CS 268: Lecture 10 (Integrated Services)

Ion Stoica

March 4, 2002

Limitations of IP Architecture in Supporting Resource Management

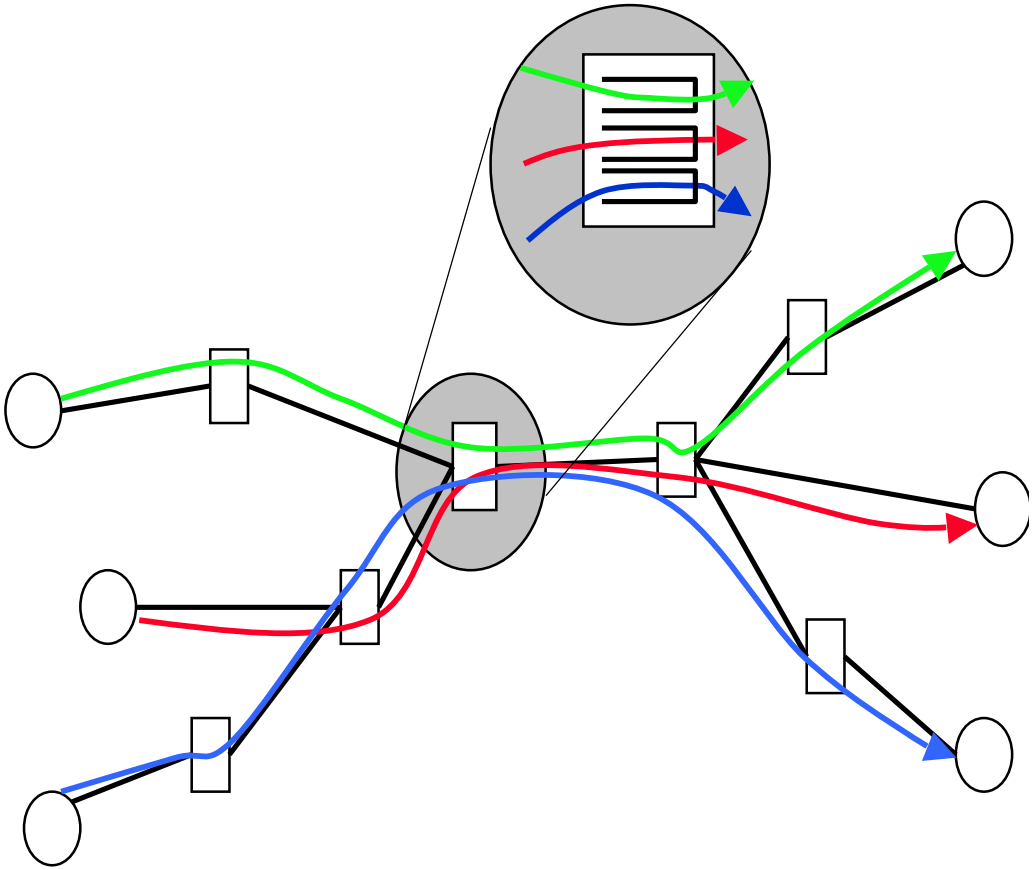
- IP provides only best effort service
- IP does not participate in resource management
  - Cannot provide service guarantees on a per flow basis
  - Cannot provide service differentiation among traffic aggregates
- Early efforts
  - Tenet group at Berkeley (Ferrari and Verma)
  - ATM
- IETF efforts
  - Integrated services initiative
  - Differentiated services initiative

Integrated Services Internet

- Enhance IP's service model
  - Old model: single best-effort service class
  - New model: multiple service classes, including best-effort and QoS classes
- Create protocols and algorithms to support new service models
  - Old model: no resource management at IP level
  - New model: explicit resource management at IP level
- Key architecture difference
  - Old model: stateless
  - New model: per flow state maintained at routers
    - used for admission control and scheduling
    - set up by signaling protocol

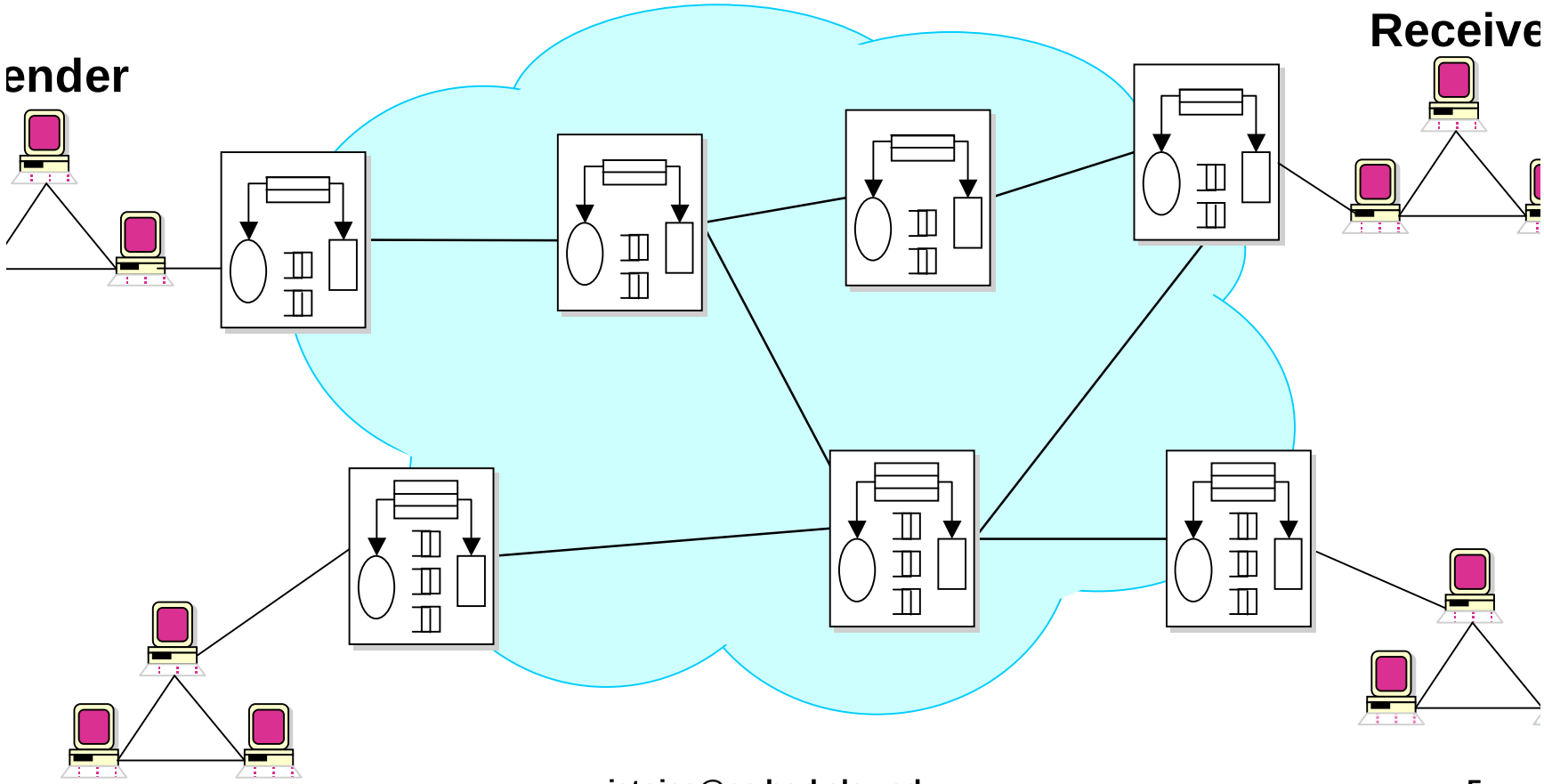
Integrated Services Network

- Flow or session as QoS abstractions
- Each flow has a fixed or stable path
- Routers along the path maintain the state of the flow



