RSVP and the Integrated Services Architecture for the Internet

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 20

"Roadmap" for Multimedia Networking

1. Introduction
   - why QoS?
   - what are the problems?

Lecture 17

2. Basic operations
   - jitter buffers (at hosts)
   - task scheduling (at hosts)
   - packet shaping (at hosts)
   - packet dropping (at routers)
   - packet scheduling (at routers)

Lecture 16

Lecture 18

Lecture 19

Today's Lecture

3. **Types of service**
   - **Integrated Services (IntServ) and Resource Reservation Protocol (RSVP)**
   - Differentiated Services (DiffServ)

"Roadmap" (cont'd)

4. Application-level feedback and control
   - Real-time Protocol (RTP), Real-time Control Protocol (RTCP)
   - Real-time Streaming Protocol (RTSP)
5. Application signaling and device control
   - Session Announcement Protocol (SAP)
   - Session Description Protocol (SDP)
   - Session Initiation Protocol (SIP)
   - Media Gateway Control Protocol (MGCP)
6. Routing
   - Multi-protocol Label Switching (MPLS)
   - multicasting

(Reminder of Problems, + Solutions)

- ✓ Less-than-ideal average delays and loss rates
- ➡ Variations in traffic loads in the network
- ✓ TCP's congestion control
- ✓ Retransmission-based error recovery
- 5. Simplistic routing algorithms
- ✓ "Burstiness" or variability of a single traffic source
  - ✓ peak rate, average rate, maximum burst size

- Schedule packet transmission carefully to control delays/losses
- Manage traffic loads through reservations and admission control (IntServ/RSVP)
- Improved packet dropping policies
- Use with jitter buffers, and to restore anchor frames of video
- …
- "Shape" traffic to reduce variability

Modes of Communication

- From one sender to one receiver: unicast
- From one sender to everybody(!): broadcasting
- From one sender to multiple, selected receivers: multicasting
- From multiple senders(?) to multiple receivers: multicasting
  - example: audio conferencing

Multicasting in the Internet

- Goal
  - more efficient communication than just using multiple unicast sessions
- Requirements
  - multicast IP addresses
  - ability of receivers to "join" a multicast group
  - distribution of duplicate data by the network, rather than by the sender
  - multicast routing
- (more later)

QoS Guarantees (Reminder)

- Deterministic (100%) guarantees
  - based on peak traffic rate
  - simple, predictable, conservative
  - Guaranteed Service
- Statistical (< 100%) guarantees
  - based on peak and mean traffic rates
  - complex, less predictable, higher utilization
  - Controlled Load Service
- No guarantees
  - the network performance is what it is
  - Best Effort Service

Incremental Deployment

- Some routers may not be RSVP/IntServ-enabled
  - incremental deployment and "backwards compatibility"
- Can QoS be backwards compatible?
  - "I'll use it if you use it, but if you don't use it, I won't use it"

