Integrated Services

CSC 790

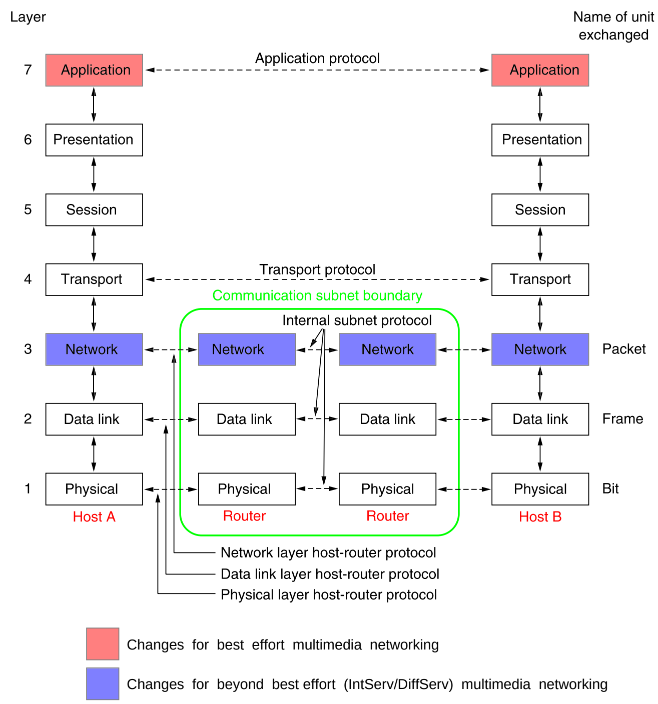
WAKE FOREST UNIVERSITY

Department of Computer Science

Fall 2009

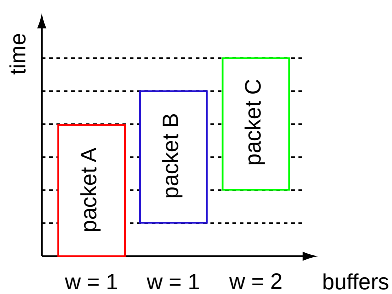
Internet QoS

- Two solutions for transmitting multimedia over the Internet
  1. Best effort multimedia
     - Sender/receiver compensate for network
     - Network is still best effort $\rightarrow$ no guarantees
  2. QoS enabled Internet (beyond best effort)
     - Changes must be made to the network architecture
     - Can provide guarantees
     - Should be compatible with current Internet
- Two proposed Internet QoS enhancements
  - Integrated Services (IntServ)
  - Differentiated Services (DiffServ)



GPS and PGPS

- Consider a system where three packets (A, B, C) arrive



- The service of the packets differ using GPS and PGPS
  - GPS service is (C, A, B)
  - PGPS service is (A, C, B)
- *PGPS serves packets based on current buffer contents, it cannot determine future arrivals*

PGPS Delay

- PGPS *may* have an additional delay of

$$\frac{p^{max}}{s}$$

where p^{max} is the maximum packet size and s is link speed

- Therefore the delay for a PGPS system is

$$\frac{p^{max}}{s \cdot \frac{w_i}{\sum w_j}} + \frac{p^{max}}{s}$$

- This can be expanded for end to end delay of n PGPS routers

$$d_i \leq \frac{c_i}{s \cdot \frac{w_i}{\sum w_k}} + \sum_{j=1}^{n-1} \frac{p_i^{max}}{s_j \cdot \frac{w_i}{\sum w_k}} + \sum_{j=1}^{n} \frac{p^{max}}{s_j}$$

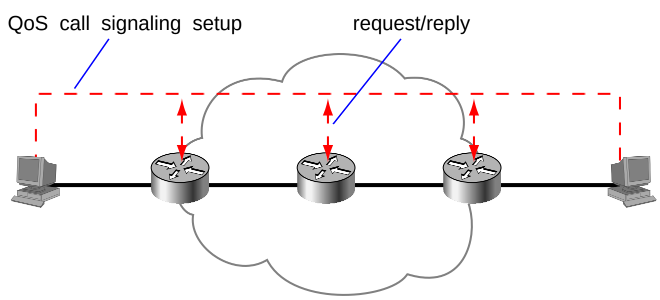
Internet Integrated Services

Integrated Services (IntServ) was developed by the IETF to provide individual QoS

- Uses many of the principles discussed in the previous lecture
  - Note this is a *beyond best-effort* approach
- Two key features are part of the IntServ architecture
  1. Reserved resources - A router is required to know what resource (bandwidth and buffer space) amounts are reserved for on-going sessions
  2. Call setup - A user requiring QoS must first be able to reserve sufficient resources at each router on the path (this requires the participation of each router on the path).

