

COMS3200

## COMS3200 – Week 11 Quality of Service

Some slides are taken from:  
"Computer Networking: A Top Down Approach Featuring  
the Internet",  
3rd edition.  
Jim Kurose, Keith Ross  
Addison-Wesley, July 2004.

School of Information Technology and Electrical Engineering  
The University of Queensland

COMS3200

## Outline

- QoS requirements for Multimedia Applications
- Transport protocols vs QoS
  - TCP, UDP
  - RTP
- Techniques to achieve QoS
- QoS in ATM networks
- QoS models for the Internet
  - Integrated Services
  - Differentiated Services
  - MultiProtocol Label Switching

2

COMS3200

## Learning objectives

After this week, you should

- understand how requested QoS may be supported at various layers of protocols
- be able to describe and compare IntServ, DiffServ and MPLS models
- be able to briefly describe functionality of RTP and RSVP protocols

3

COMS3200

## Recap: QoS Parameters

- Data rate (bandwidth)
- Delay (latency)
- Jitter (Delay variation)
- Reliability
  - Bit error rate
  - Packet error rate

4

COMS3200

## QoS Requirements of Applications

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

What about Network Games, Instant Messaging?


T-405 5

COMS3200

## QoS in today's Internet

**Internet is built on IP: only "best-effort service"**

- *no* guarantees on delay, loss, etc.



But we said multimedia apps require QoS and level of performance to be effective!

Today's Internet multimedia applications use application-level techniques to mitigate (as best as possible) effects of delay, loss

There are also efforts to introduce QoS for IP networks.

6

**COMS3200**

## QoS in today's Internet

Most common solution to Internet QoS problems:

- Bandwidth – network overprovisioning
- Delay - network overprovisioning, packet loss recovery for late packets
- Jitter – application level buffering to remove/minimise jitter
- Recovery from packet loss
  - Retransmit if time permits
  - Conceal error, interpolate data
  - Forward Error Correction (FEC)

7

**COMS3200**

## Recovery from packet loss: FEC

**Forward error correction (FEC): simple scheme**

- for every group of n chunks create a redundant chunk by XOR-ing the n original chunks
- send out n+1 chunks, increasing the bandwidth by factor 1/n.
- can reconstruct the original n chunks if there is at most one lost chunk from the n+1 chunks ("parity block")
- Playout delay needs to be fixed to the time to receive all n+1 packets
- **Tradeoff:**
  - increase n, less bandwidth waste
  - increase n, longer playout delay
  - increase n, higher probability that 2 or more chunks will be lost

8

**COMS3200**

## How should the Internet evolve to better support QoS?

**Laissez-faire**


- no major changes
- more bandwidth when needed
- Application-layer solutions
  - content distribution networks
  - application-layer multicast

**Integrated services philosophy:**

- Fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- Requires new, complex software in hosts & routers

**Differentiated services philosophy:**

- Fewer changes to Internet infrastructure, yet provide 1st and 2nd class service.



What's your opinion?

9

**COMS3200**

## Streaming Multimedia: UDP or TCP?

**UDP**

- server sends at rate appropriate for client (oblivious to network congestion!)
  - often send rate = encoding rate = constant rate
  - then, receiving rate = constant rate - packet loss
- short playout delay of 2-5 seconds (buffer) to compensate for network delay jitter
- error recover: time permitting

**TCP**

- send at maximum possible rate under TCP
- Packet arrival rate fluctuates due to TCP congestion control
- larger playout delay: smooth TCP delivery rate
  - OK for non-interactive real time applications
- HTTP/TCP passes more easily through firewalls

10

**COMS3200**

## Real-Time Protocol (RTP)

- RTP specifies a packet structure for packets carrying audio and video data
- RFC 1889
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - timestamping

- RTP runs in the end systems
- RTP packets are encapsulated in UDP segments
- Interoperability: If two Internet phone applications run RTP, then they may be able to work together

11

**COMS3200**

## RTP runs on top of UDP

RTP libraries provide a transport-layer interface that extends UDP:

- port numbers, IP addresses (UDP)
- payload type identification
- packet sequence numbering
- time-stamping

transport layer

}

Application
RTP
UDP
IP
Data Link
Physical

Time-stamps are used for

- Timing (delay, jitter)
- Synchronisation of multiple streams

12

## RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment
- RTP header indicates type of audio encoding in each packet
  - sender can change encoding during a conference
- RTP header also contains sequence numbers and timestamps

13

## RTP and QoS

- RTP does **not** provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

14

## RTP Header

Payload Type	Sequence Number	Timestamp	Synchronization Source Identifier	Miscellaneous Fields
--------------	-----------------	-----------	-----------------------------------	----------------------

**RTP Header**

- **Payload Type (7 bits):** Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs the receiver through this payload type field
  - Payload type 0: PCM mu-law, 64 kbps
  - Payload type 3: GSM, 13 kbps
  - Payload type 7: LPC, 2.4 kbps
  - Payload type 26: Motion JPEG
  - Payload type 31: H.261
  - Payload type 33: MPEG2 video
- **Sequence Number (16 bits):** Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence

15

## RTP Header (2)

- **Timestamp field (32 bytes long).** Reflects the sampling instant of the first byte in the RTP data packet
  - For audio, timestamp clock typically increments by one for each sampling period (for example, each 125 µsecs for a 8 KHz sampling clock)
  - if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.
- **SSRC field (32 bits long).** Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC. Supports multiplexing many streams onto one stream of UDP packets.

16

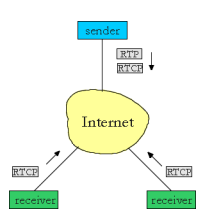
## Real-Time Control Protocol (RTCP)

- Works in conjunction with RTP
- Each participant in RTP session periodically transmits RTCP control packets to all other participants
- Each RTCP packet contains sender and/or receiver reports
  - report statistics useful to applications
- Statistics include number of packets sent, number of packets lost, interarrival jitter, etc.
- Feedback can be used to control performance
  - Sender may modify its transmissions based on feedback (e.g. encoding, data rate)

17

## RTCP - Continued

- For an RTP session there is typically a single multicast address; all RTP and RTCP packets belonging to the session use the multicast address.
- RTP and RTCP packets are distinguished from each other through the use of distinct port numbers. (RTCP port = RTP port +1)
- To limit traffic, each participant reduces his RTCP traffic as the number of conference participants increases.



18

COMS3200

## RTCP Packets

**Receiver report packets:**

- Fraction of packets lost, last sequence number, average interarrival jitter.

**Sender report packets:**

- SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.

19

COMS3200

## RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.
- The 75 kbps is equally shared among receivers:
  - With R receivers, each receiver gets to send RTCP traffic at  $75/R$  kbps.
- Sender gets to send RTCP traffic at 25 kbps.
- Participant determines RTCP packet transmission period by calculating avg RTCP packet size (across the entire session) and dividing by allocated rate.

**Example**

- Suppose one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
- RTCP gives 75% of this rate to the receivers; remaining 25% to the sender

20

COMS3200

## Approaches to QoS

- Overprovisioning (large network capacity)
- Resource reservation for streams/flows (IntServ)
  - reserving bandwidth, buffer space, CPU cycles for a flow (over a particular path)
  - Line in telephone network
- Service classes (DiffServ)
  - Prioritize certain traffic classes
- Traffic Engineering (MPLS)
  - Apply fast packet switching
  - Design and manage network so that there are always enough resources

21

COMS3200

## Techniques for achieving QoS

- Buffering (delayed playback)
- Traffic shaping at the source
  - regulating average rate
  - regulating burstiness of packets
- Traffic policing
  - User specifies required QoS
  - Network monitors the sender whether it follows the specification
- Packet scheduling in routers
- Admission control
  - Sender specifies required QoS
  - Network checks the whole path (routers or switches) whether this QoS can be provided

22

COMS3200

## Traffic Shaping: Leaky Bucket

- Leaky bucket – enforcing regular data rate

T-5.32

COMS3200

## Traffic Shaping: Token Bucket

- Token bucket – limiting burstiness
  - bucket can hold  $b$  tokens
  - tokens generated at rate  $r$  token/sec unless bucket full
  - Allows to save up tokens → short traffic bursts