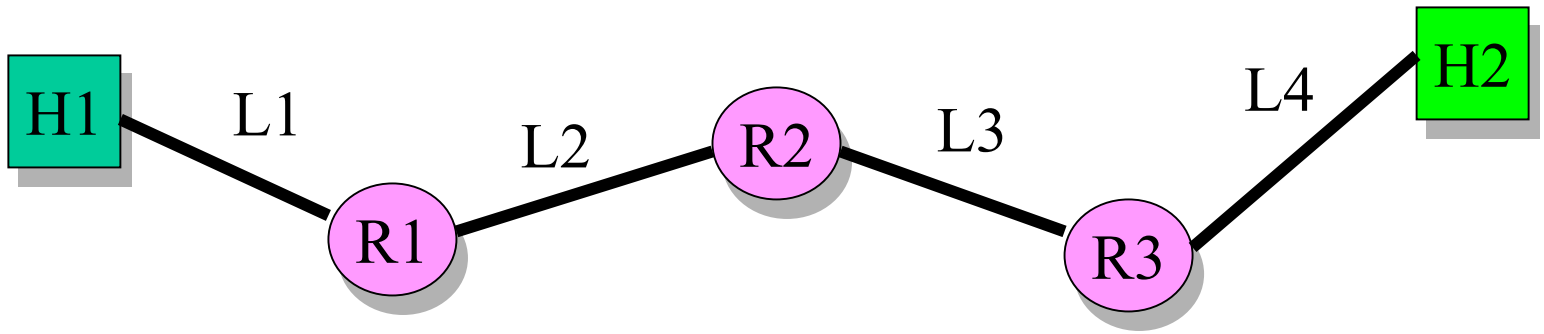
The RSVP Protocol

- Purpose: announce / signal...
  - the sending application requirements to receivers
  - the receivers' resource requirements to the network
- Senders announce their traffic characteristics and requirements: PATH messages
- Receivers initiate request for resources along the path: RESV messages
- Calculation of resource requirements or QoS is not within RSVP scope!
- RSVP is unidirectional
  - Reservations are established from sender to receiver

RSVP (cont'd)

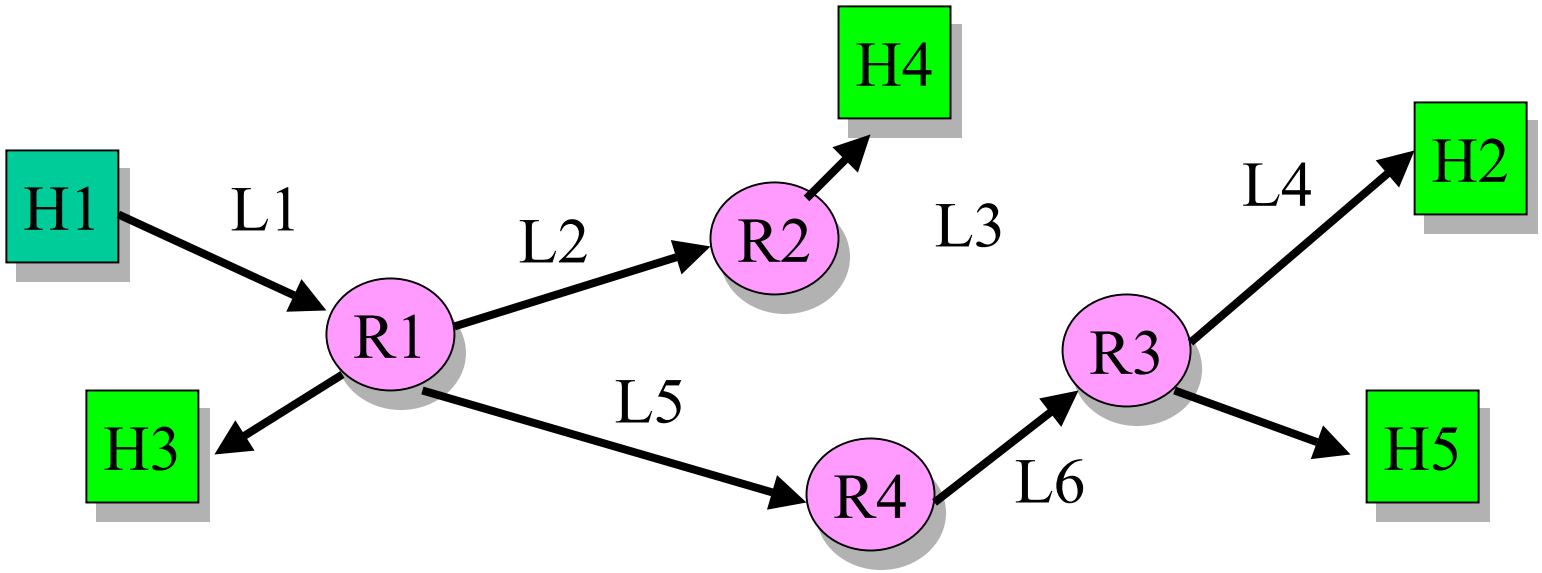
- Runs directly over IP (unreliable)
- RSVP is a hop-by-hop protocol
  - routers have to process the messages and possibly modify their contents
  - requires the IP "router alert" option to be specified