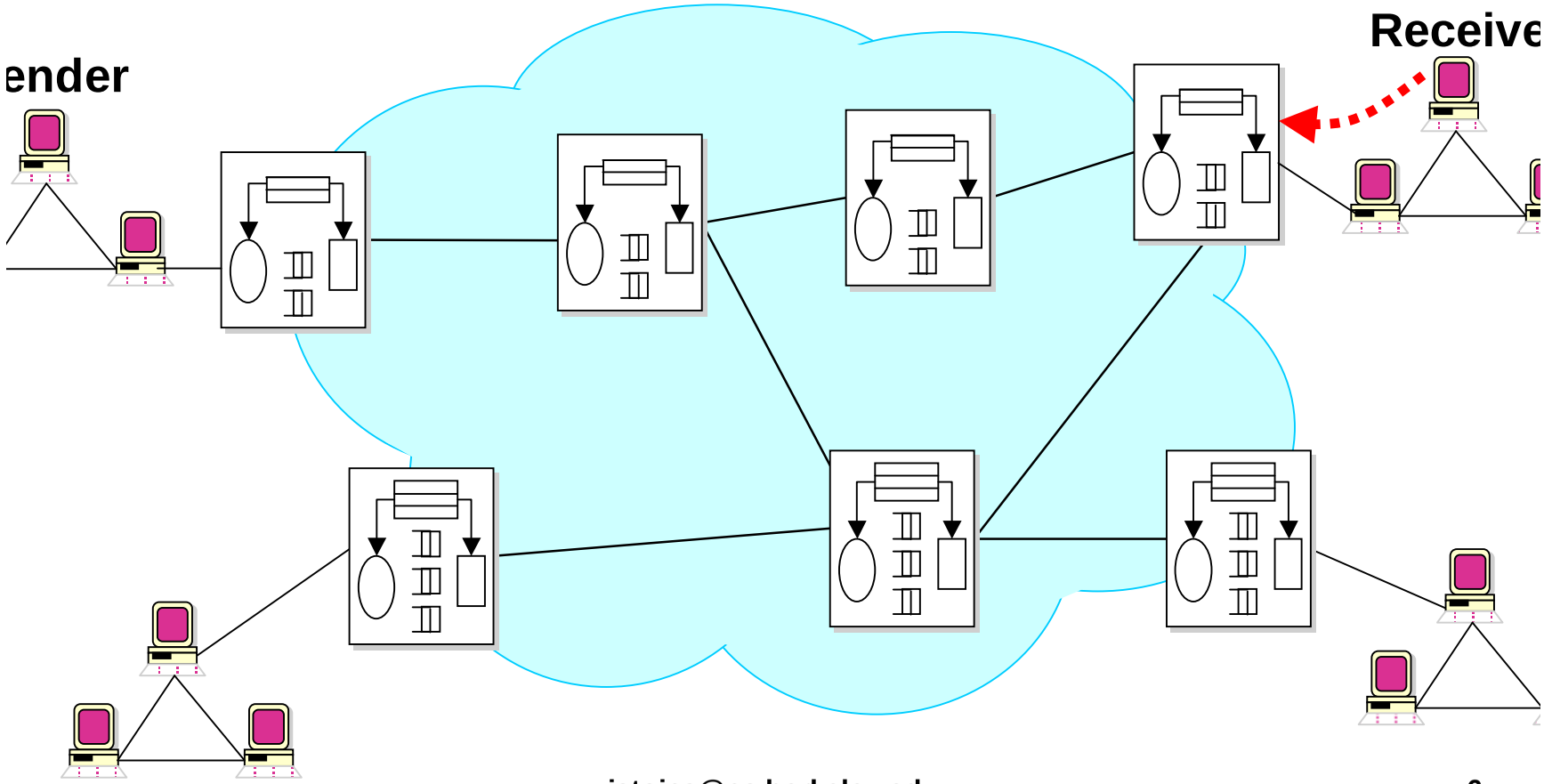
Integrated Services Example

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



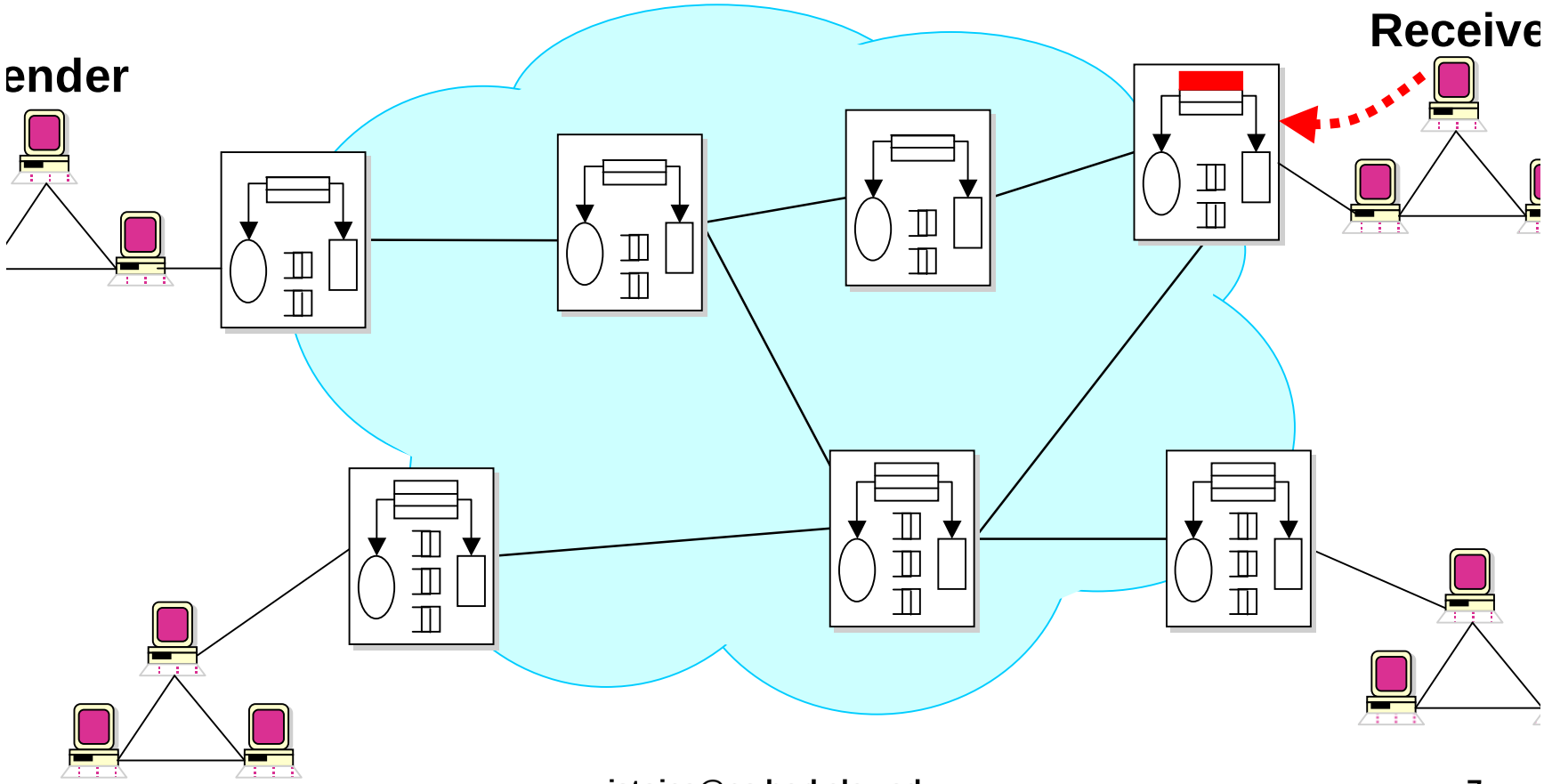
Integrated Services Example

- Allocate resources - perform per-flow admission control



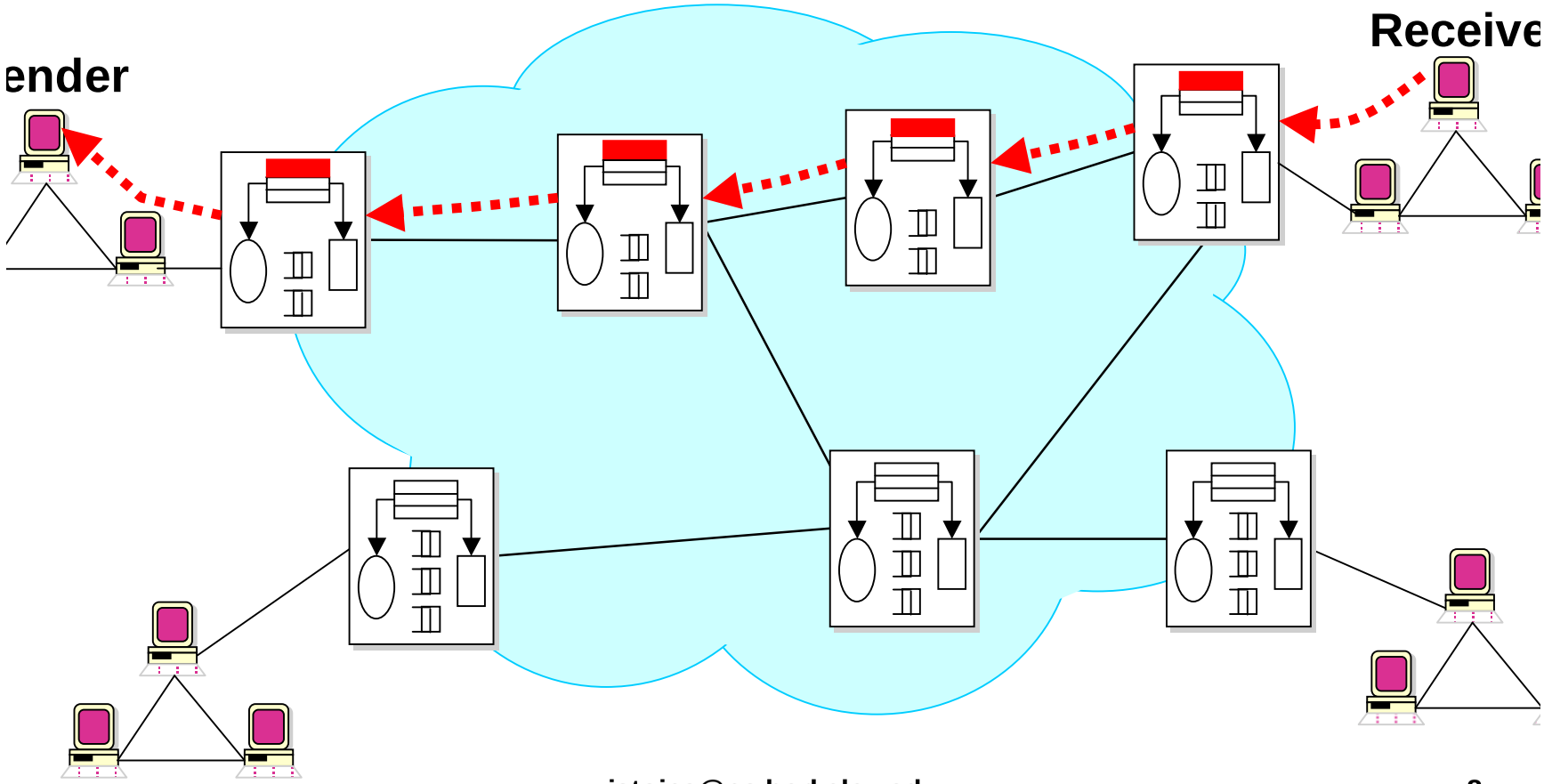
Integrated Services Example

- Install per-flow state



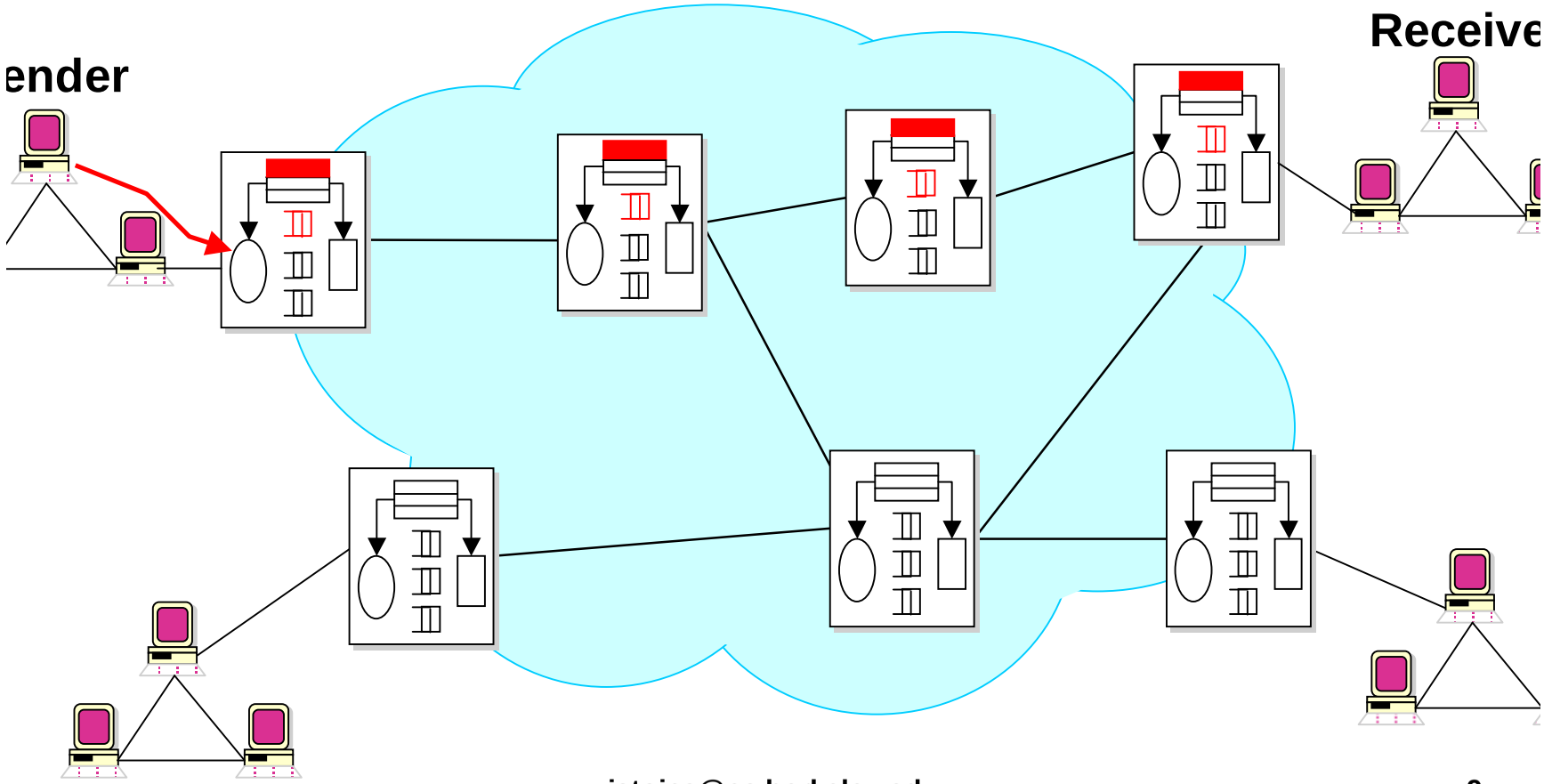
Integrated Services Example

- Install per flow state



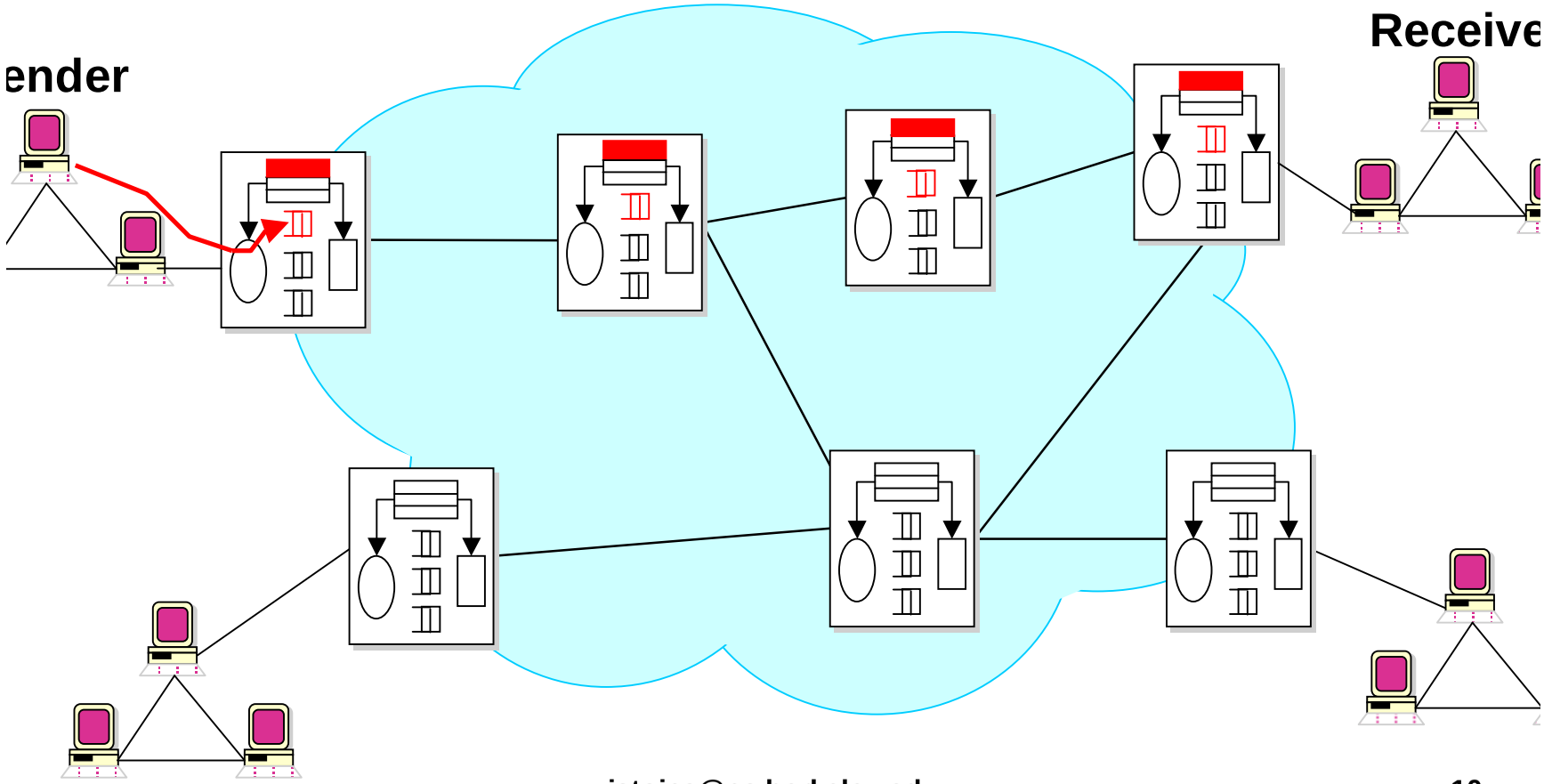
Integrated Services Example: Data Path

- Per-flow classification



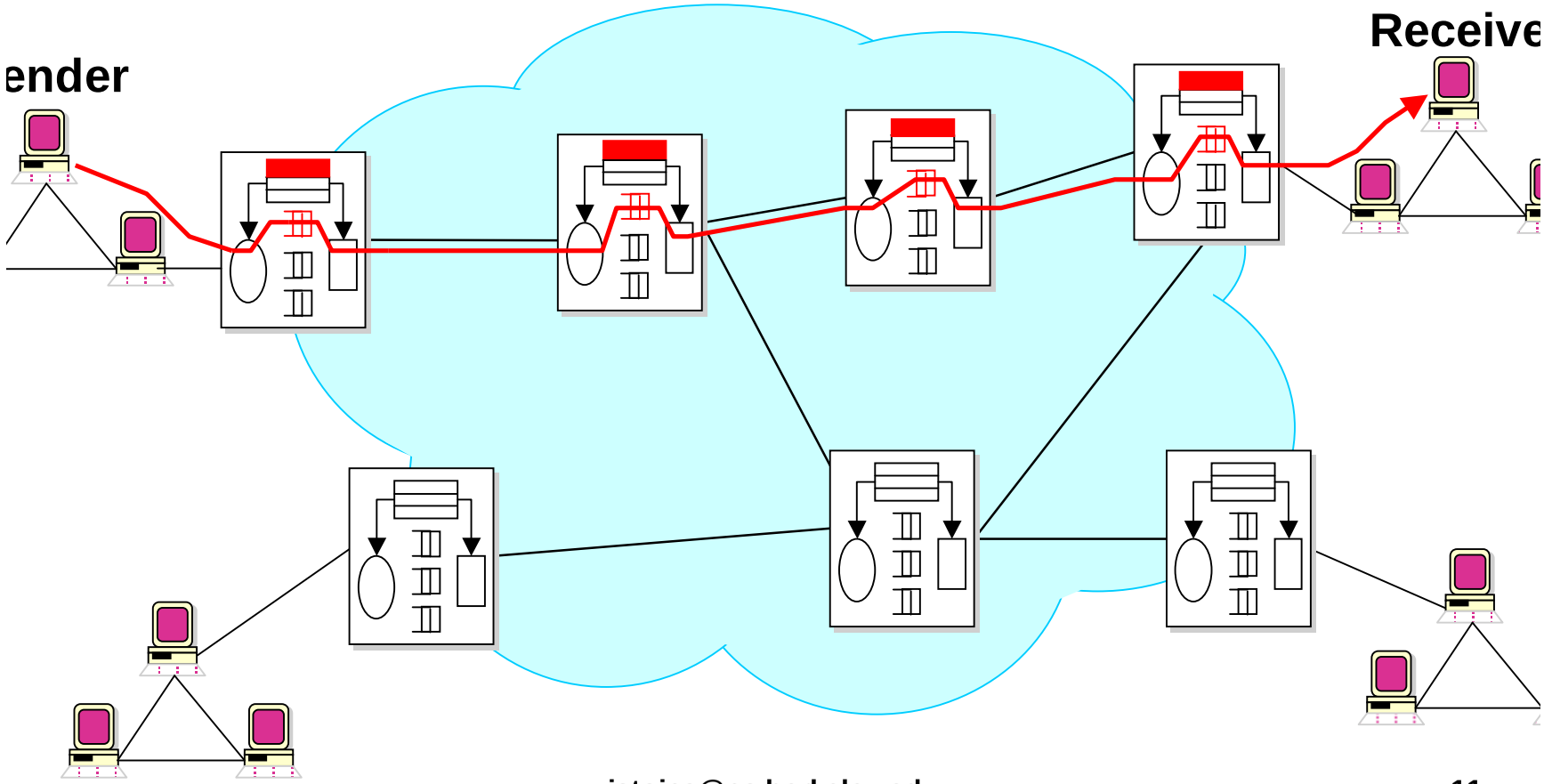
Integrated Services Example: Data Path

- Per-flow buffer management

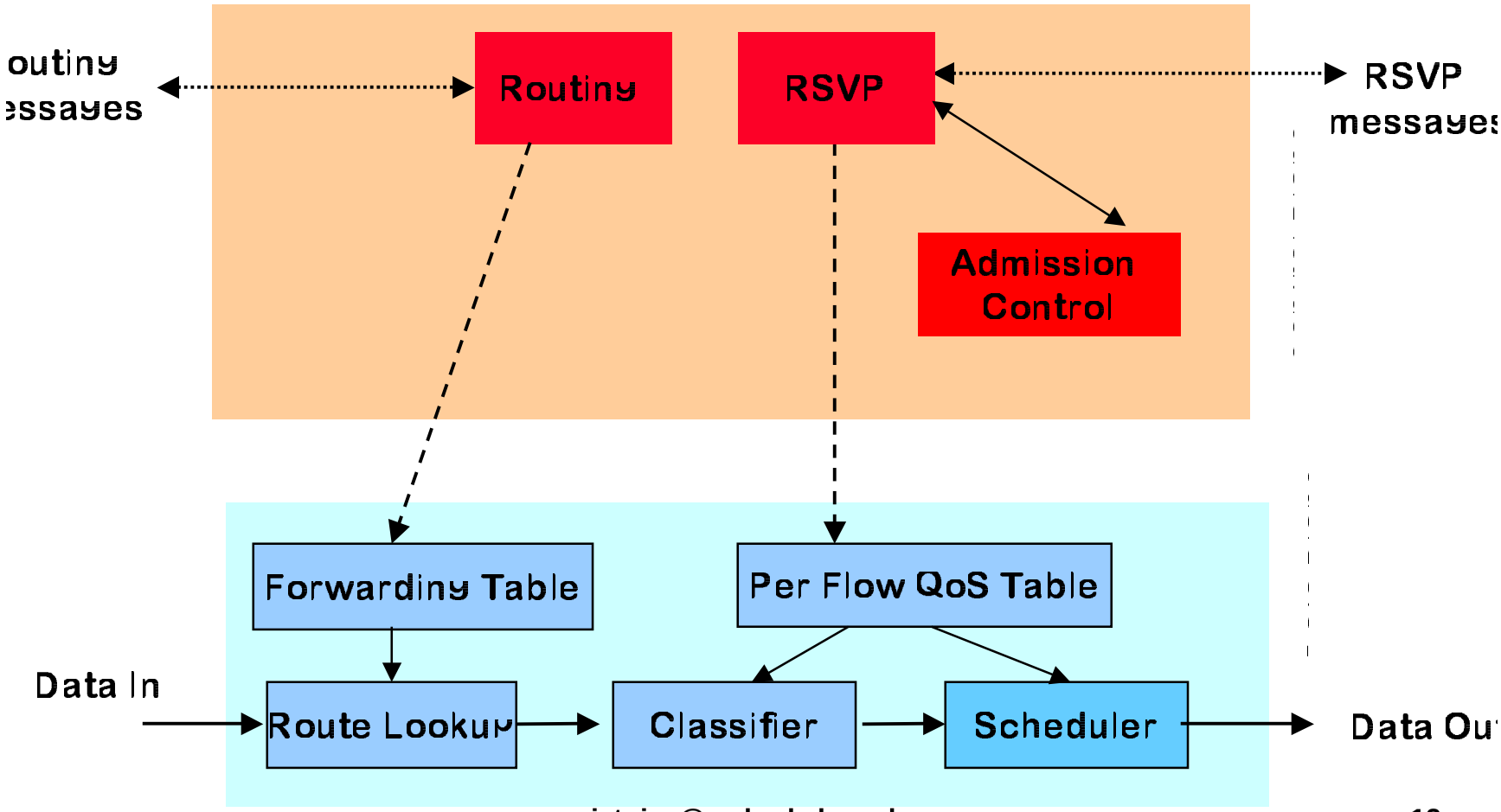


