CEG 7450:
(Integrated Services)

Reading

[PG93] Abhay K. Parekh and Robert Gallager, ``A generalized processor sharing approach to flow control in integrated services networks: the single node case,'' IEEE/ACM Trans. on Networking, Vol. 1, No. 3, pages 344--357, June 1993

What is the Problem?

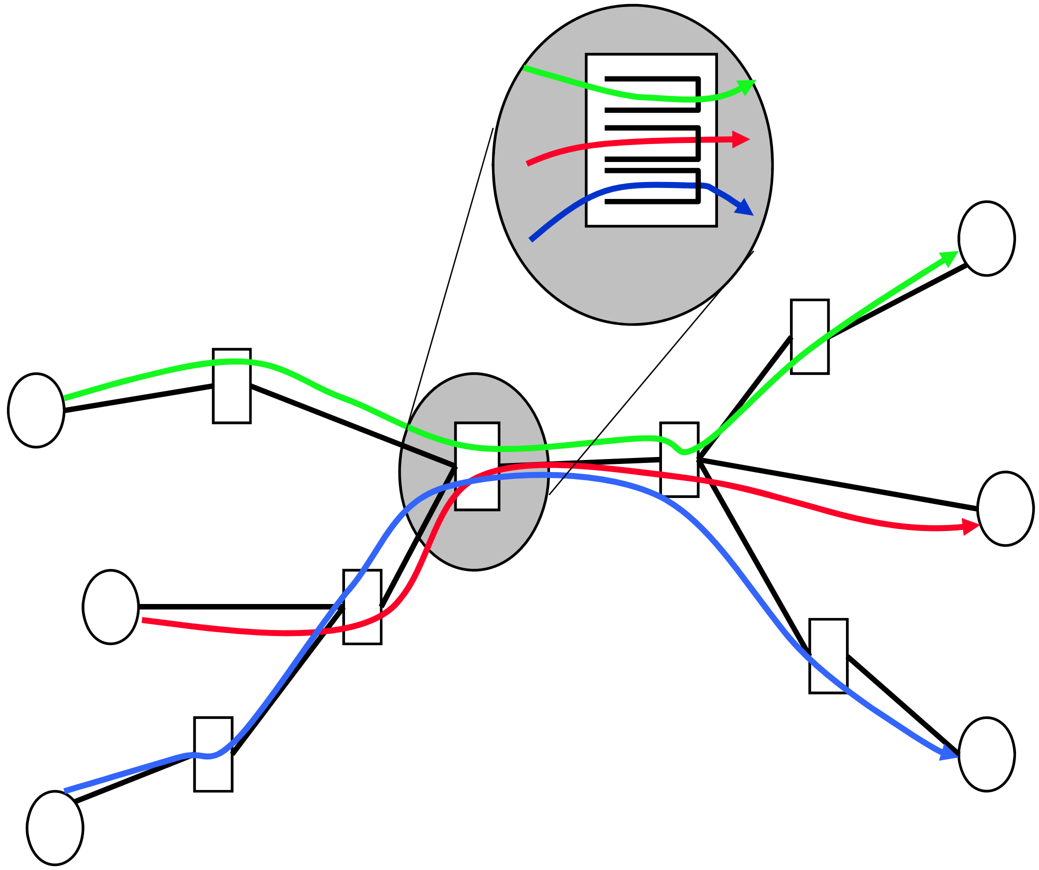
- Goal: provide support for wide variety of applications:
  - Interactive TV, IP telephony, on-line game (distributed simulations), VPNs, etc
- Problem: deal with network congestion
- During congestion all packets are treated the same
  - All packets get the same delay
- Only control possible at end-hosts
  - Feedback loop too large (e.g., 100s of ms) for real-time applications (e.g., interactive communication)
  - Trust issue → how can you trust users that will react properly in case of congestion?

