Outline

- Multimedia Networking Applications
- Streaming stored audio and video
- Real-time Multimedia: Internet phone study
- Protocols for Real-Time interactive applications
  - RTP, RTCP, SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services and Differentiated Services
- RSVP

1

Real-time Transport 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
  - Time stamping
- 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

2

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.

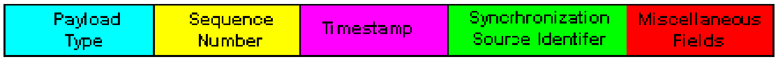
3

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.

4

RTP Header



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.

5

RTP Header (2)

- Timestamp field (32 bits 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 usecs 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.

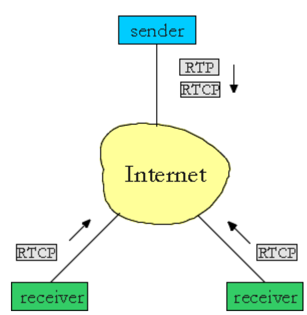
6

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 application
- 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

7

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.
- To limit traffic, each participant reduces his RTCP traffic as the number of conference participants increases.

8

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.

Source description packets:

- e-mail address of sender, sender's name, SSRC of associated RTP stream.
- Provide mapping between the SSRC and the user/host name.

9

Synchronization Of Streams

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - Not tied to the wall-clock time
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - Timestamp of the RTP packet
  - Wall-clock time for when packet was created.
- Receivers can use this association to synchronize the playout of audio and video.

10

RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.

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
- 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.

11

SIP

- Session Initiation Protocol
- Comes from IETF

SIP long-term vision

- All telephone calls and video conference calls take place over the Internet
- People are identified by names or e-mail addresses, rather than by phone numbers.
- You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using.

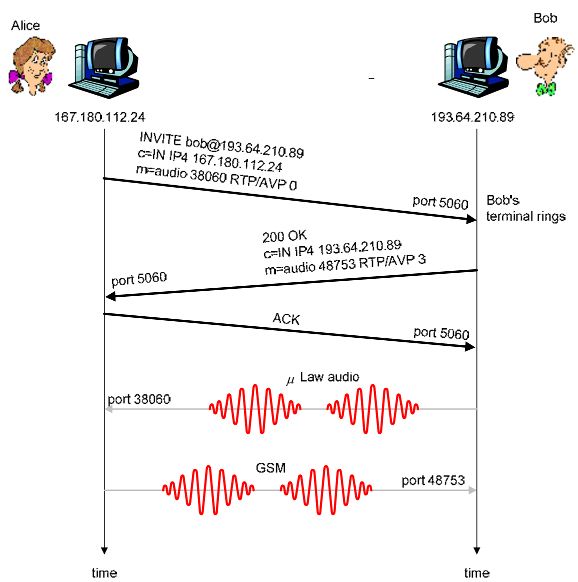
12

SIP Services

- Setting up a call
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms to end call.
- Determine current IP address of callee.
  - Maps mnemonic identifier to current IP address
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

13

Setting Up a Call To a Known IP Address



- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address & preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- Default SIP port number is 5060.

14

Setting Up a Call (more)

- Codec negotiation:
  - Suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply and list encoders he can use.
  - Alice can then send a new INVITE message, advertising an appropriate encoder.
- Rejecting the call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden".
- Media can be sent over RTP or some other protocol.

15

Example of SIP Message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:
- HTTP message syntax
- sdp = session description protocol
- Call-ID is unique for every call.

- Here we don't know Bob's IP address. Intermediate SIP servers will be necessary.
- Alice sends and receives SIP messages using the SIP default port number 506.
- Alice specifies in Via: header that SIP client sends and receives SIP messages over UDP

16

Name Translation And User Location

- Caller wants to call callee, but only has callee's name or e-mail address.
- Need to get IP address of callee's current host:
  - User moves around
  - DHCP protocol
  - User has different IP devices (PC, PDA, car device)
- Result can be based on:
  - Time of day (work, home)
  - Caller (don't want boss to call you at home)
  - Status of callee (calls sent to voicemail when callee is already talking to someone)

Service provided by SIP servers:

- SIP registrar server
- SIP proxy server

17

SIP Registrar

- When Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server
  (similar function needed by Instant Messaging)