Call Admission

1. Traffic characterization and QoS specification
   - User must declare QoS (R_{spec}) and traffic (T_{spec})
2. Signaling for call setup
   - R_{spec} and T_{spec} sent to each router in path
3. Per element call admission
   - Once router receives R_{spec} and T_{spec} must accept or deny
   - If all the routers accept, then session can start



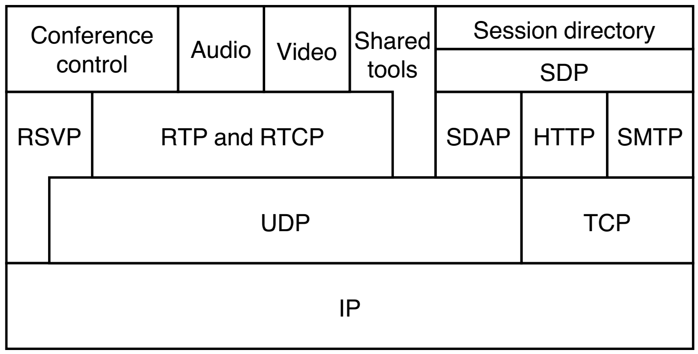
RSVP

Resource ReSerVation Protocol (RSVP) is an Internet signaling protocol for reserving bandwidth [RFC 2205] (Zhang 1993)

- Two important principal characteristics
  1. Provides reservations for bandwidth in multicast trees
     What about unicast?
  2. It is **receiver-oriented**, receiver of the data flow reserves
     Why is this appropriate?

  Aren't IP routes dynamic? Will the forward and reverse paths be the same? How is all this possible?

- What RSVP is **not**
  - Does not specify how the network reserves bandwidth
    What does this mean?
  - Not a routing protocol



RSVP Messages

The following are the *primary* messages used by RSVP (*using UDP*)

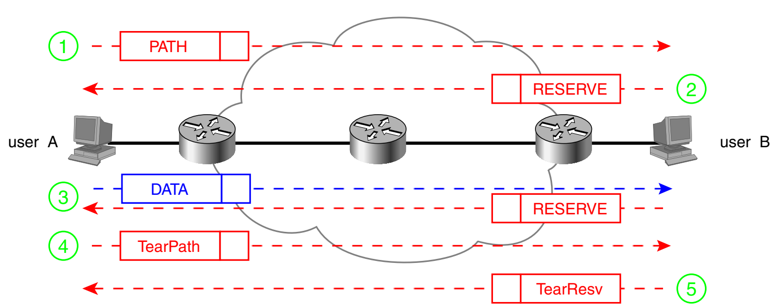
- **Path** - Originates from sender
  - Install reverse routing state in each router
  - Provide receiver information about sender and path
  - *Route pinning* - Gathering path information
- **Reserve** - Originates from receiver (*receiver-oriented reservations*)
  - Carries reservation request to routers and sender
- Tear-down
  - Informs routers to remove reservation for the session
  - *Actually two messages*, **PathTear** and **ResvTear**

Basic RSVP Operation

Assume user A wants to establish a session with B (A stream to B)

1. User A sends a **path** message to user B
   - Message contains T_{spec} and addresses of sender/receiver
     Why is T_{spec} important?
2. Routers on the path receive path message
   - Router stores (locally) sender info and previous router address
     Why is this important?
   - Reservation for this session is **not** made at this time
3. User B receives path message and sends **reserve** message to A
   - Based on the the path message receiver determines the amount to reserve

4. Routers receive reserve message
   - Reserves the necessary resources if available
   - Message forwarded to next router
5. User A receives reserve message and starts session
   - User B periodically sends reserve message to maintain reservation (soft-state, will expire)
6. Once session complete, users A and B send tear-down messages