A Solution: Integrated Services

- 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

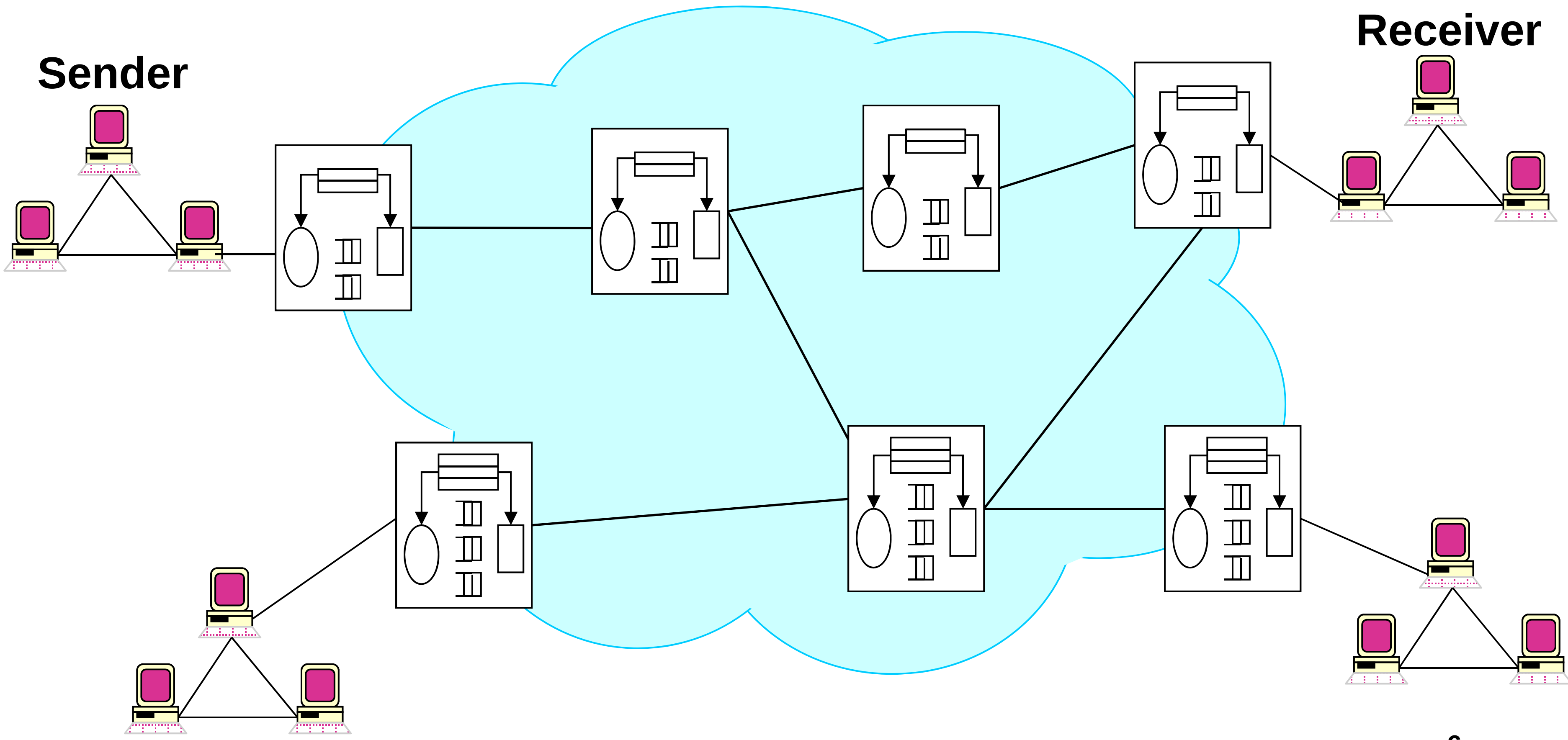
Integrated Services Network

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