24

**Scheduling And Policing Mechanisms**

- **scheduling**: choose next packet to send on link
- **FIFO (first in first out) scheduling**: send in order of arrival to queue
  - real-world example?
  - **discard policy**: if packet arrives to full queue: which to discard?
    - tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly

25

**Scheduling Policies: more**

**Priority scheduling**: transmit highest priority queued packet

- multiple *classes*, with different priorities
  - Class based Queuing (CBQ)
  - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..
  - E.g in home network, define ssh traffic class and FTP traffic class, with ssh having higher priority ( )

26

**Scheduling Policies: still more**

**round robin scheduling**:

- multiple classes
- cyclically scan class queues, serving one from each class (if available)
- Isn't this the same as having just one class?

27

**Scheduling Policies: still more**

**Weighted Fair Queuing**:

- generalized Round Robin
- each class gets weighted amount of service in each cycle

28

**Policing Mechanisms**

**Goal**: limit traffic to not exceed declared parameters

Three common-used criteria:

- **(Long term) Average Rate**: how many pkts can be sent per unit time (in the long run)
  - crucial question: what is the interval length: 100 packets per sec or 6000 packets per min have same average!
- **Peak Rate**: e.g., 6000 pkts per min. (ppm) avg.; 1500 pps peak rate
- **(Max.) Burst Size**: max. number of pkts sent consecutively (with no intervening idle)

29

**Policing Mechanisms**

**Token Bucket**: limit input to specified Burst Size and Average Rate.

- bucket can hold  $b$  tokens
- tokens generated at rate  $r$  token/sec unless bucket full
- **over interval of length  $t$ : number of packets admitted less than or equal to  $(r t + b)$ .**

30

**COMS3200**

## QoS in ATM networks

- Asynchronous Transfer Mode (ATM) network was the first technology which
  - can guarantee QoS (and therefore easily integrate various kinds of traffic)
  - can be used for LANs, MANs, WANs
- ATM is complex and therefore difficult to manage
  - it is slowly disappearing
  - used in the past in WANs and as backbone for LANs
  - Currently mostly used on long distance telecommunication links (but being slowly replaced by IP with MPLS – MPLS will be discussed later)

31

**COMS3200**

## QoS in ATM networks

- ATM is switch based
- ATM is connection oriented
  - Connection request goes through all switches on the path from the source to the destination
- Why described – the first Internet QoS model (IntServ) borrowed most of the ATM concepts

32

**COMS3200**

## QoS in ATM networks

- ATM traffic types
  - constant bit rate (CBR)
  - variable bit rate - real-time (VBR-rt)
  - variable bit rate - non real-time (VBR-nrt)
  - available bit rate (ABR)
- VBR is described by average rate, peak rate and max cell burst length
- For CBR: average rate = peak rate

33

**COMS3200**

## QoS in ATM networks

QoS guarantees based on

- User specification of required QoS
  - average data rate
  - peak rate
  - peak rate burst length
- Admission control
  - checking capacity of switches during connection establishment
- Reservation of resources for connections
- Traffic policing (token bucket algorithm)

34

**COMS3200**

## Traffic policing in ATM

CLP – Priority, packets with CLP=1 may be discarded by switches

35

**COMS3200**

## IETF Integrated Services

- architecture for providing QoS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info of allocated resources, QoS req's
- admit/deny new call setup requests:

**Question:** can newly arriving flow be admitted with performance guarantees while not violating QoS guarantees made to already admitted flows?

36

**Intserv: QoS guarantee scenario**

- **Resource reservation**
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control
- QoS-sensitive scheduling (e.g., WFQ)

37

**Intserv QoS: Service models [RFC2211, RFC 2212]**

- **Guaranteed service:**
  - worst case traffic arrival: leaky-bucket-policed source
  - simple (mathematically provable) *bound* on delay [Parekh 1992, Cruz 1988]
- **Controlled load service:**
  - "a quality of service closely approximating the QoS that same flow would receive from an unloaded network element."