RSVP Operation

- If resources are not available upon receiving reservation message
  - Router sends error message back to receiver
  - Receiver sends ResvTear message
- Routers maintain reservation per session, *soft-state*
  - Stores T_{spec}, next hop, filter, timeout
  - Each reservation will expire, receiver must refresh
  - *This will become a problem...*
- Reservation $\rightarrow$ reserve resources + packet filter
  - Reservation does **not** determine which packets can use resources, specifies what amount is reserved for whom (entity)
  - Filter determines which packets can use resource
  - There can be *no filter*, a *static filter*, or a *dynamic filter*

RSVP and Virtual Circuits

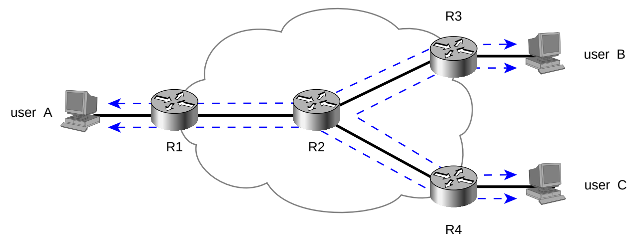
- RSVP has some similarities to establishing a VC
  - Bandwidth and buffer space reserved along a path
- However one major difference is routing is done by IP
  - As a result routes can change periodically
  - RSVP has been designed to *deal* these events
  - Changing routes is managed using soft-state
- If a route changes, signaling messages will take a new path
  - Messages refresh the reservations along the path
  - If the path changes, then new reservations are created

    What about the old reservations?

RSVP and Multicasting

We have only considered unicast RSVP, its strength is multicasting

- Multicasting - A special form of broadcast in which packets are delivered to a specific sub group of users
- Consider three users, all can send/receive audio



  - Only allocate enough bandwidth for one channel per link
  - Router will determine if enough resources are already reserved

How would this change if sending video?

Reservation Styles

- Reservation request can include options, called *reservation style*
  - Determine how requests merged, forwarded to upstream node
- **Wild-card-Filter** (WF) style
  - Implies shared reservation and wild-card sender
  - Receivers share single reservation, size is largest of requests
  - Represented as $WF(*, \{Q\})$ where $*$ represents the wild-card sender and Q is the flow spec
- **Filter-Fixed** (FF) style
  - Implies shared reservation and explicit sender selection
  - Reservation creates shared reservation and specifies senders
  - Represented as $FF(S_1(Q_1), S_2(Q_2), ...S_n(Q_n))$ where S_i is the sender and Q_n is the flow spec

- **Shared-Explicit** (SE)
  - Implies shared reservation and wild-card sender
  - Receivers share single reservation, size is largest of requests
  - Represented as $SE((S_1, S_2..., S_n), \{Q_n\})$ where S_i is the sender and Q_n is the flow spec
- WF and SE are designed for multicast applications
  - Unlikely all sources will transmit at once
    For example?
  - Often reserve twice the bandwidth of one user

Cost of Implementation

- Scheduling is approximately 10% overhead
- Consider a low-end 68040 based router
  - Processing time, 0.73 msec for PATH, 0.37 msec for RESV
  - Approximately 1,000 flow setups/sec
  - Processing of PATH (RESV) refresh: 0.33 msec (0.29 msec)
  - Approximate capacity is 1,600 flows
  - About 500 bytes/flow
- Refresh bandwidth 100 kb/s for 1000 flows (30 sec refresh)
- Message sizes, PATH is 208 bytes and RESV is 148 bytes

Concerns with RSVP

- RSVP provides many significant enhancements to the Internet
  - Not widely deployed
  - Primary concern is *scalability* (its a *fine grain* solution)
- Every RSVP flow that passes through a router has a reservation
  - Suppose every flow on an OC-48 (2.5 Gbps) link is a 64 Kbps audio stream
  - Total number of flows is $\frac{2.5 \times 10^9}{64 \times 10^3} = 39000$
  - Every reservation requires state at the router and is refreshed
  - *This is just for one link...*
- RSVP is acceptable for a small network, but another solution is needed for larger networks... DiffServ