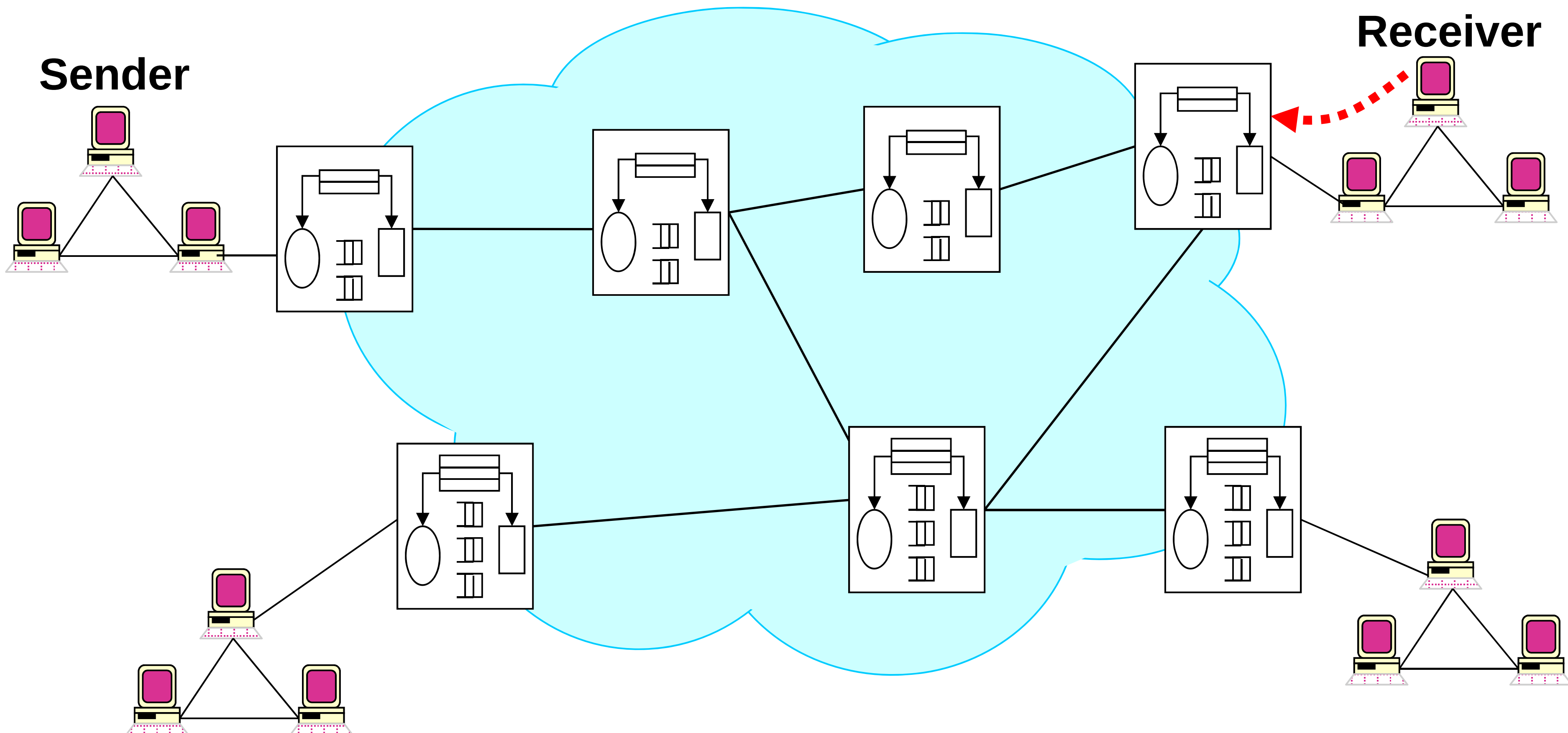
Integrated Services Example

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



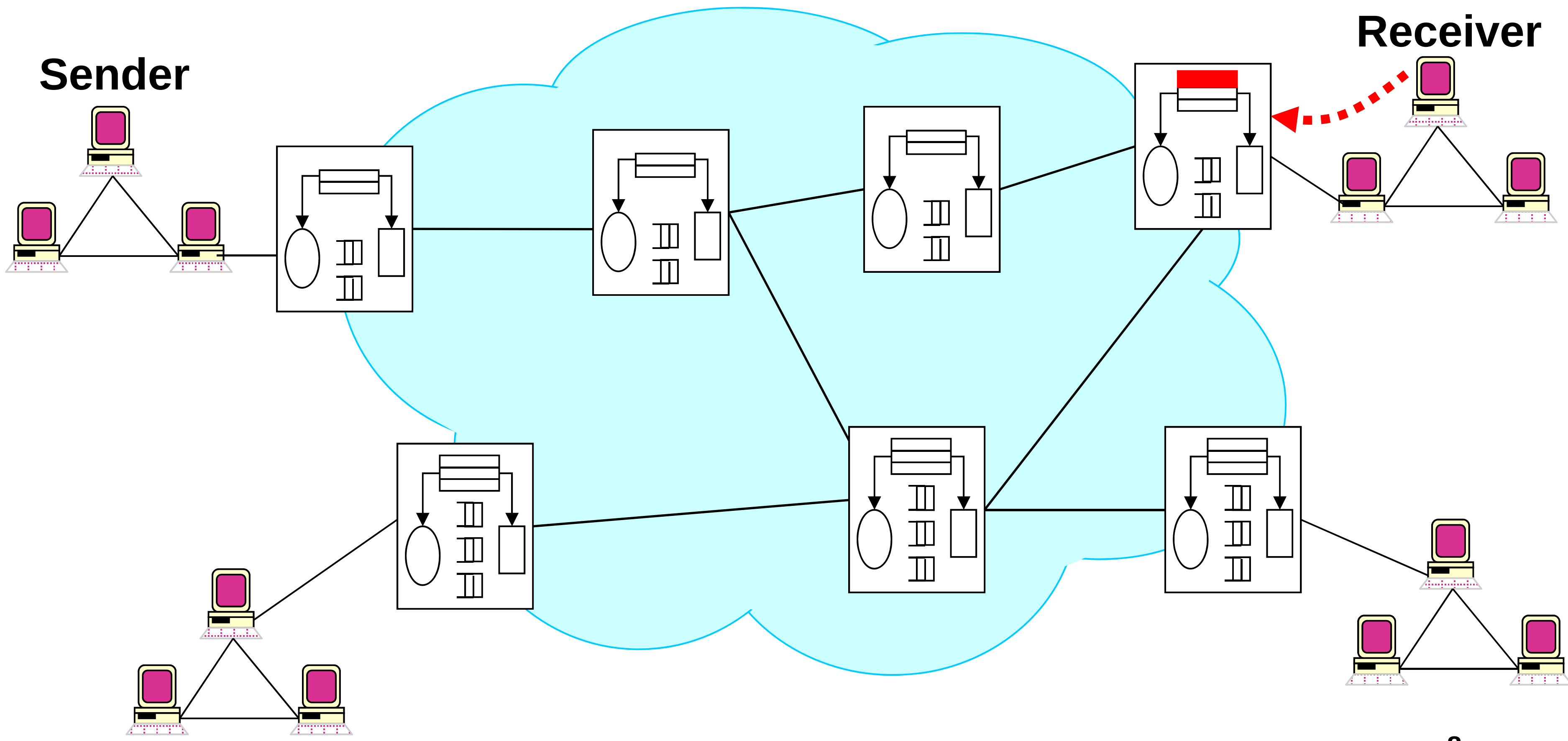
Integrated Services Example

- Allocate resources - perform per-flow admission control