Issues in Resource Reservation
Point-to-Point (One-to-One) Communication



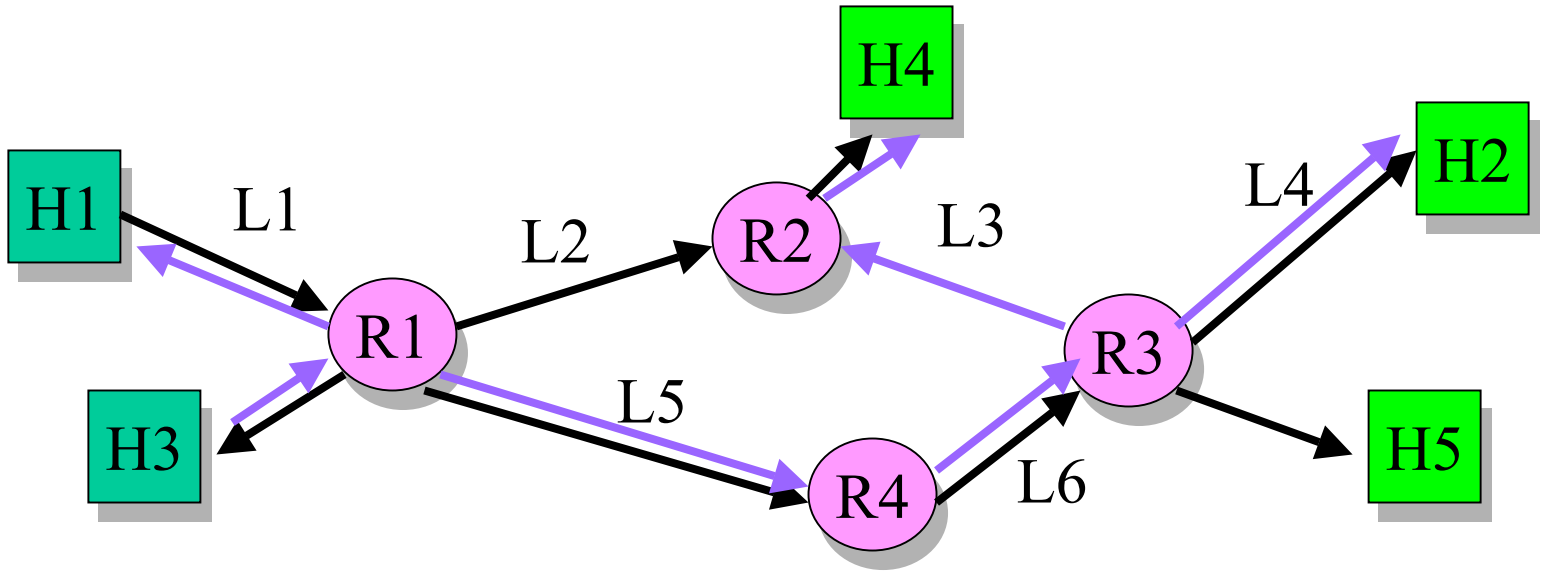
- Goal: establish a virtual circuit from H1 to H2
  - reserve "resources" in routers R1, R2, and R3.
- Resources are ….
  - link capacity on transmission links
  - buffer capacity in routers to hold packets in transit
  - CPU capacity at all routers to forward packets from H1 in real-time

Issues in Resource Reservation
One-to-Many (Multicast)



- Apply the point-to-point method recursively throughout the multicast tree?
- How do we handle differing link/router capacities?
  - reduce all receivers to the speed of the slowest link/receiver?
- How do we add new users and delete old ones, while the application is "in progress"?

Issues in Resource Reservation
Many-to-Many Multicast



- H1 and H3 independently reserve resources
- How can we utilize resources efficiently?
  - consider audio conference (not everyone talking at the same time)

RSVP Message Format

- RSVP Header
  - Version, Flags, Message Type, Checksum, TTL, Total Length
- RSVP Objects
  - Number and Type, Length, and Value
- Extensible; easy to add define new objects

RSVP Objects

- Session Object
  - destination IP address, port, and protocol type
  - note: destination IP address may be a multicast address (reaching multiple receivers)
- Hop Object
  - next hop in path, and/or previous hop in path
  - necessary to accomplish route "pinning" (later)

Objects (cont'd)

- Style Object
  - type of reservation merging (later)
- Filter Specification (Filterspec) Object
  - which sender (traffic source) this applies to
  - IP address and port of sender
  - Sender Template Object: same as Filterspec
- Sender Traffic Specifcation (Tspec) Object
  - descriptor of traffic characteristics of a flow
  - token rate, token bucket size
  - peak data rate
  - minimum policed unit (minimum packet size)
  - maximum packet size