Register Message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

18

SIP Proxy

- Alice sends invite message to her proxy server
  - Contains address sip:bob@domain.com
- Proxy responsible for routing SIP messages to callee
  - Possibly through multiple proxies.
- Callee sends response back through the same set of proxies.
- Proxy returns SIP response message to Alice
  - Contains Bob's IP address

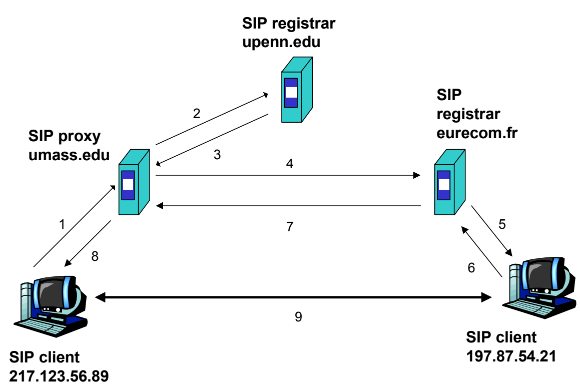
- Note: proxy is analogous to local DNS server

19

Example

Caller jim@umass.edu with places a call to keith@upenn.edu

(1) Jim sends INVITE message to umass SIP proxy. (2) Proxy forwards request to upenn registrar server. (3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr



(4) umass proxy sends INVITE to eurecom registrar. (5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client. (6-8) SIP response sent back (9) media sent directly between clients.
Note: also a SIP ack message, which is not shown.

20

Comparison with H.323

- H.323 is another signaling protocol for real-time, interactive
- H.323 is a complete, vertically integrated suite of protocols for multimedia conferencing: signaling, registration, admission control, transport and codecs.
- SIP is a single component. Works with RTP, but does not mandate it. Can be combined with other protocols and services.
- H.323 comes from the ITU (telephony).
- SIP comes from IETF: Borrows much of its concepts from HTTP. SIP has a Web flavor, whereas H.323 has a telephony flavor.
- SIP uses the KISS principle: Keep it simple stupid.

Outline

- Multimedia networking applications
- Streaming stored audio and video
- Real-time Multimedia: Internet phone study
- Protocols for Real-Time interactive applications
  - RTP, RTCP, SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services and Differentiated Services
- RSVP

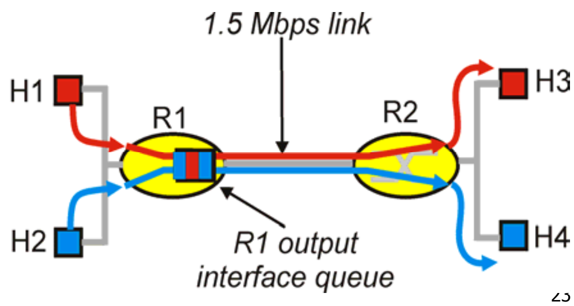
Improving QOS in IP Networks

Thus far: "making the best of best effort"

Future: next generation Internet with QoS guarantees

- RSVP: signaling for resource reservations
- Differentiated Services: differential guarantees
- Integrated Services: firm guarantees

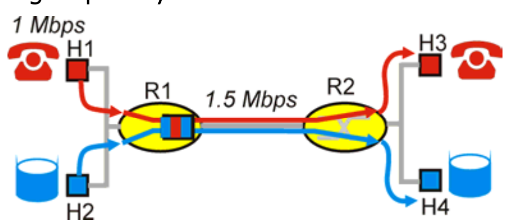
- Simple model for sharing and congestion studies:



23

Principles For QOS Guarantees

- Example: 1MbpsI P phone, FTP share 1.5 Mbps link.
  - bursts of FTP can congest router, cause audio loss
  - want to give priority to audio over FTP



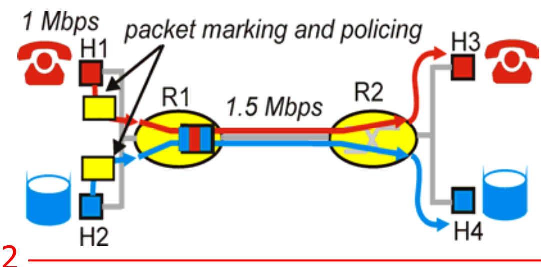
Principle 1

Packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly.

