging of reservations
- Sender/receiver notified of changes
RSVP Functional Diagram

Host

Router

Application

Packet Classifier

Packet Scheduler

Policy Control

Admissions Control

Routing Process

Packet Classifier

Packet Scheduler

Policy Control

Admissions Control
Stateful Solution: Guaranteed Services

- Achieve per-flow bandwidth and delay guarantees
- Example: guarantee 1MBps and < 100 ms delay to a flow
Stateful Solution: Guaranteed Services

- Allocate resources - perform per-flow admission control
Stateful Solution: Guaranteed Services

- Install per-flow state
Stateful Solution: Guaranteed Services

- **Challenge:** maintain per-flow state consistent
Stateful Solution: Guaranteed Services

- Per-flow classification
Stateful Solution: Guaranteed Services

- Per-flow buffer management
Stateful Solution: Guaranteed Services

- Per-flow scheduling
Stateful Solution Complexity

- **Data path**
  - Per-flow classification
  - Per-flow buffer management
  - Per-flow scheduling

- **Control path**
  - Install and maintain per-flow state for data and control paths
Stateless vs. Stateful

- **Stateless** solutions are more
  - scalable
  - robust
- **Stateful** solutions provide more powerful and flexible services
  - guaranteed services + high resource utilization
  - fine grained differentiation
  - protection
Can we achieve the best of two worlds, i.e., provide services implemented by stateful networks while maintaining advantages of stateless architectures?

Yes, in some interesting cases. DPS, CSFQ.

Can we provide reduced state services, i.e., maintain state only for larger granular flows rather than end-to-end flows?

Yes: Diff-serv
**DiffServ: Basic Ideas**

The real question is to choose which packets shall be dropped. The first definition of differential service is something like "not mine."
-- Christian Huitema

- Differentiated services provide a way to specify the relative priority of packets
- Some data is more important than other data
- People who pay for better service get it

source: Gordon Schaffee
Goals

- Ability to charge differently for different services
- Lightweight, scalable service discrimination suitable for network backbones
  - No per flow state or per flow signaling
- Deploy incrementally, then evolve
  - Build simple system at first, expand if needed in future
- Make service separate from signaling

source Gordon Schaffee
Differentiated Services (DiffServ)

- Intended to address the following difficulties with Intserv and RSVP:
  - **Scalability**: maintaining states by routers in high speed networks is difficult due to the very large number of flows.
  - **Flexible Service Models**: Intserv has only 2 classes, want to provide more qualitative service classes; want to provide ‘relative’ service distinction (Platinum, Gold, Silver, ...)
  - **Simpler signaling**: (than RSVP) many applications and users may only want to specify a more qualitative notion of service.
Architecture

- All policy decisions made at network boundaries
  - Boundary routers implement policy decisions by tagging packets with appropriate priority tag
  - Traffic policing at network boundaries

- No policy decisions within network
  - Routers within network forward packets according to their priority tags

source Gordon Schaffee
Differentiated Services Model

- **Edge routers**: traffic conditioning (policing, marking, dropping), SLA negotiation
  - Set values in DS-byte in IP header based upon negotiated service and observed traffic.
- **Interior routers**: traffic classification and forwarding (near stateless core!)
  - Use DS-byte as index into forwarding table
Diffserv Architecture

**Edge router:**
- per-flow traffic management
- marks packets as in-profile and out-profile

**Core router:**
- per class TM
- buffering and scheduling based on marking at edge
- preference given to in-profile packets
- Assured Forwarding
Scope of Service Class

- Packet priorities limited to an ISP
  - Extend with bilateral ISP agreements
- How can scope of priority be extended?
- Differentiated services is unidirectional

source Gordon Schaffee
Packet format support

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6: renamed as “DS”
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused
It may be desirable to limit traffic injection rate of some class; user declares traffic profile (e.g., rate and burst size); traffic is metered and shaped if non-conforming.
Per-hop Behavior (PHB)

- **PHB**: name for interior router data-plane functions
  - Includes scheduling, buff. mgmt, shaping etc

- **Logical spec**: PHB does not specify mechanisms to use to ensure performance behavior

- **Examples**:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B
PHB (contd)

- PHBs under consideration:
  - Expedited Forwarding (EF, premium): departure rate of packets from a class equals or exceeds a specified rate (logical link with a minimum guaranteed rate)
    - Emulates leased-line behavior
  - Assured Forwarding (AF): 4 classes, each guaranteed a minimum amount of bandwidth and buffering; each with three drop preference partitions
    - Emulates frame-relay behavior
Premium Service
Van Jacobson (LBL)

- Conservative allocation of resources
  - Provisioned according to peak capacity profiles
- Shaped at boundaries to remove bursts
- Out of profile packets dropped

- Defines a virtual leased line: fixed maximum bandwidth, but available when needed

source Gordon Schaffee
Premium Service Example

Drop always

Fixed Bandwidth

source Gordon Schaffee
AF PHB Group (RFC 2597)

- Provides forwarding of IP packets in four independent service classes
  - at each hop, each class has its own, configurable forwarding resources
- within each class, an IP packet is assigned one of three levels of drop precedence
  - lower drop precedence means higher probability of forwarding
- forwarding resources (buffer space and bandwidth) can be allocated using
  - FBA, CBQ, WFQ, priorities, etc.
- dropping of packets is based on the Random Early Drop (RED) algorithm
  - each level of drop precedence (green, yellow, red) has its own RED threshold

source Juha Heinänen
Example of Output Behavior

Each AF class has its own queue and forwarding resources

source Juha Heinänen
RED with Multiple Thresholds

Discard Probability

0 1

“Red” Threshold  “Yellow” Threshold  “Green” Threshold

Average Queue Length

source Juha Heinänen
Further readings

- See [http://www.ecse.rpi.edu/Homepages/shivkuma/research/cong-papers.html#qos](http://www.ecse.rpi.edu/Homepages/shivkuma/research/cong-papers.html#qos) for a list of papers on Scheduling, Buffer Management, Admission Control, Int-serv/Diff-serv and Network Calculus
Summary

- Best effort
- Differentiated services
- Integrated services

Axes:
- QoS (Quality of Service)
- Flow differentiation
- Complexity
Conclusions