Objects (cont'd)

- Flow Specification (Flowspec) Object
  - receiver's requirements for a flow
  - all the Tspec parameters
  - additional: rate required and Slack (later)
- Advertisement Specification (Adspec) Object
  - used by routers to describe amount of available resources, etc. (later)
- Reservation Confirmation Object
  - identity of the receiver, request for confirmation of successful reservation

Objects (cont'd)

- (Error Specification Object)
- (Integrity Object)
- (Time Values Object)
- (Policy Data Object)
- (Scope Object)
  - to avoid looping in multicasting (later)

Router State

- Most controversial aspect: routers have to maintain state about a flow!
  - inconsistent with Internet "philosophy"
- Values to store about a reservation
  - Sender Template
  - Sender Tspec
  - Hop object, with previous hop (upstream router) in the path
  - Adspec
  - Timeout timer
- Problem: time and space required to compute and store the state

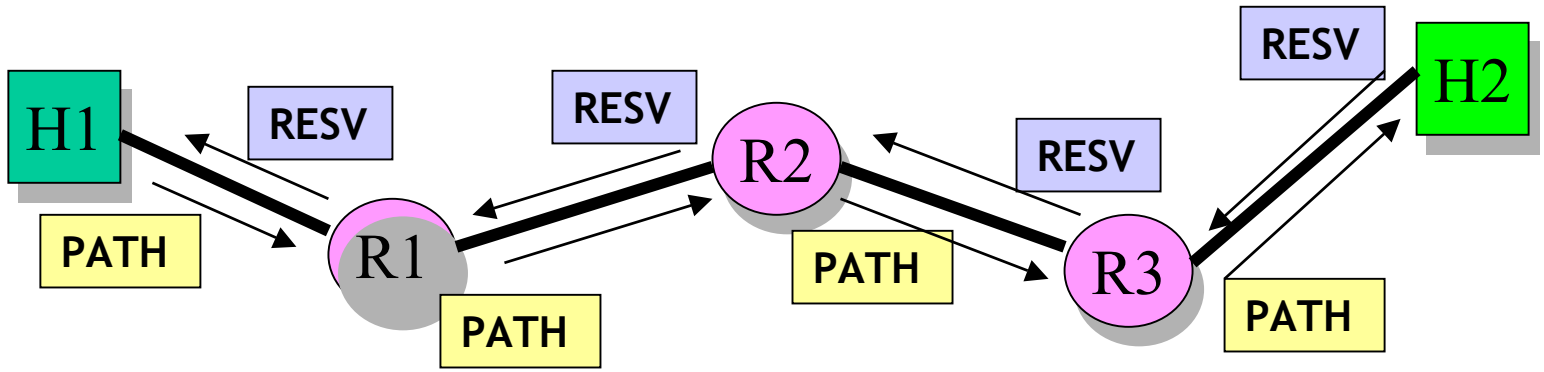
Router State (cont'd)

- Robustness and inconsistency problems
  - what if a sender or a receiver crashes?
  - what if a router crashes?
- "Soft" state -- state is "refreshed" periodically
  - means messages are retransmitted every refresh interval
  - default refresh interval value: every 30 seconds
  - default "timeout" value for state: 3 * refresh interval

Routing and RSVP

- RSVP is routing-protocol independent
  - relies on RSVP messages and data packets to follow the same, reserved path
- No standard for how routing and RSVP interoperate
  - will routing find a suitable path (sufficient resources) given one exists?