24

Principles For QOS Guarantees (more)

- What if applications misbehave (audio sends higher than declared rate)
  - Policing: force source adherence to bandwidth allocations
- Marking and policing at network edge:
  - Similar to ATM UNI (User Network Interface)

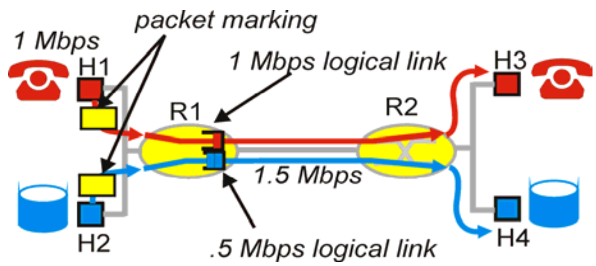


Principle 2
Provide protection (*isolation*) for one class from others

25

Principles For QOS Guarantees (more)

- Allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flows doesn't use its allocation

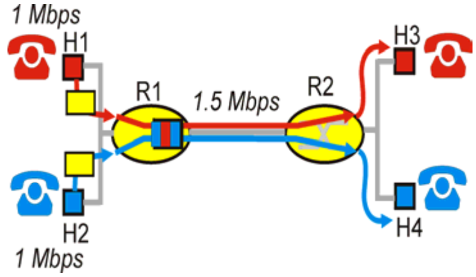


Principle 3
While providing isolation, it is desirable to use resources as efficiently as possible

26

Principles For QOS Guarantees (more)

- *Basic fact of life:* can not support traffic demands beyond link capacity

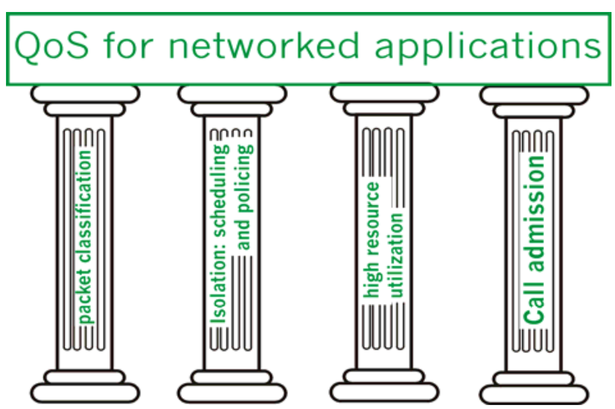


Principle 4

Call Admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

27

Summary of QoS principles



Let's next look at mechanisms for achieving this ….

28

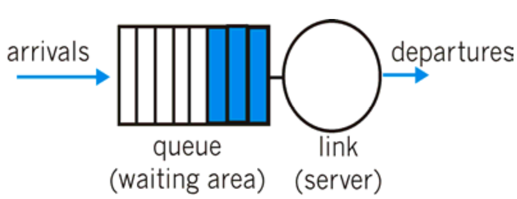
Outline

- Multimedia networking applications
- Streaming stored audio and video
- Real-time Multimedia: Internet phone study
- Protocols for Real-Time interactive applications
  - RTP, RTCP, SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services and Differentiated Services
- RSVP

29

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: who to discard?
    - Tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly

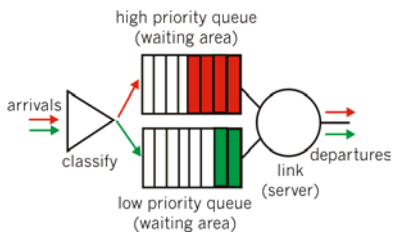


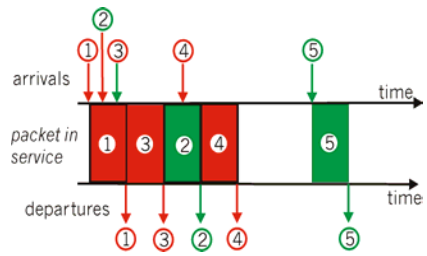
30

Scheduling Policies: More

Priority scheduling: transmit highest priority queued packet

- Multiple *classes*, with different priorities
  - Class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..
  - Real world example?