Integrated Services Example

- Per-flow scheduling



How Things Fit Together



Service Classes

- Multiple service classes
- Service can be viewed as a contract between network and communication client
  - end-to-end service
  - other service scopes possible
- Three common services
  - best-effort (“elastic” applications)
  - hard real-time (“real-time” applications)
  - soft real-time (“tolerant” applications)

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

Role of RSVP in the Architecture

- Signaling protocol for establishing per flow state
- Carry resource requests from hosts to routers
- Collect needed information from routers to hosts
- At each hop
  - consults admission control and policy module
  - sets up admission state or informs the requester of the failure

RSVP Design Features

- IP Multicast centric design
- Receiver initiated reservation
- Different reservation styles
- Soft state inside network
- Decouple routing from reservation

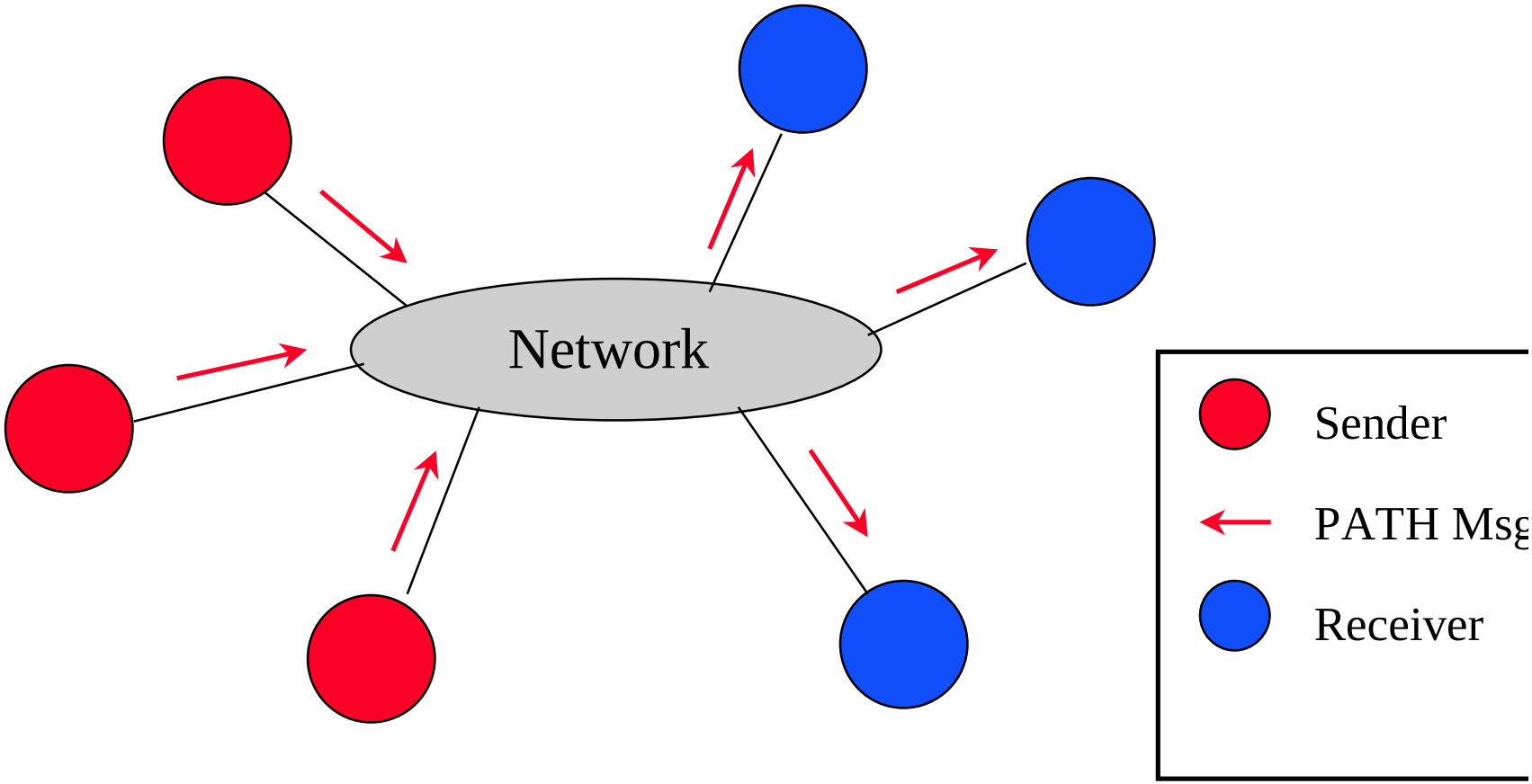
IP Multicast

- Best-effort MxN delivery of IP datagrams
- Basic abstraction: IP multicast group
  - identified by Class D address: 224.0.0.0 - 239.255.255.255
  - sender needs only to know the group address, but not the membership
  - receiver joins/leaves group dynamically
- Routing and group membership managed distributedly
  - no single node knows the membership
  - tough problem
  - various solutions: DVMRP, CBT, PIM