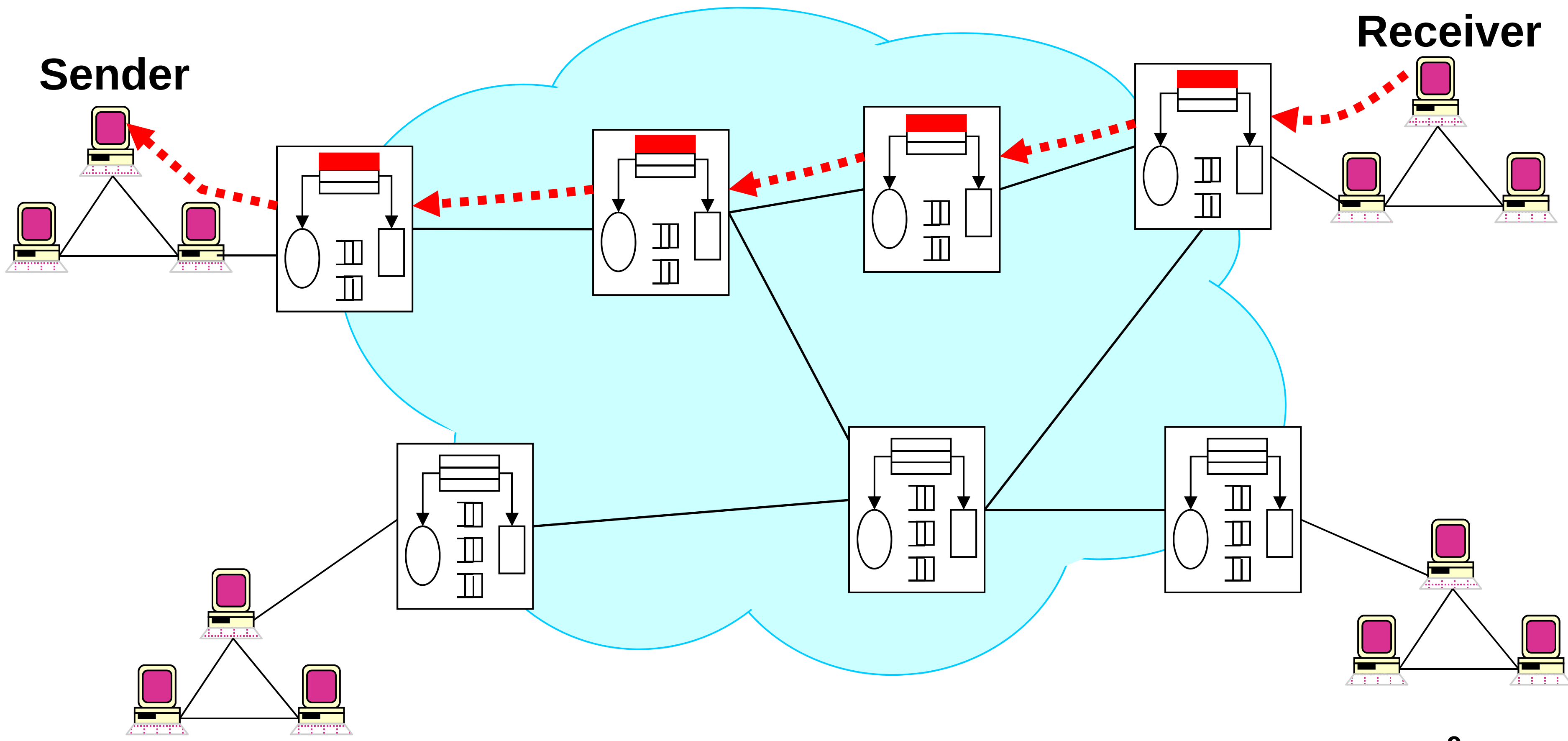
Integrated Services Example

- Install per-flow state



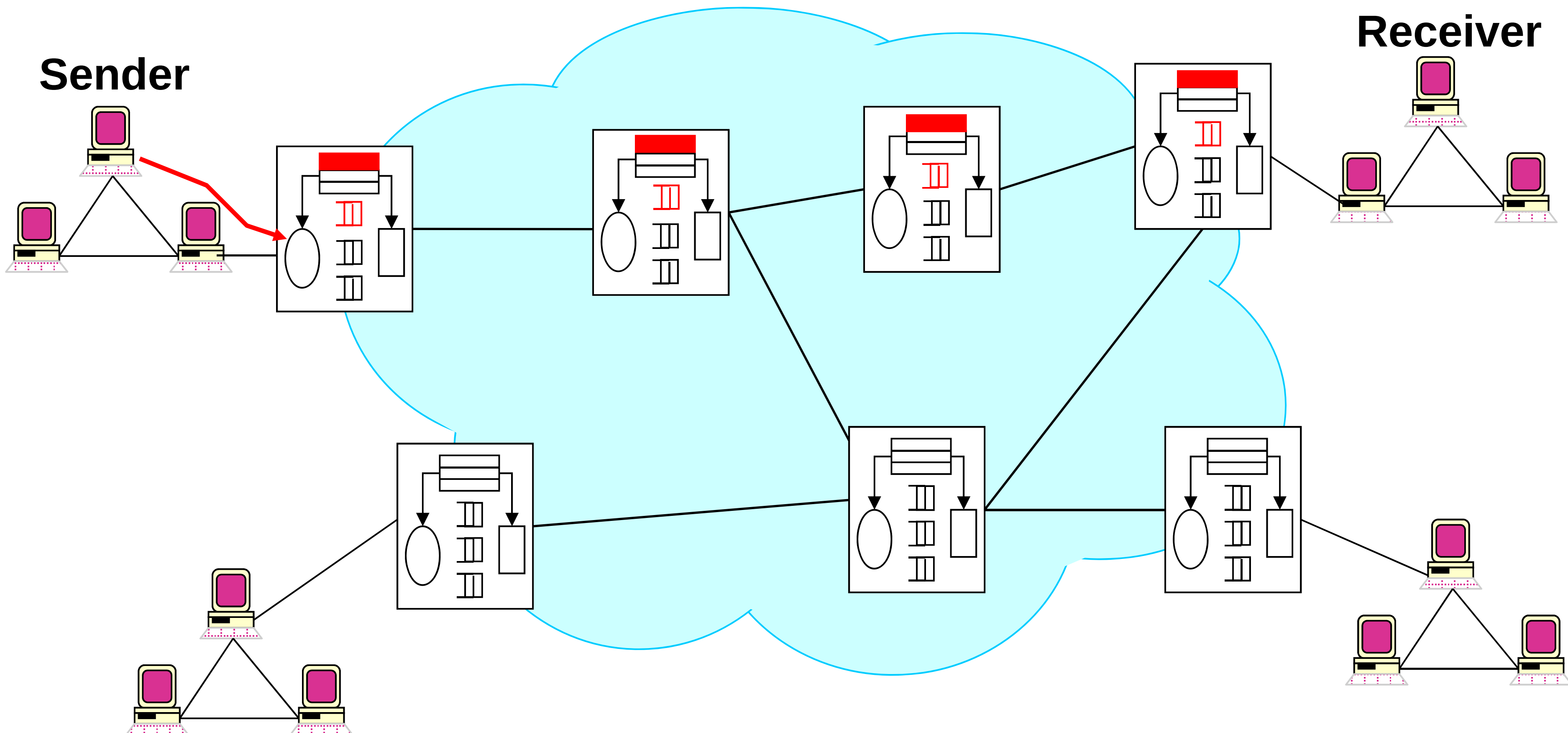
Integrated Services Example

- Install per flow state