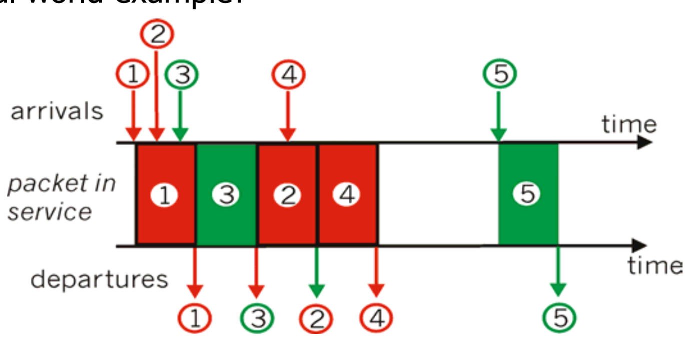


31

Scheduling Policies: Still More

Round robin scheduling:

- Multiple classes
- Cyclically scan class queues, serving one from each class (if available)
- Real world example?

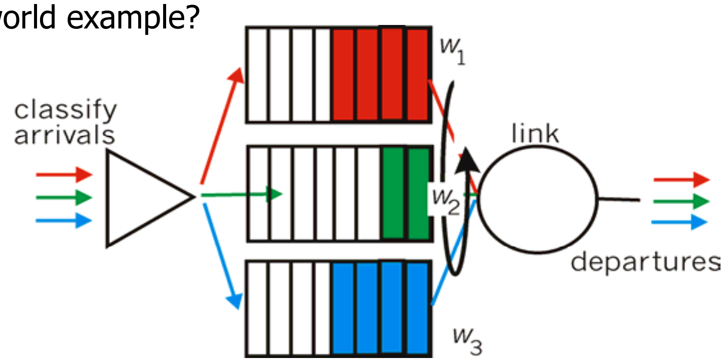


32

Scheduling Policies: Still More

Weighted Fair Queuing:

- Generalized Round Robin
- Each class gets weighted amount of service in each cycle
- Real-world example?



33

Policing Mechanisms

Goal: Limit traffic to not exceed declared parameters

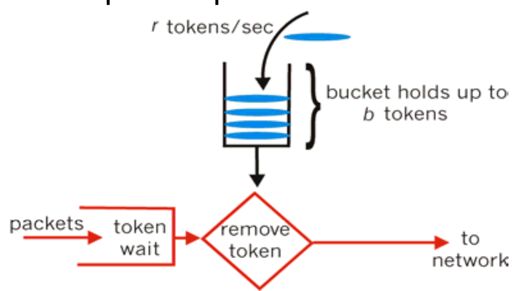
Three common-used criteria:

- *(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 ppm peak rate
- *(Max.) Burst Size:* max. number of pkts sent consecutively (with no intervening idle)

34

Policing Mechanisms

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

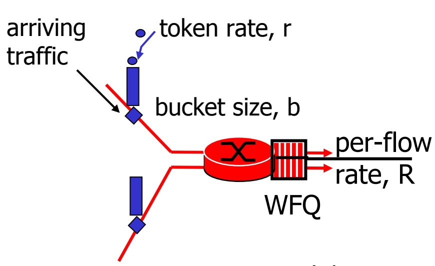


- Bucket can hold b tokens
- Tokens generated at rate *r token/sec* unless bucket full
- *Over interval of length t: number of packets admitted less than or equal to (r t + b).*

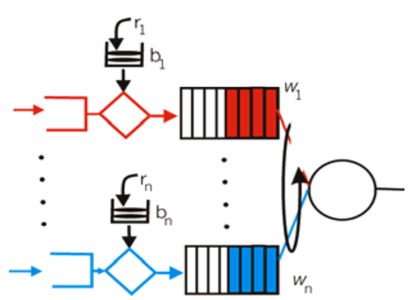
35

Policing Mechanisms (more)

- Token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee*!



$D_{max} = b/R$



36

Outline

- Multimedia networking applications
- Streaming stored audio and video
- Real-time Multimedia: Internet phone study
- Protocols for Real-Time interactive applications
  - RTP, RTCP, SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services and Differentiated Services
- RSVP