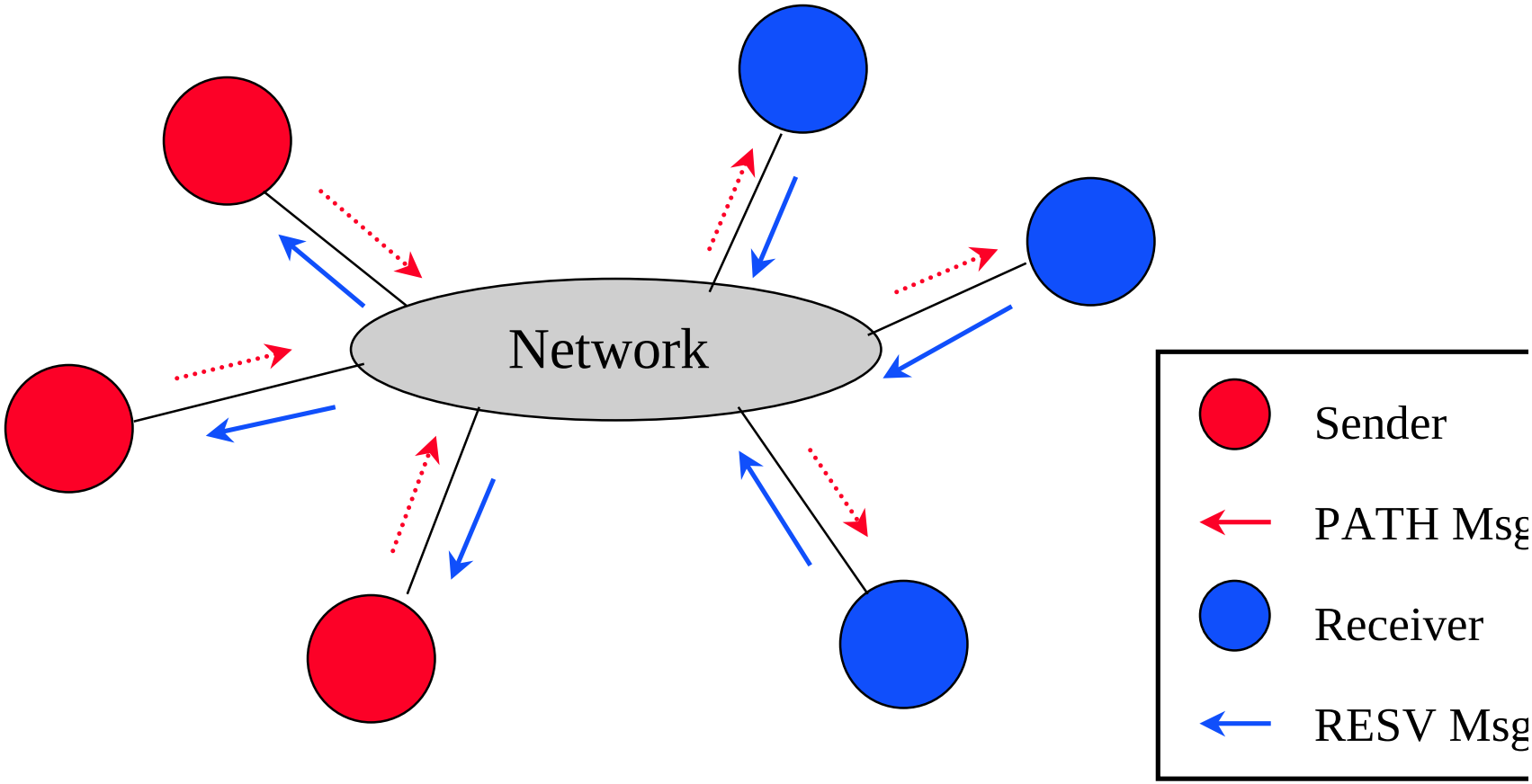
RSVP Reservation Model

- Performs signaling to set up reservation state for a session
- A session is a simplex data flow sent to a unicast or a multicast address, characterized by
  - <IP dest, protocol number, port number>
- Multiple senders and receivers can be in session

The Big Picture



The Big Picture (2)

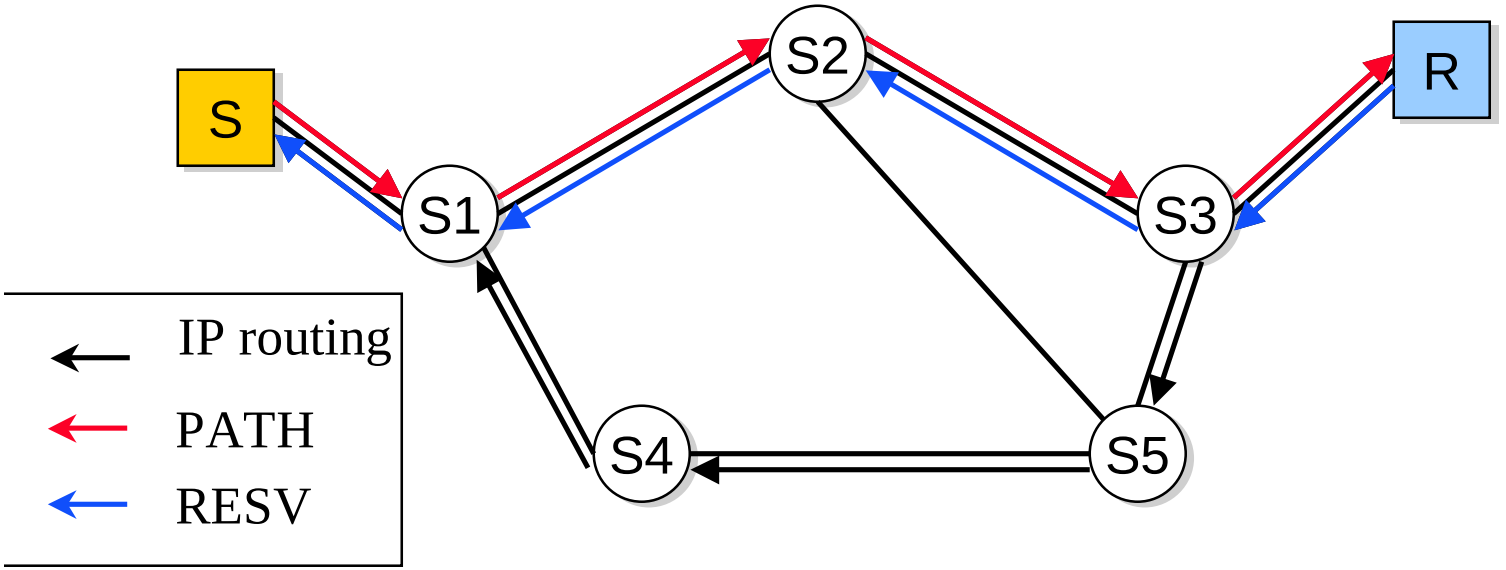


RSVP Basic Operations

- Sender sends PATH message via the data delivery path
  - set up the path state each router including the address of previous hop
- Receiver sends RESV message on the reverse path
  - specifies the reservation style, QoS desired
  - set up the reservation state at each router
- Things to notice
  - receiver initiated reservation
  - decouple the routing from reservation
  - two types of state: path and reservation

Route Pinning

- Problem: asymmetric routes
  - You may reserve resources on R→S3→S5→S4→S1→S, but data travels on S→S1→S2→S3→R !
- Solution: use PATH to remember direct path from S to R, I.e., perform route pinning

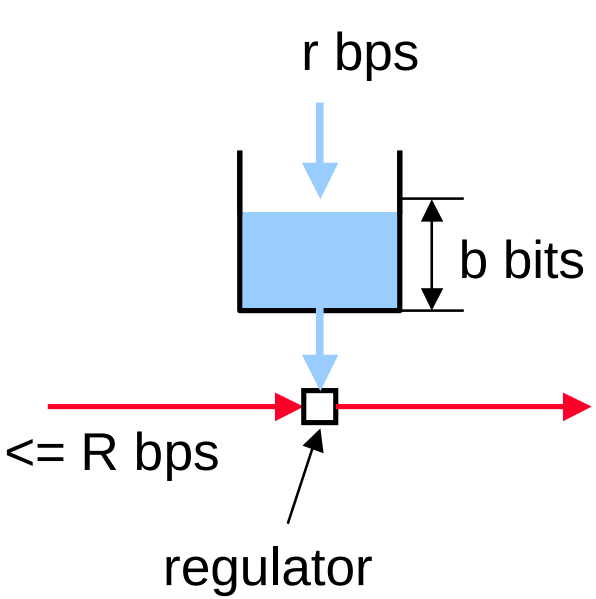


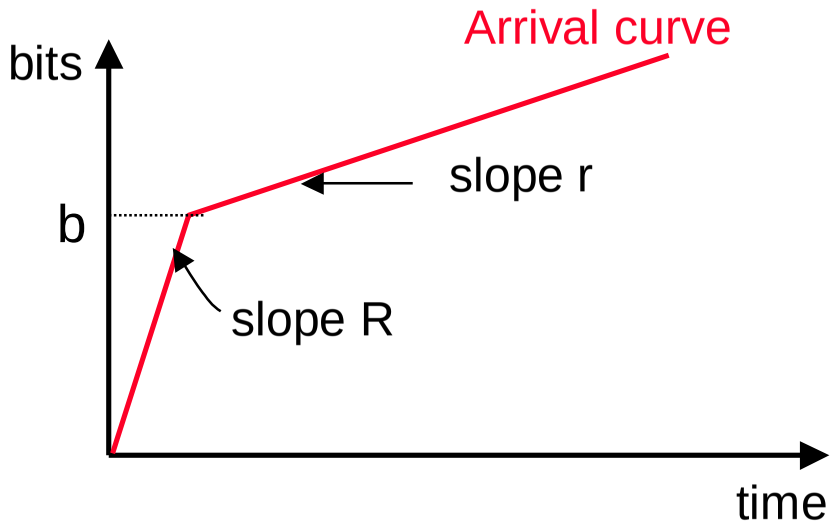
PATH and RESV messages

- PATH also specifies
  - Source traffic characteristics
    - use token bucket
  - Reservation style – specify whether a RESV message will be forwarded to this server
- RESV specifies
  - Queueing delay and bandwidth requirements
  - Source traffic characteristics (from PATH)
  - Filter specification, i.e., what senders can use reservation
  - Based on these routers perform reservation

Token Bucket

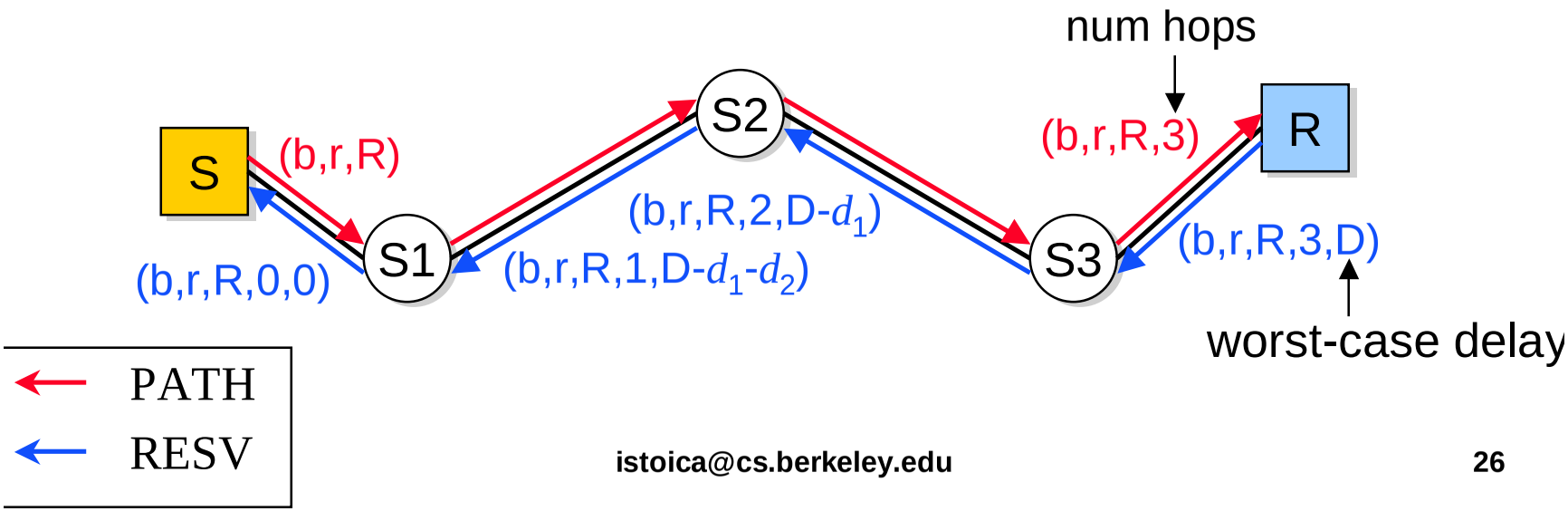
- Characterized by two parameters (r, b)
  - r – average rate
  - b – token depth
- Assume flow arrival rate <= R bps (e.g., R link capacity)
- A bit is transmitted only when there is an available token
- Arrival curve – maximum amount of bits transmitted by time t