Route Pinning and Message Propagation



- Route "pinning"
  - make sure the RESV message retraces the path taken by the PATH message
  - why?
  - during propagation of PATH message, record the "upstream" router identity

Sender Announcements (PATH Message)

- Sending application prepares to send multimedia data to one or more receivers
  - notifies receiver(s) with PATH message
- Important objects in PATH message
  - Session object: destination ID
  - Hop object: for route pinning
  - Sender Template object: sender identity
  - Sender TSpec object: traffic specification
  - Adspec object (optional)

IntServ QoS Parameters for a Path

- Non-IntServ Flag
  - set to 1 or more routers on a path is not enabled for IntServ
- Number of IntServ hops
- Available path bandwidth
  - 32-bit floating-point value (up to 40 TB)
- Minimum path latency (without queueing)
  - 1 microsecond up to 2 minutes
- Maximum packet size (MTU) allowed along the path

Adspec

- Adspec is added by routers to the PATH message as it propagates to the receivers
- Includes the above IntServ parameters
- For Guaranteed Service, Adspec also includes...
  1. total "rate-dependent" delay *Ctot*
     - e.g., sum of the transmission delays of all links in the path
  2. total "rate-independent" delay *Dtot*
     - e.g., sum of the propagation delays of all routers in the path
  - calculated by routers and inserted in Adspec

Guaranteed Service Delay Bounds

- Quantitative bounds on delay
- Accomplished through shaping and packet scheduling (e.g., WFQ)
- Requires receiver to provide the Flowspec (as part of RESV message)
  - includes Rspec = rate required (R) and slack term (S)

Guaranteed Service Delay Bounds
Computed by Receiver

- Sender's Tspec includes…
  - bucket size *b*
  - token generation rate *r*
  - peak rate *p*
- Router Adspec includes
  - max packet size *M*
  - Available path bandwidth
  - Minimum path latency
  - *Ctot*
  - *Dtot*
- Receiver must calculate *R*, rate to be assigned to flow

GS Delay Bounds (cont'd)

- Application determines maximum total end-to-end delay bound
  - Ex.: for voice, 150ms
- Receiver calculates allowable queueing delay in network
  - End-to-end queueing delay "budget" = end-to-end delay bound – minimum path latency
  - Ex.: minimum path latency = 30ms, max queueing delay = 150-30 = 120ms
- End-to-end queueing delay for Guaranteed Service
  - = $(b-M)*(p-R) \ / \ (R*(p-r)) + (M+Ctot)/R + Dtot$ (assuming $R < p$)
  - = $(M+Ctot)/R + Dtot$ (*assuming* $R \geq p$)

Receiver Reservation Requests (RESV Message)

- Given max queueing delay, solve for *R*
- If calculated *R* exceeds maximum path bandwidth (in Adspec)
  - accept max path bandwidth instead, or
  - give up (no reservation)!
- If acceptable, receiver generates a RESV message and sends to sender

RESV Message (cont'd)

- Important objects in RESV message
  - Session object: destination (receiver) ID
  - Hop object
  - Filterspec object: source (sender) ID
  - Merging Style object
  - Flowspec object
- Flowspec contains
  - sender's Tspec, with max packet size replace by path MTU
  - Rspec = *R*, rate required to be reserved

RESV Message Processing

- RESV message propagates back to the sender
  - routers may accept or reject the reservation request
  - if accept, router installs state and propagates message
  - if reject, router sends RESVERR message to receiver

Other RSVP Messages

- State can be explicitly removed using PATHTEAR and RESVTEAR teardown messages
- PATHERR message for communicating errors / failure during notification
- RESVERR message for communicating errors / failure during reservations
- CONFIRM message for telling receivers of successful reservation establishment

Reservation Merging "Styles"

- For multiple receivers, and multiple senders case
- All receivers must agree on what style of reservation merging will be done
- Fixed Filter Style
  - each sender has its own distinct reservation
  - reservations are managed separately