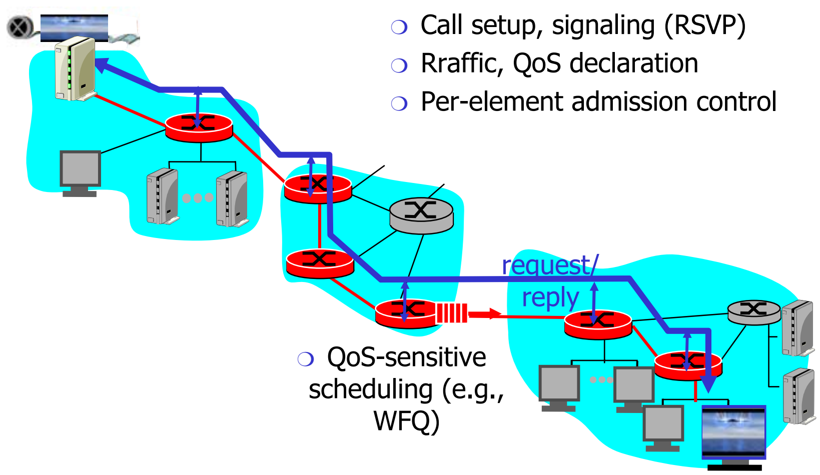
IETF Integrated Services

- Architecture for providing QOS guarantees in IP networks for individual application sessions
- Resource reservation: routers maintain state info (à la VC) of allocated resources, QoS req's
- Admit/deny new call setup requests:

Question: Can newly arriving flow be admitted with performance guarantees while not violated QoS guarantees made to already admitted flows?

Intserv: QoS Guarantee Scenario

- Resource reservation
  - Call setup, signaling (RSVP)
  - Rraffic, QoS declaration
  - Per-element admission control



- QoS-sensitive scheduling (e.g., WFQ)

39

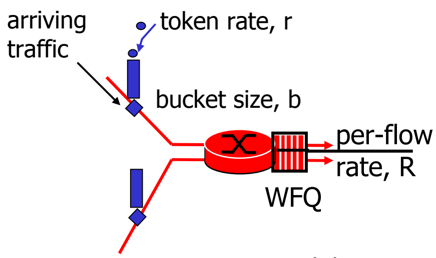
Intserv QoS: Service Models [rfc2211, rfc 2212]

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."



$D_{max} = b/R$

40

IETF Differentiated Services

Concerns with Intserv:

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows
- Flexible Service Models: Intserv has only two classes. Also want "qualitative" service classes
  - "Behaves like a wire"
  - Relative service distinction: Platinum, Gold, Silver

Diffserv approach:

- Simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don't define define service classes, provide functional components to build service classes

41

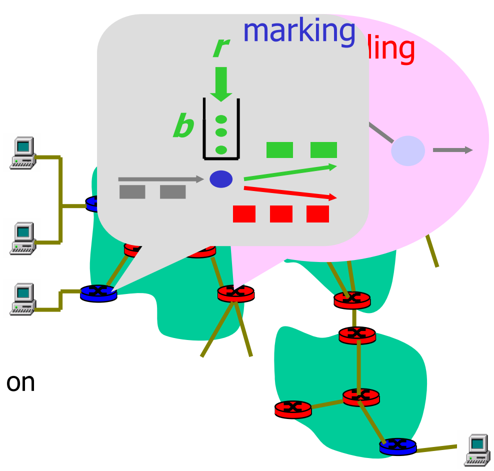
Diffserv Architecture

Edge router:

- Per-flow traffic management
- Marks packets as in-profile and out-profile

Core router:

- Per class traffic management
- Buffering and scheduling based on marking at edge
- Preference given to in-profile packets
- Assured Forwarding



42

Signaling in the Internet

connectionless (stateless) forwarding by IP routers + best effort service = no network signaling protocols in initial IP design

- New requirement: reserve resources along end-to-end path (end system, routers) for QoS for multimedia applications
- RSVP: Resource Reservation Protocol [RFC 2205]
  - " … allow users to communicate requirements to network in robust and efficient way." i.e., signaling !
- Earlier Internet Signaling protocol: ST-II [RFC 1819]

43

RSVP Design Goals

1. Accommodate heterogeneous receivers (different bandwidth along paths)
2. Accommodate different applications with different resource requirements
3. Make multicast a first class service, with adaptation to multicast group membership
4. Leverage existing multicast/unicast routing, with adaptation to changes in underlying unicast, multicast routes
5. Control protocol overhead to grow (at worst) linear in # receivers
6. Modular design for heterogeneous underlying technologies

44

RSVP: Does Not …

- Specify how resources are to be reserved
  - Rather: a mechanism for communicating needs
- Determine routes packets will take
  - That’s the job of routing protocols
  - Signaling decoupled from routing
- Interact with forwarding of packets
  - Separation of control (signaling) and data (forwarding) planes