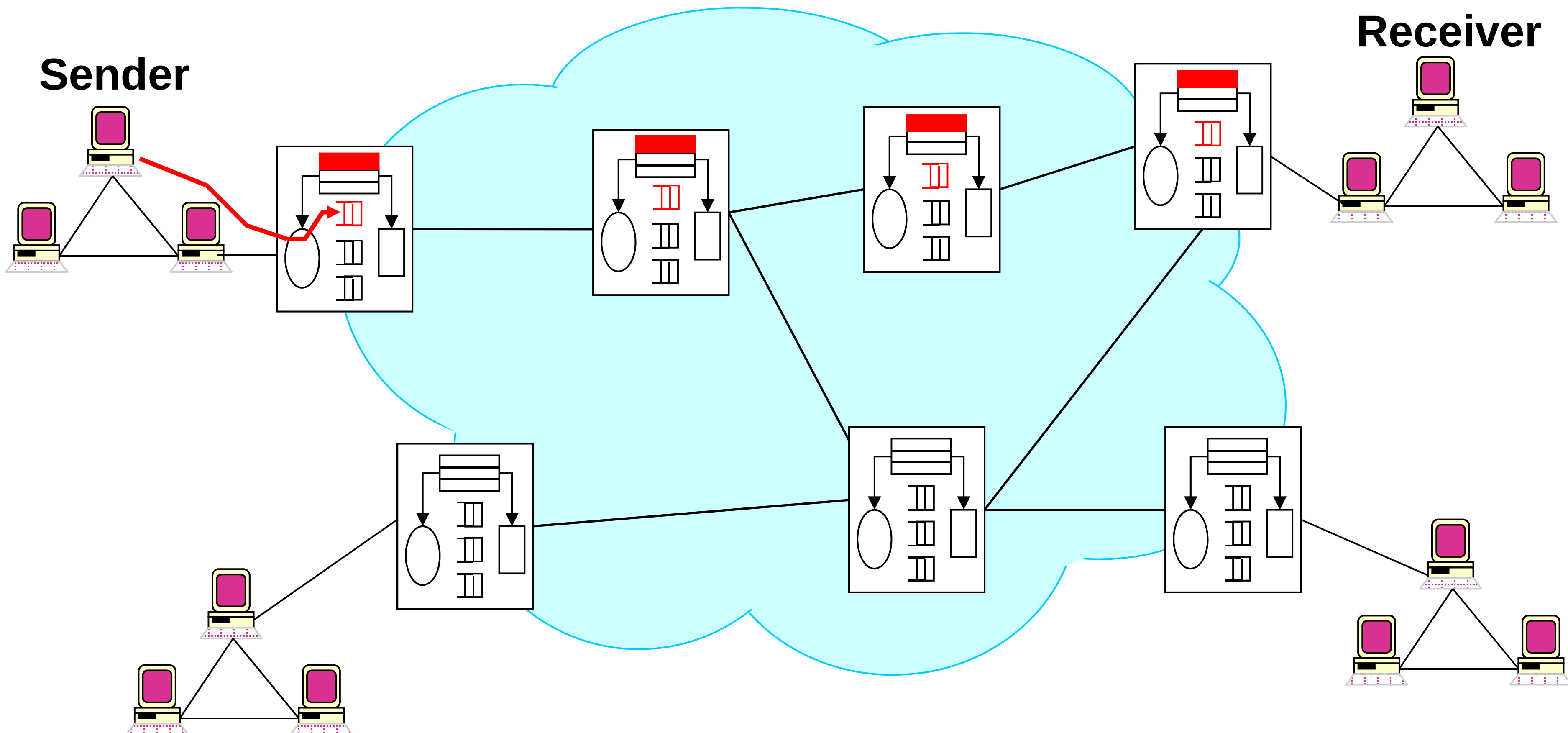
Integrated Services Example: Data Path

- Per-flow classification



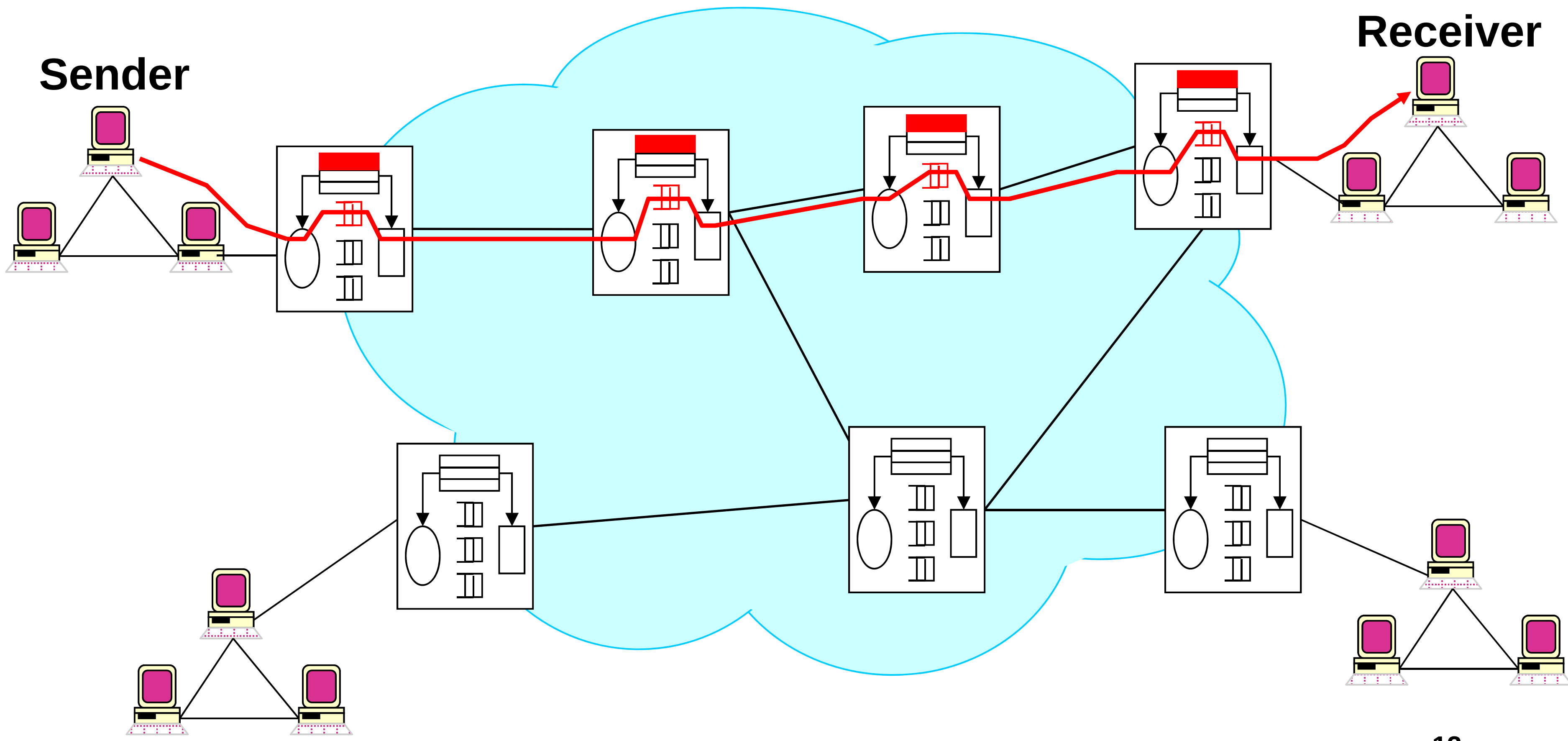
Integrated Services Example: Data Path

- Per-flow buffer management