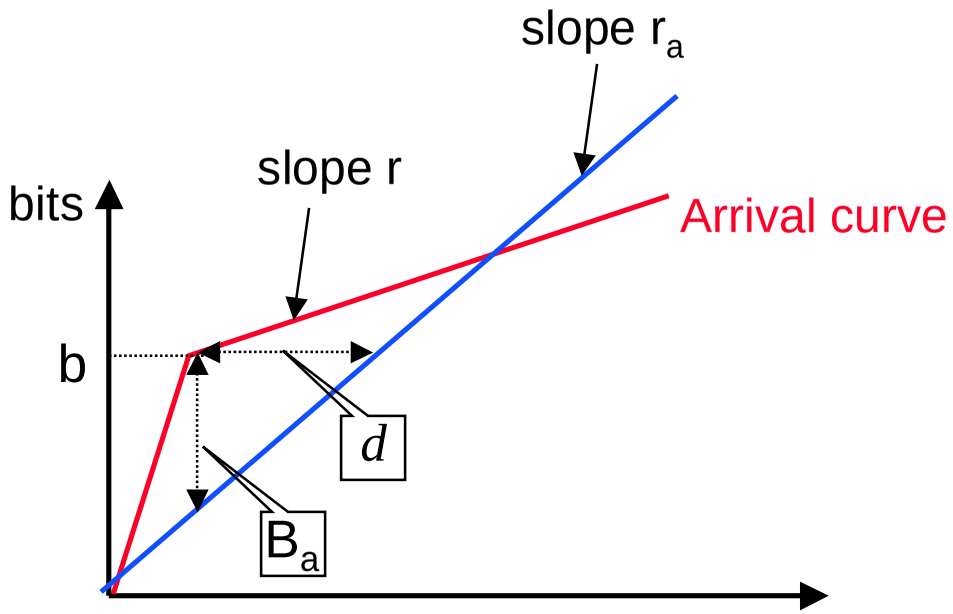
End-to-End Reservation

- When R gets PATH message it knows
  - Traffic characteristics (tspec): (r,b,R)
  - Number of hops
- R sends back this information + worst-case delay in RESV
- Each router along path provide a per-hop delay guarantee and forward RESV with updated info
  - In simplest case routers split the delay



Per-hop Reservation

- Given (b,r,R) and per-hop delay d
- Allocate bandwidth r_a and buffer space B_a such that to guarantee d



Reservation Style

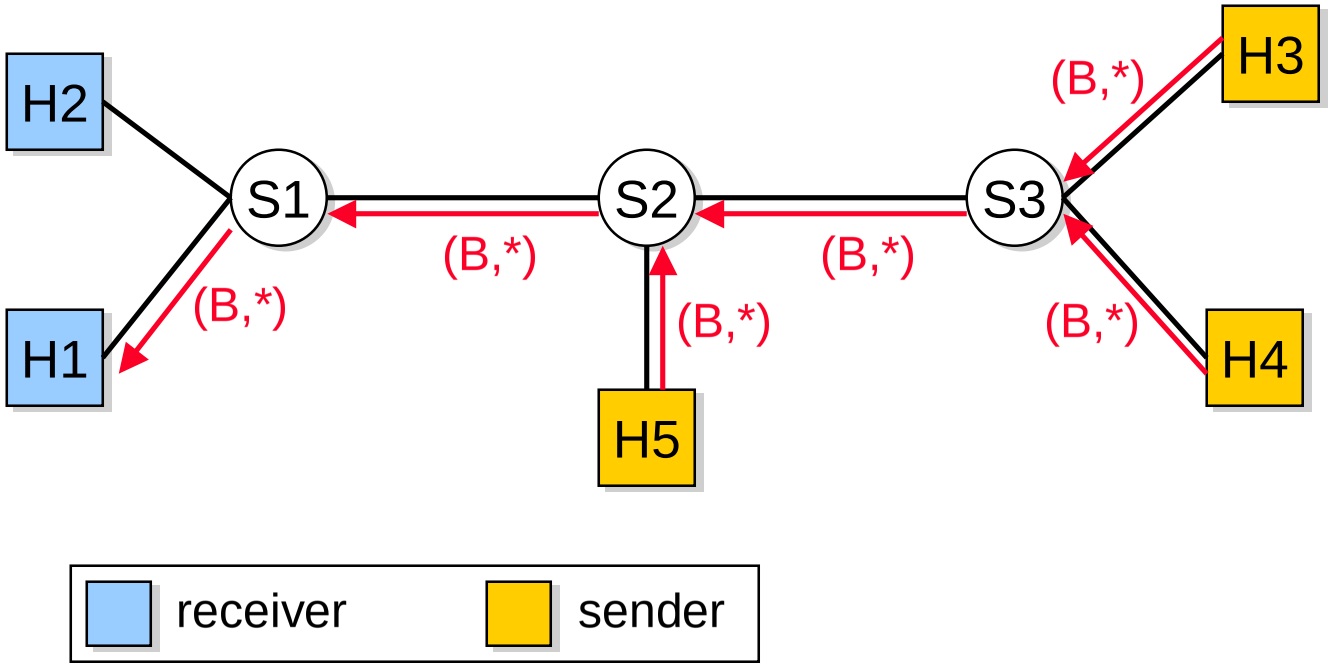
- Motivation: achieve more efficient resource utilization in multicast (M x N)
- Observation: in a video conferencing when there are M senders, only a few can be active simultaneously
  - multiple senders can share the same reservation
- Various reservation styles specify different rules for sharing among senders

Reservation Styles and Filter Spec

- Reservation style
  - use filter to specify which sender can use the reservation
- Three styles
  - wildcard filter: does not specify any sender; all packets associated to a destination shares same resources
    - Group in which there are a small number of simultaneously active senders
  - fixed filter: no sharing among senders, sender explicitly identified for the reservation
    - Sources cannot be modified over time
  - dynamic filter: resource shared by senders that are (explicitly) specified
    - Sources can be modified over time

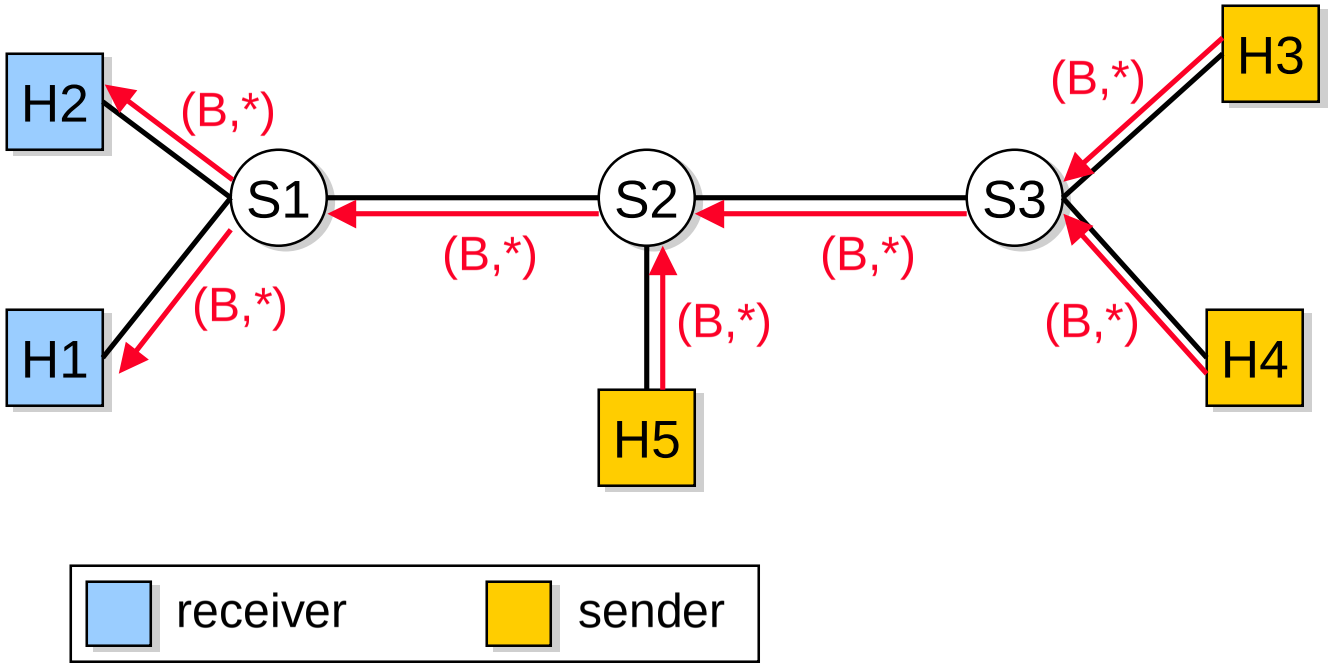
Wildcard Filter Example

- Receivers: H1, H2; senders: H3, H4, H5
- Each sender sends B
- H1 reserves B; listen from one server at a time