$D_{max} = b/R$  (max queuing delay) 38

**Call Admission**

Arriving session must :

- declare its QoS requirement
  - **R-spec:** defines the QoS being requested
    - controlled-load
    - guaranteed: delay target
- characterize traffic it will send into network
  - **T-spec:** defines traffic characteristics

39

**TSpec**

- Example of flows with fixed and variable bandwidth requirements

40

**Specification of required QoS**

- **TSpec:** describes flow's traffic characteristics
  - average bandwidth + burstiness of the flow: *token bucket* filter
  - token rate  $r$
  - bucket depth  $B$  (max burst)
  - must have a token to send a byte
  - must have  $n$  tokens to send  $n$  bytes
  - start with no tokens
  - accumulate tokens at rate of  $r$  per second
  - can accumulate no more than  $B$  tokens

41

**Admission Control**

- Per-router mechanism
  - decides if a new flow can be supported
  - looks at TSpec and RSpec and checks whether router has capacity to guarantee QoS
  - decision may be *policy* based

Checking whether network will not go beyond its capacity if new flows added

42

COMS3200

## Policing

- User Policing
  - per-packet checks whether traffic source conforms to Tspec

Checking whether users do not violate their contracts with network

43

COMS3200

## Packet Processing

- *classification*: associate each packet with the appropriate reservation
- *scheduling*: manage queues so each packet receives the requested service

Appropriate packet queuing and processing in switches and routers

44

COMS3200

## Resource Reservation Protocol (RSVP)

- Protocol associated with *IntServ*
- Network layer protocol
- Designed to work with IPv4 and IPv6
- Reserves network resources necessary to provide required QoS
  - Does not define how routers reserve resources (e.g. scheduling)
- Designed to support multicast

45

COMS3200

## RSVP

- Basic concepts:
  - Resources are reserved for flows, ie. traffic streams
  - RSVP packet carries a flowspec, ie., required QoS
  - RSVP does not understand flowspecs, it is up to routers to interpret them
  - Flow is identified by destination IP address (and optionally destination port)

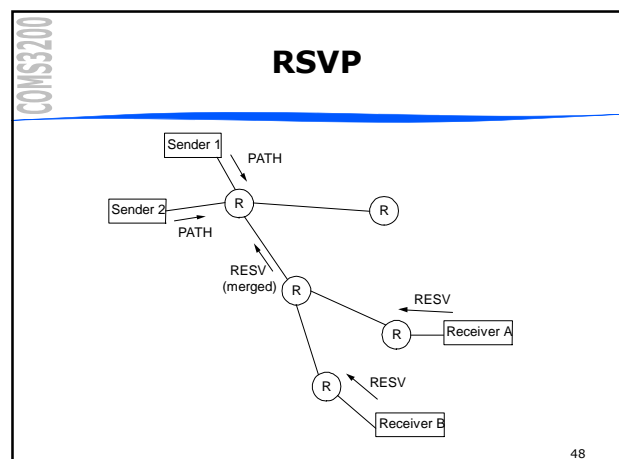
46

COMS3200

## RSVP

- Receiver-oriented
- Two messages: PATH and RESV
- Source periodically transmits PATH message
- Destination responds with RESV message
  - Resource reservations are made by routers along a path specified by the sender
  - Reservations from receivers of a multicast stream can be merged

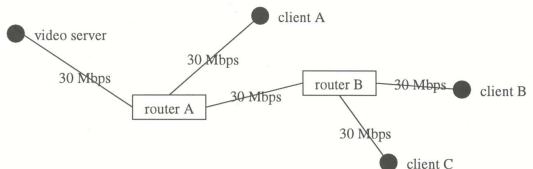
47



**COMS3200**

## RSVP

- Typical scenario for a client joining a video session:
  - Join multicast group
  - Reserve resources (streams can be merged)
  - Can also support heterogeneous receivers



50

**COMS3200**

## RSVP

- “Soft state” protocol
  - Sender has to refresh the path periodically
  - Receivers have to refresh reservations periodically
  - → Robustness and survivability
- RSVP summary:
  - Can be used to reserve resources for IP packets to provide required QoS
  - Can be used together with RTP