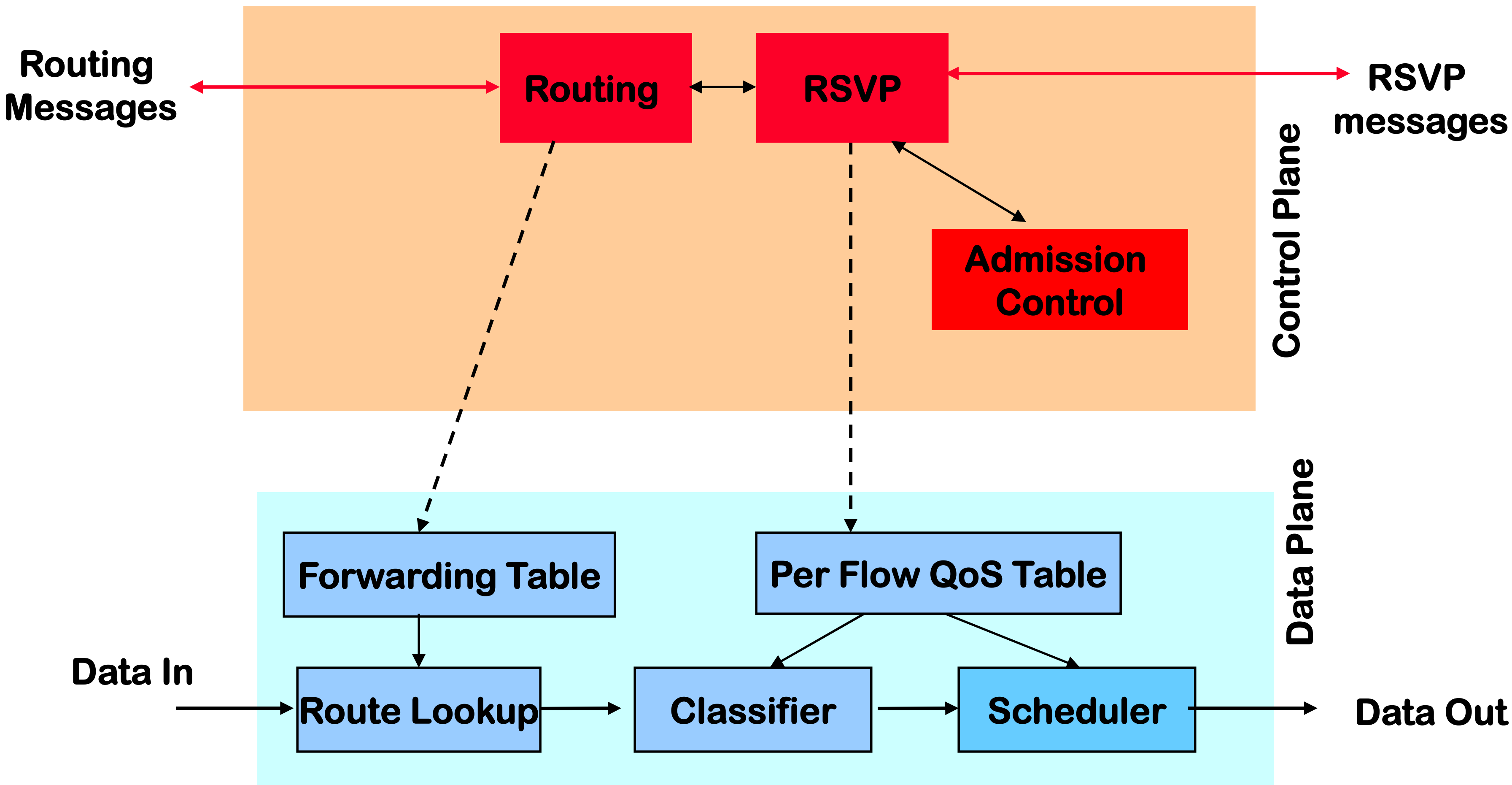


Integrated Services Example

- 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

- RSVP: 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

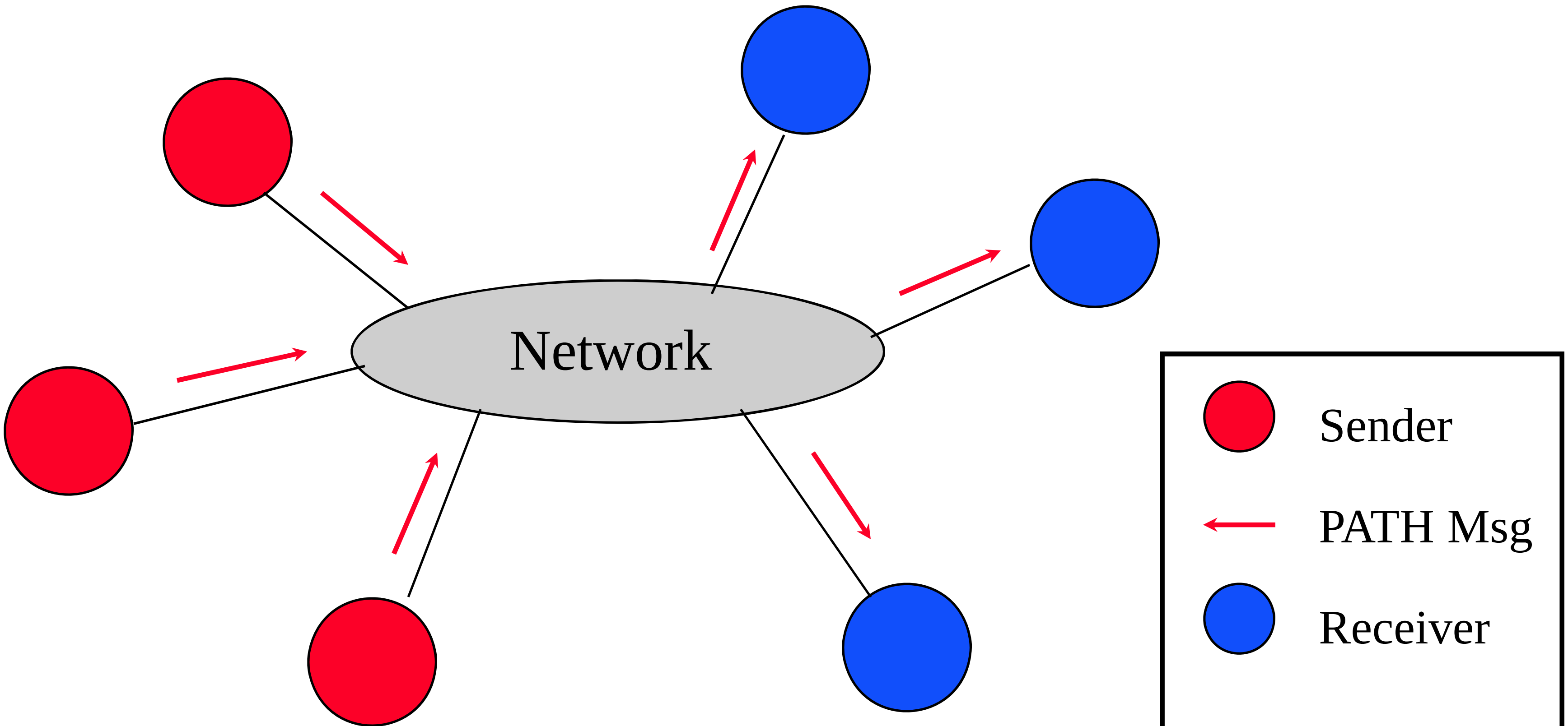