45

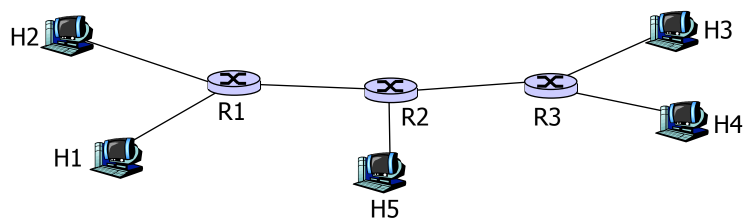
RSVP: Overview Of Operation

- Senders, receiver join a multicast group
  - Done outside of RSVP
  - Senders need not join group
- Sender-to-network signaling
  - *Path message:* make sender presence known to routers
  - Path teardown: delete sender’s path state from routers
- Receiver-to-network signaling
  - *Reservation message:* reserve resources from sender(s) to receiver
  - Reservation teardown: remove receiver reservations
- Network-to-end-system signaling
  - Path error
  - Reservation error

46

RSVP: Simple Audio Conference

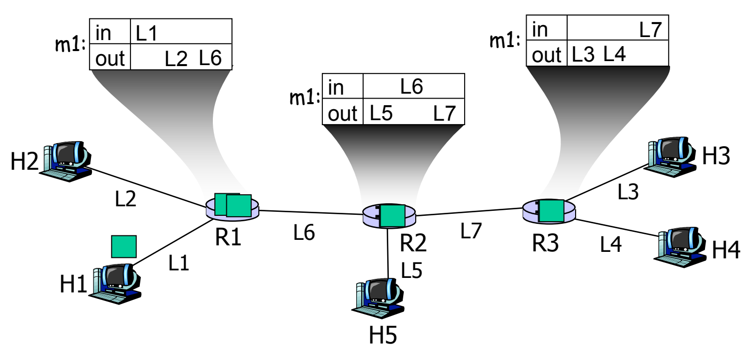
- H1, H2, H3, H4, H5 both senders and receivers
- Multicast group m1
- No filtering: packets from any sender forwarded
- Audio rate: *b*
- Only one multicast routing tree possible



47

RSVP: Building Up Path State

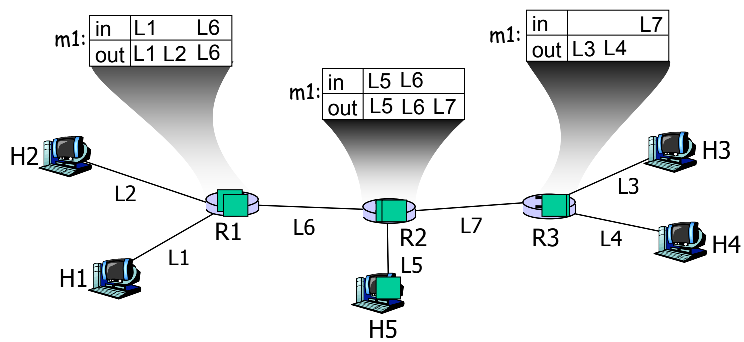
- H1, …, H5 all send path messages on *m1:*
  (address=*m1*, Tspec=*b*, filter-spec=no-filter,refresh=100)
- Suppose H1 sends first path message



48

RSVP: Building Up Path State

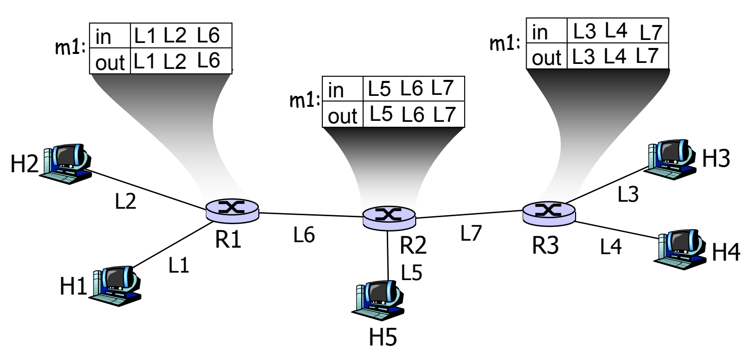
- Next, H5 sends path message, creating more state in routers