50

**COMS3200**

## RSVP versus ATM

- RSVP
  - receiver generates reservation
  - soft state (refresh/timeout)
  - separate from route establishment
  - QoS can change dynamically
  - receiver heterogeneity

51

**COMS3200**

## RSVP versus ATM

- ATM
  - sender generates connection request
  - hard state (explicit delete)
  - concurrent with route establishment
  - QoS is static for life of connection
  - uniform QoS to all receivers (if one stream sent to many receivers)

52

**COMS3200**

## Differentiated Services

- Problem with IntServ: scalability
  - all routers have to store per-flow state
    - Problem for backbone routers
- All routers have to understand RSVP
- DiffServ: class based approach
- DiffServ defines a set of classes with corresponding forwarding rules

53

**COMS3200**

## DiffServ

- Simple traffic differentiation (redefined TOS field in IPv4 header)- traffic classes
- Avoids per-flow, per-user state
- Marked packets receive a per-hop behaviour (PHB) along the path (implemented using buffer management and packet scheduling)
- Scalable (does not depend on hop-by-hop application signaling)

54

**Differentiated Services**

- Different PHBs
  - expedited forwarding
  - assured forwarding

55

**Differentiated Services – expedited forwarding**

Expedited forwarding mechanism

- Two classes of service
  - Regular
  - Expedited
- Each router has two queues
- Packet scheduling: weighted fair queueing
  - e.g if 10% traffic is expedited assign 20% of bandwidth for expedited and 80% for regular
  - expedited will have low delay

56

**Differentiated Services – expedited forwarding**

57

**Differentiated Services – assured forwarding**

Assured forwarding mechanism

- Four classes of service (defined in TOS)
- Three discard probabilities if congestion happens (low, medium, high)
- Classes are allocated by sender or edge (ingress) router
- Packets are passed through shaper/dropper filter

58

**Differentiated Services – assured forwarding**

59

**MPLS**

- Multiprotocol Label Switching (MPLS)
  - Approach created by router vendors (not IETF)
- Uses concept from virtual-circuit networks
  - Fixed-size label (local connection id) used for switching
- Allows fast switching (forwarding)
- Requires MPLS enabled routers

60

COMS3200

## MPLS

- Is independent from IP
  - uses its own header
- A label is an index into a forwarding table
- Labels change on each hop (similarly to connection identifiers in connection-oriented packet switching (VC))
- Table entry includes
  - Outgoing line
  - New label
- MPLS allows Traffic Engineering → QoS
  - IP routing can be overridden

61

COMS3200

## MPLS

- Simple traffic engineering
  - e.g. if there are two paths to destination, it is possible to send some packets along one path and others along another (by attaching appropriate labels)
- MPLS uses classes of service
- MPLS can be used to implement DiffServ
  - PHB determined from label value
- MPLS can be used to implement VPNs
  - Provides isolation of resources and addressing between VPNs

62

COMS3200

## MPLS

- Differences to VCs:
  - Many flows can be aggregated in one label (**FEC- Forwarding Equivalence Class**)
  - Forwarding tables are created differently than in VCs (**data driven** or **control driven** approach)

63

COMS3200

## MPLS

Example: TCP over IP and PPP

64

COMS3200

## Summary

- In the past the Internet provided **best effort** communication services
- *Real-time* apps require **guaranteed QoS**
- Techniques for achieving QoS
- Internet models for network layer support: IntServ, DiffServ, MPLS

65

COMS3200

## Readings

- Tanenbaum: 5.4, 5.6.5, 6.4.3
- (4<sup>th</sup> ed 5.3 (5.3.1, 5.3.2), 5.4, 6.4.3)
- Next week – Network Security
  - parts of Chapter 8
  - 8(intro), 8.1-8.1.4, 8.2 (8.2.1, 8.2.2), 8.3, 8.4 (8.4.1, 8.4.2, 8.4.3 (MD5)), 8.5, 8.6 (8.6.1, 8.6.2, 8.6.3), 8.7 (8.7.1, 8.7.5), 8.9.3
- (4<sup>th</sup> ed: Same sections)

66