Wildcard Filter Example

- H2 reserves B

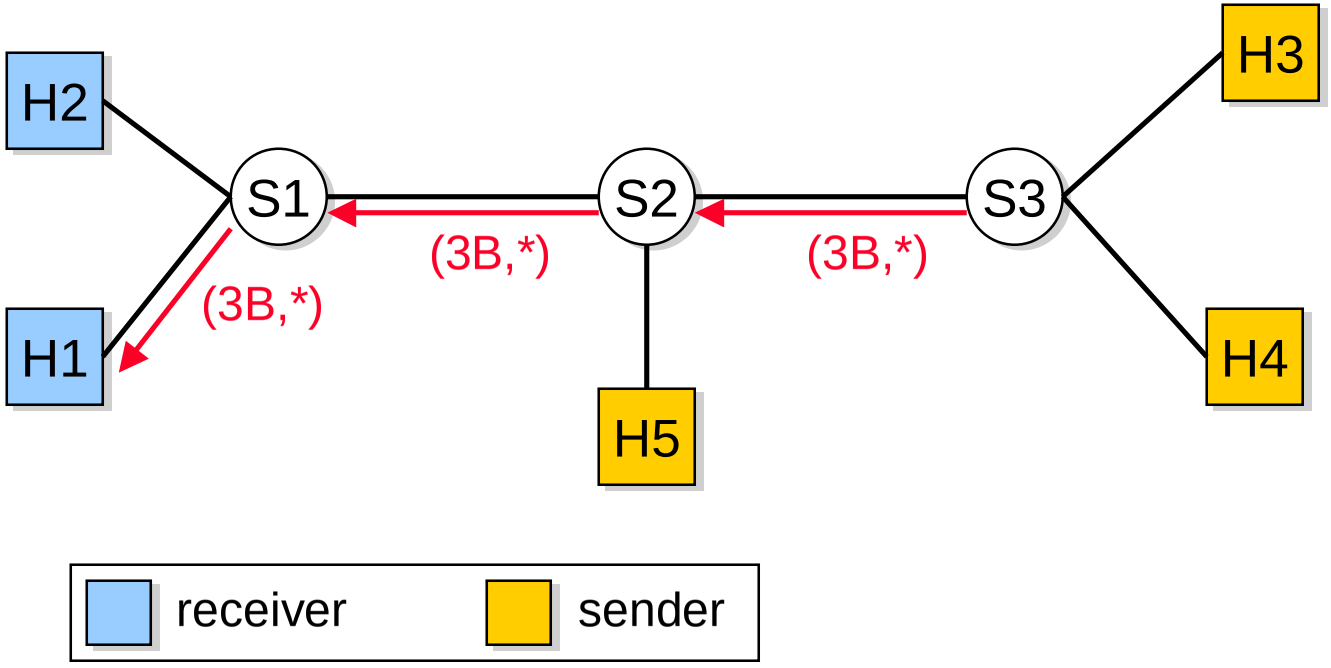


Wildcard Filter

- Advantages
  - Minimal state at routers
    - Routers need to maintain only routing state augmented by reserved bandwidth on outgoing links
- Disadvantages
  - May result in inefficient resource utilization

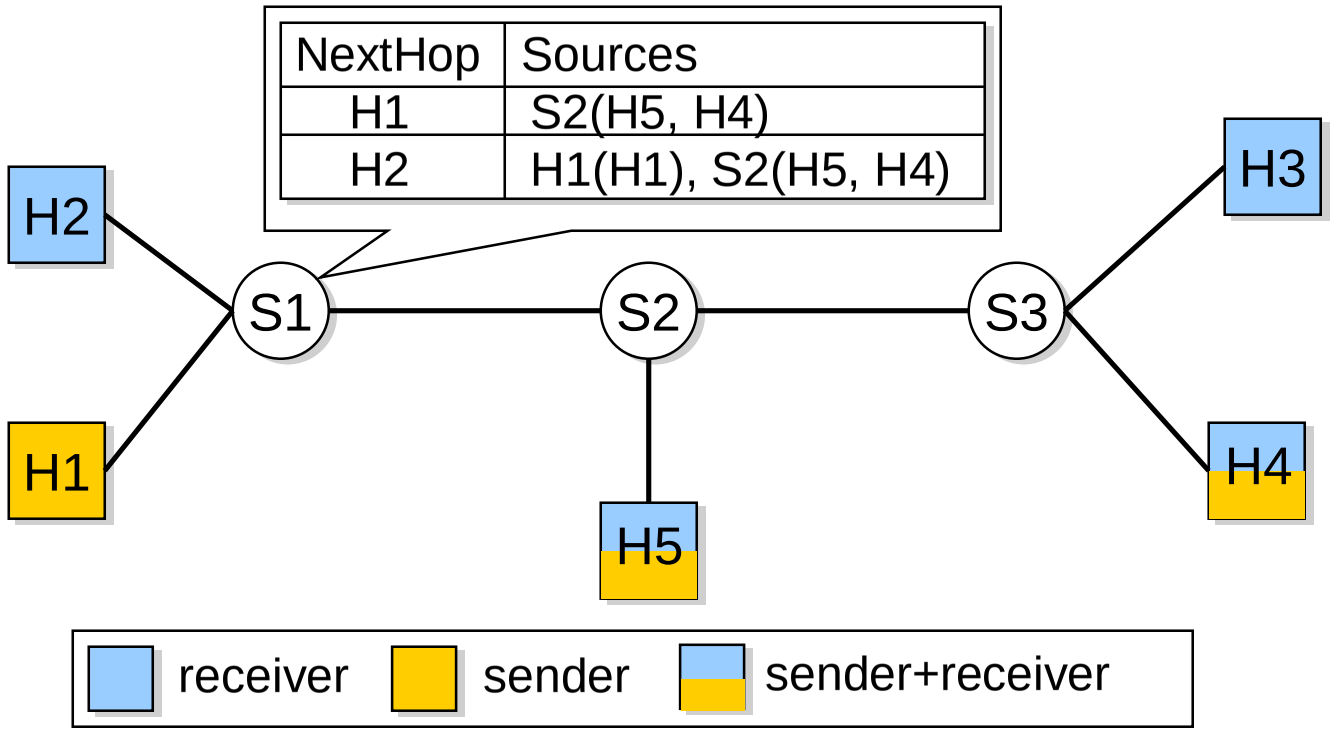
Wildcard Filter: Inefficient Resource Utilization Example

- H1 reserves 3B; wants to listen from all senders simultaneously
- Problem: reserve 3B on (S3:S2) although 2B sufficient !



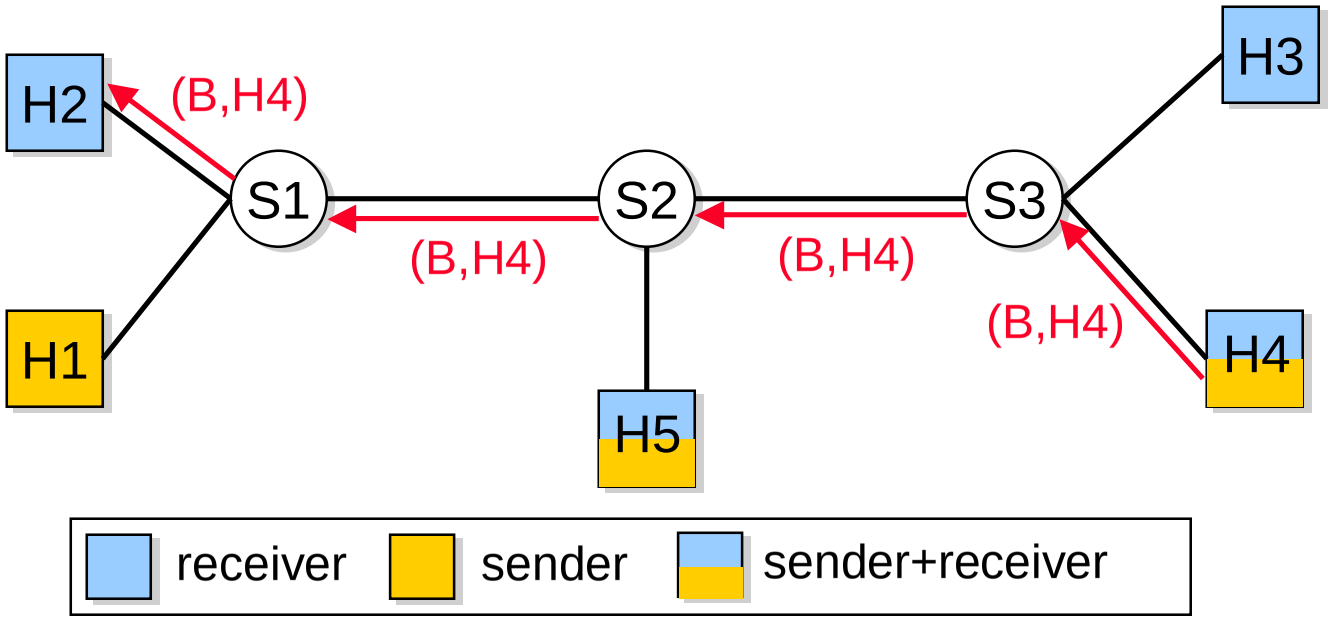
Fixed Filter Example

- Receivers: H2, H3, H4, H4; Sender: H1, H4, H5
- Routers maintain state for each receiver in the routing table