49

RSVP: Building Up Path State

- H2, H3, H5 send path msgs, completing path state tables

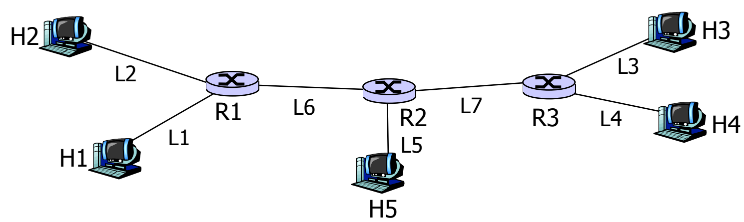


50

RSVP: *Receiver* Reservation Example 1

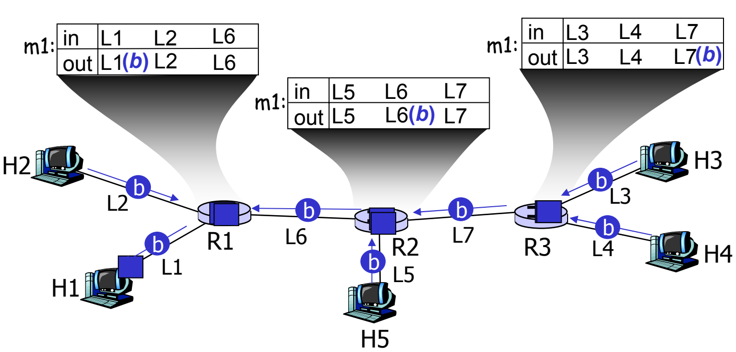
H1 wants to receive audio from all other senders

- ❒ H1 reservation msg flows uptree to sources
- ❒ H1 only reserves enough bandwidth for 1 audio stream
- ❒ Reservation is of type "no filter" – any sender can use reserved bandwidth



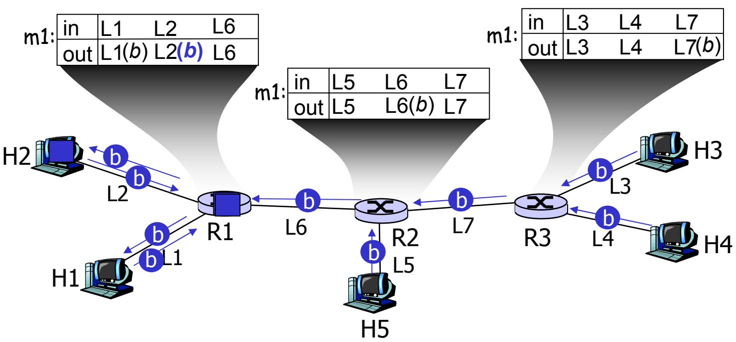
RSVP: *Receiver* Reservation Example 1 (2.)

- ❒ H1 reservation msgs flows uptree to sources
- ❒ Routers, hosts reserve bandwidth b needed on downstream links towards H1



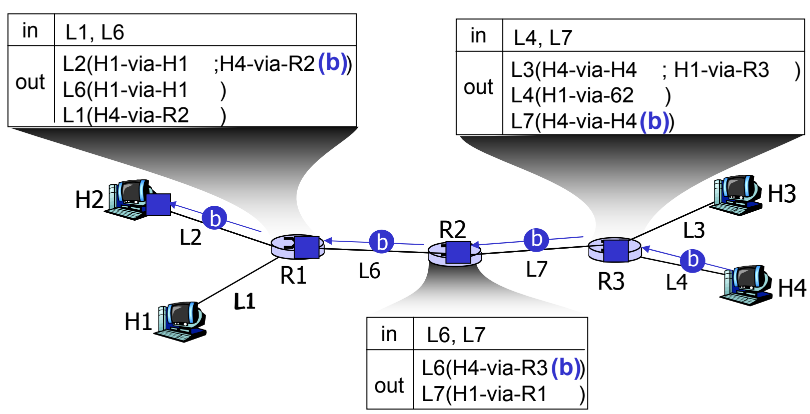
RSVP: *Receiver* Reservation Example 1 (3.)

- next, H2 makes no-filter reservation for bandwidth *b*
- H2 forwards to R1, R1 forwards to H1 and R2 (?)
- R2 takes no action, since *b* already reserved on L6



RSVP: *Soft-State*

- Senders periodically resend path msgs to refresh (maintain) state
- Receivers periodically resend resv msgs to refresh (maintain) state
- Path and resv msgs have TTL field, specifying refresh interval



Multimedia Networking: Summary

- Multimedia applications and requirements
- Making the best of today's best effort service
- Scheduling and policing mechanisms
- Next generation Internet: Intserv, RSVP, Diffserv