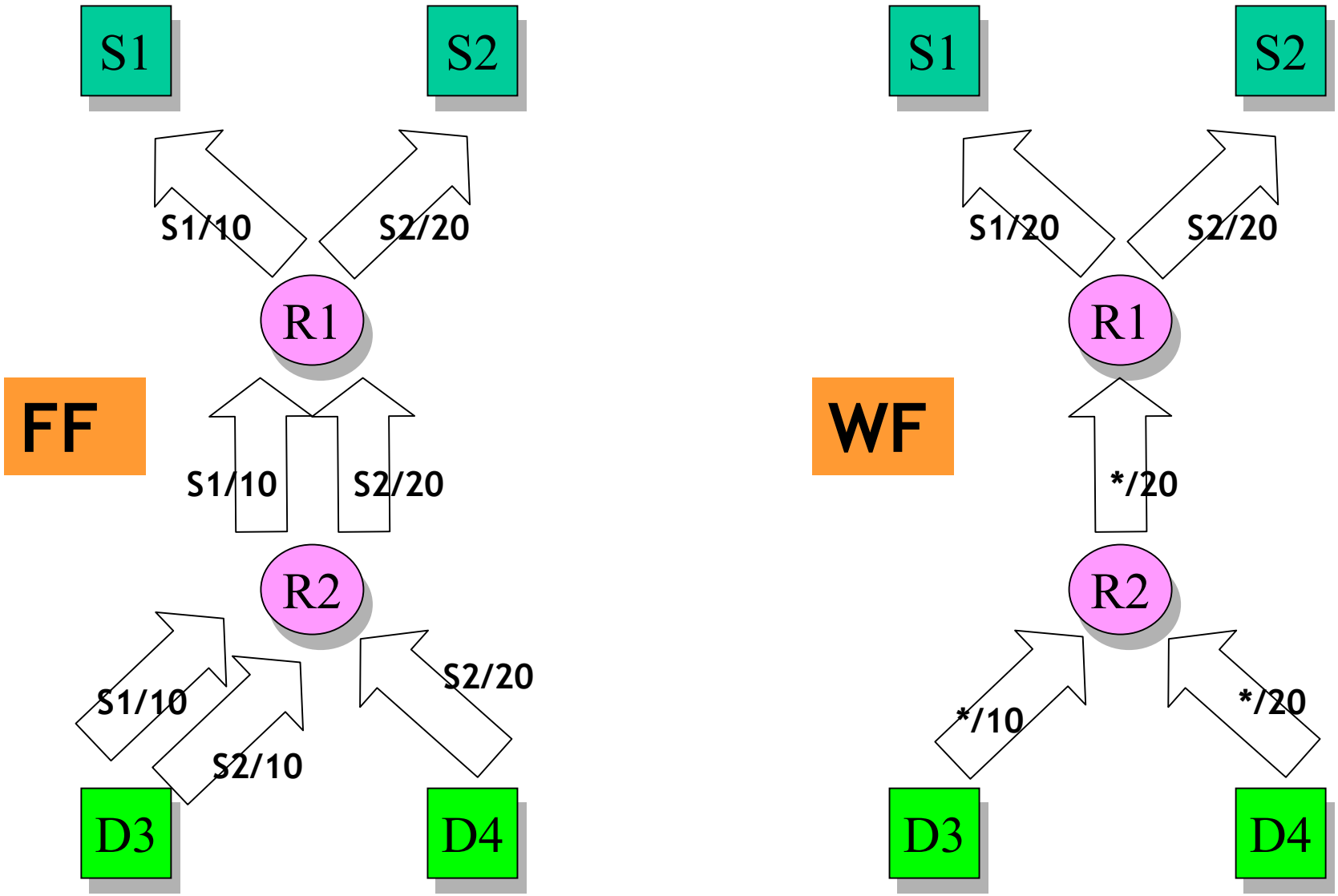
Merging Styles (cont'd)

- Wildcard Filter Style
  - only one reservation, shared by all senders
  - ex.: audio conferencing
- Shared Explicit Filter Style
  - specific senders share a single reservation
  - receiver identifies which senders

Reservation Merging

- Fixed filter
  - maximum resource amount requested by any receiver is reserved
  - to each sender
- Wilcard Filter
  - maximum resource amount is reserved
  - to all senders
- Questions
  - what do you do with RESVERR messages (admission denied) for merged flows?
  - what do you do about RESVTEAR for a single receiver?
    - soft state takes care of this?

Merging (cont'd)



Merging of Tspecs

- Token bucket rate = sum of (token bucket rates)
- Token bucket size = sum of (token bucket sizes)
- Peak rate = sum of (peak rates)
- Minimum policed unit = min of (min policed unit)
- Maximum packet size = max of (max packet size)

Sources of Information

- Durham and Yavatkar, "Inside the Internet Resource Reservation protocol", 1999
- Internet:
  - RSVP tutorial: http://ntrg.cs.tcd.ie/htewari/papers/white97rsvp.pdf
  - RFC2212: Specification of Guaranteed Quality of Service
  - RFC2205: Resource Reservation Protocol – Version 1 Functional Spec