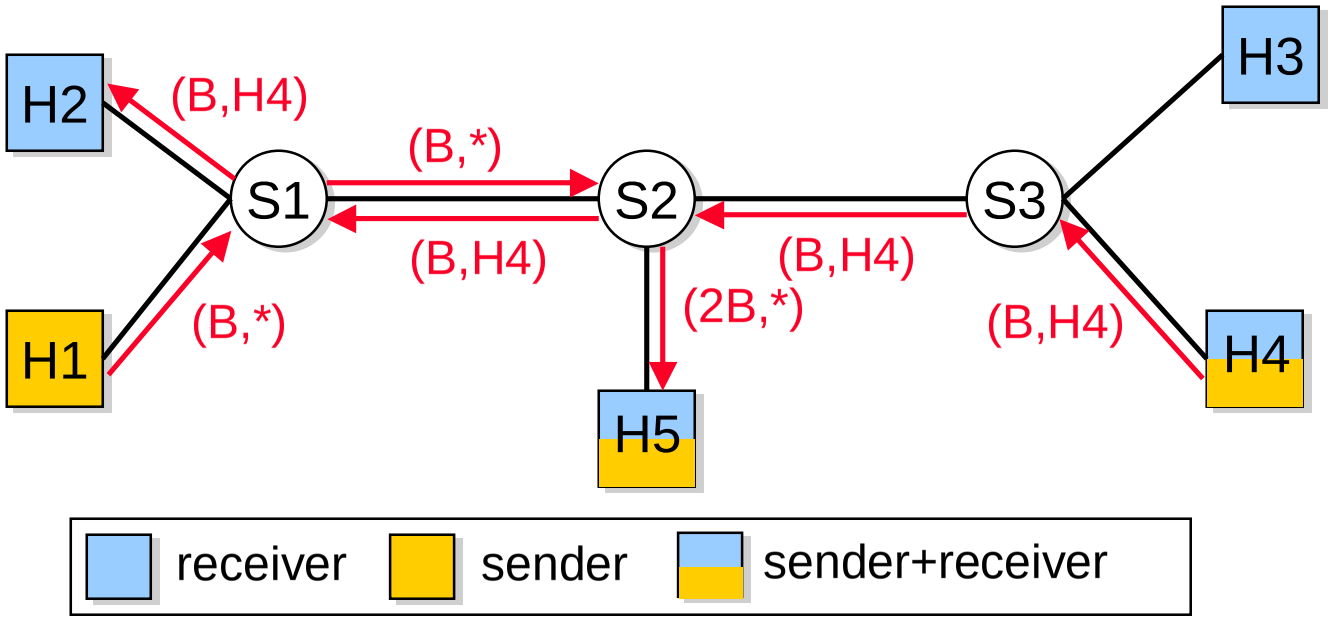
Fixed Filter Example

- H2 wants to receive B only from H4



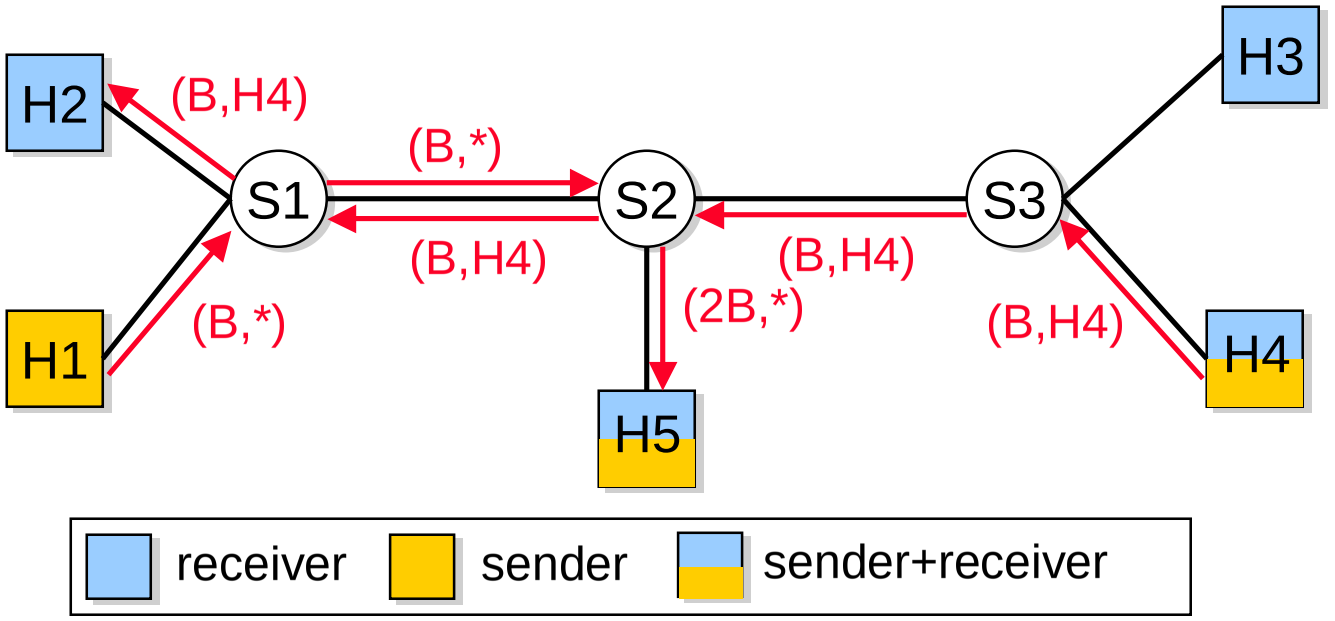
Dynamic Filter Example

- H5 wants to receive 2B from any source



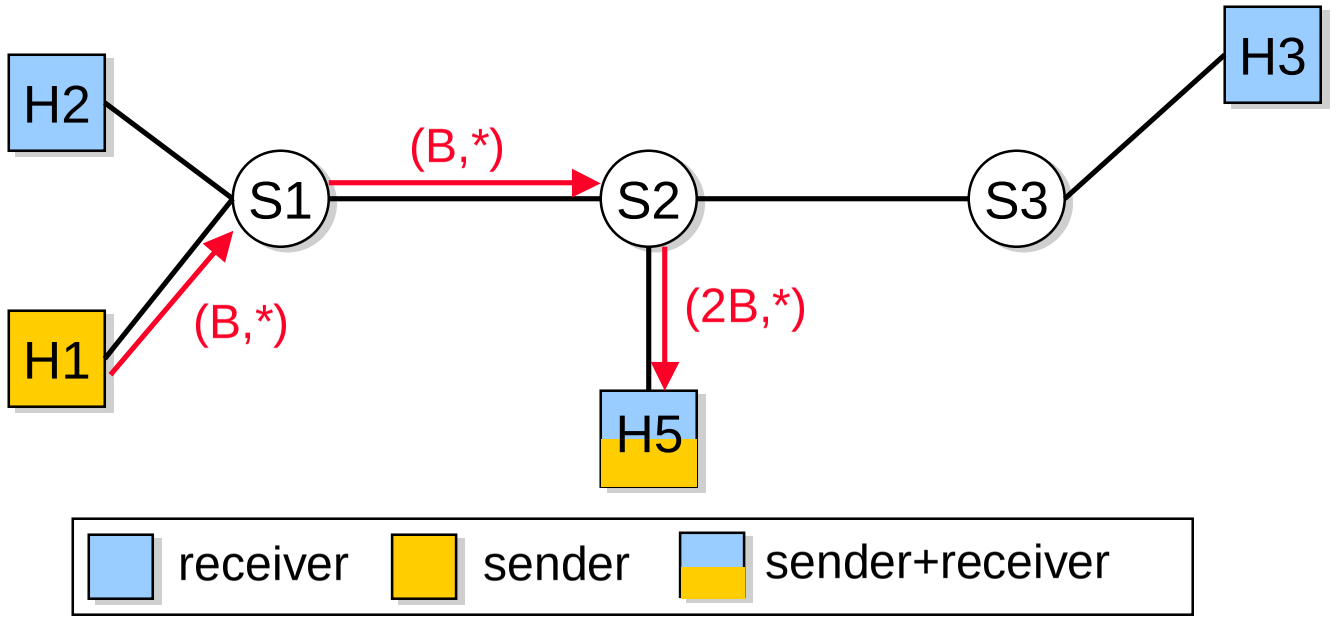
Tire-down Example

- H4 leaves the group
  - H4 no longer sends PATH message
  - State corresponding to H4 removed



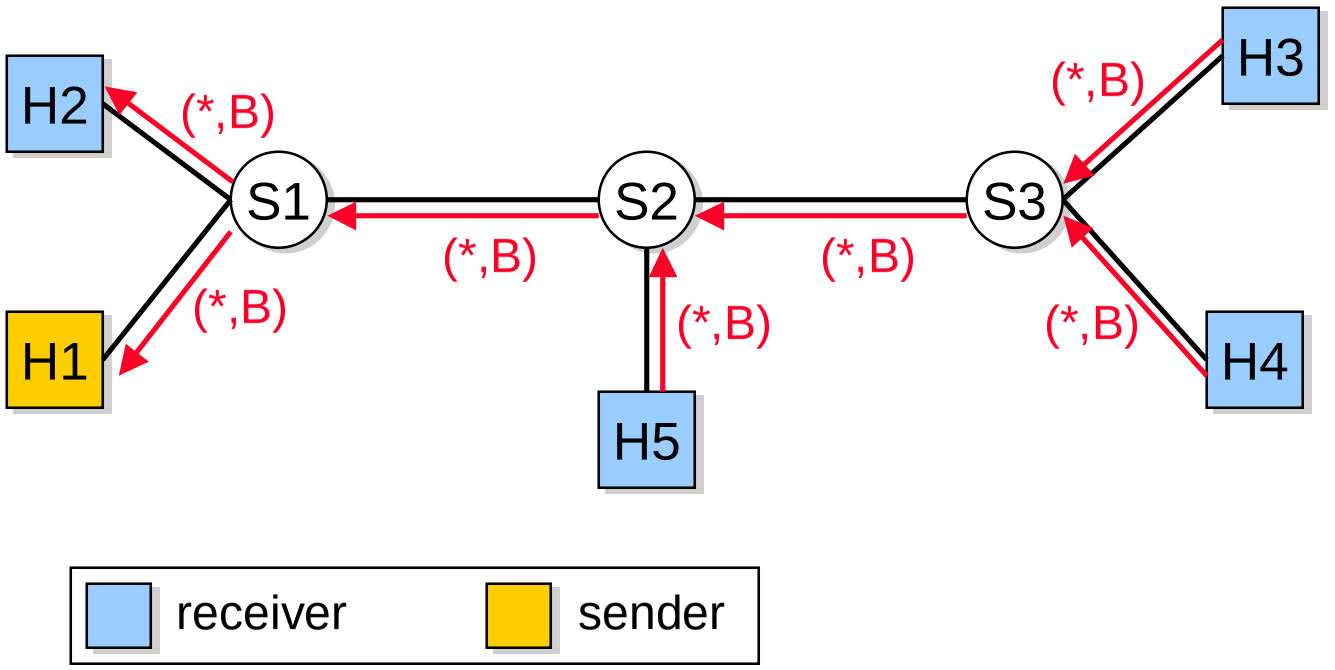
Tire-down Example

- H4 leaves the group
  - H4 no longer sends PATH message
  - State corresponding to H4 removed



Fixed Filter Example

- Receivers: H2, H3, H4, H4; Sender: H1



Soft State

- Per session state has a timer associated with it
  - path state, reservation state
- State lost when timer expires
- Sender/Receiver periodically refreshes the state, resends PATH/RESV messages, resets timer
- Claimed advantages
  - no need to clean up dangling state after failure
  - can tolerate lost signaling packets
    - signaling message need not be reliably transmitted
  - easy to adapt to route changes
- State can be explicitly deleted by a Teardown message

RSVP and Routing

- RSVP designed to work with variety of routing protocols
- Minimal routing service
  - RSVP asks routing how to route a PATH message
- Route pinning
  - addresses QoS changes due to “avoidable” route changes while session in progress
- QoS routing
  - RSVP route selection based on QoS parameters
  - granularity of reservation and routing may differ
- Explicit routing
  - Use RSVP to set up routes for reserved traffic

Recap of RSVP

- PATH message
  - sender template and traffic spec
  - advertisement
  - mark route for RESV message
  - follow data path
- RESV message
  - reservation request, including flow and filter spec
  - reservation style and merging rules
  - follow reverse data path
- Other messages
  - PathTear, ResvTear, PathErr, ResvErr

Question

- What do you think about the design decision to make RSVP IP multicast centric?

What is still Missing?

- Classification algorithm
- Scheduling algorithm
- Admission control algorithm
- QoS Routing algorithm