IP Multicast

- Best-effort M x N 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 in a distributed manner
  - no single node knows the membership
  - tough problem
  - various solutions: DVMRP, CBT, PIM

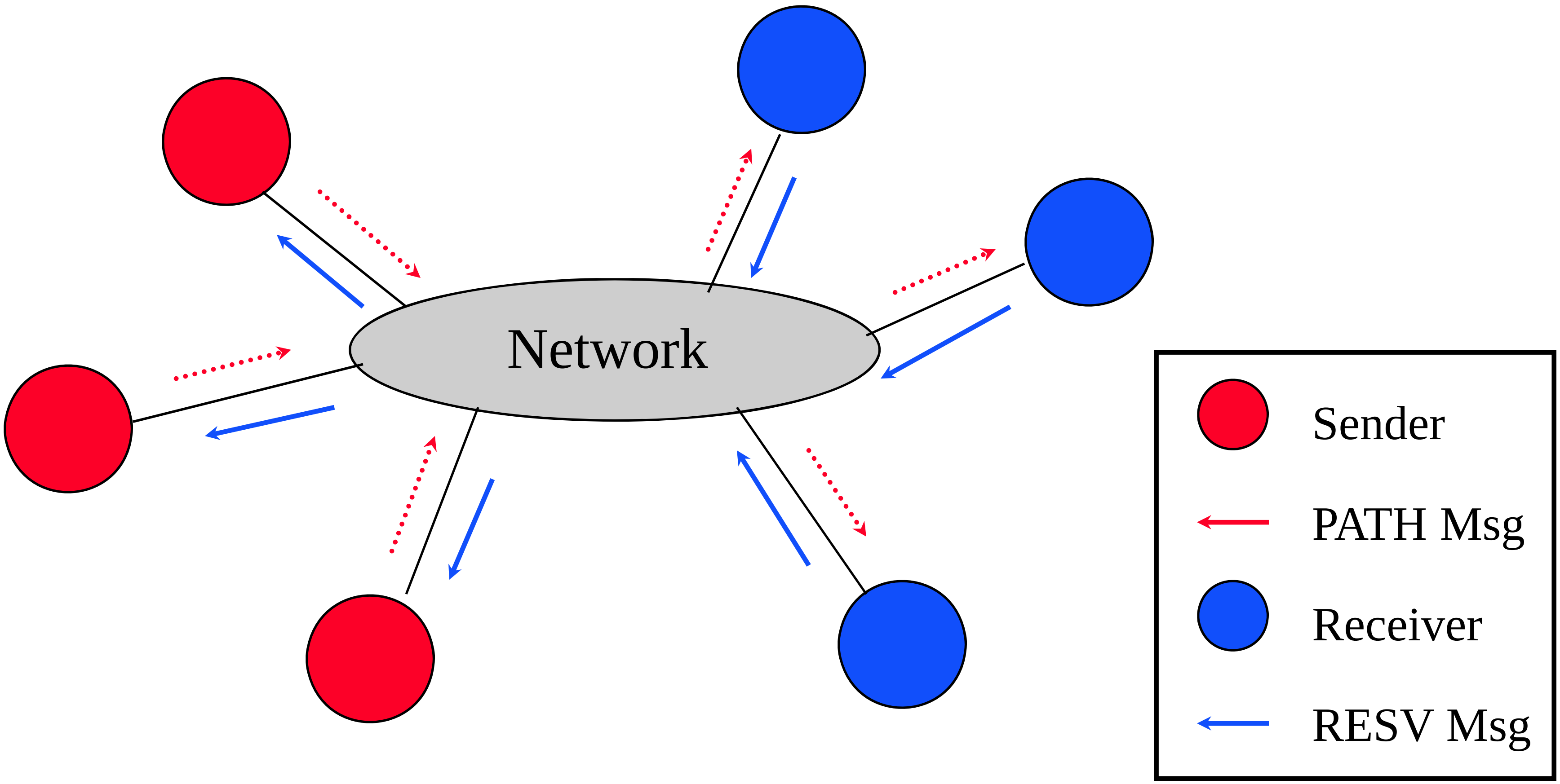
RSVP Reservation Model

- 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
- Multiple senders and receivers can be in session

The Big Picture



The Big Picture (2)



Resource ReserVation Protocol

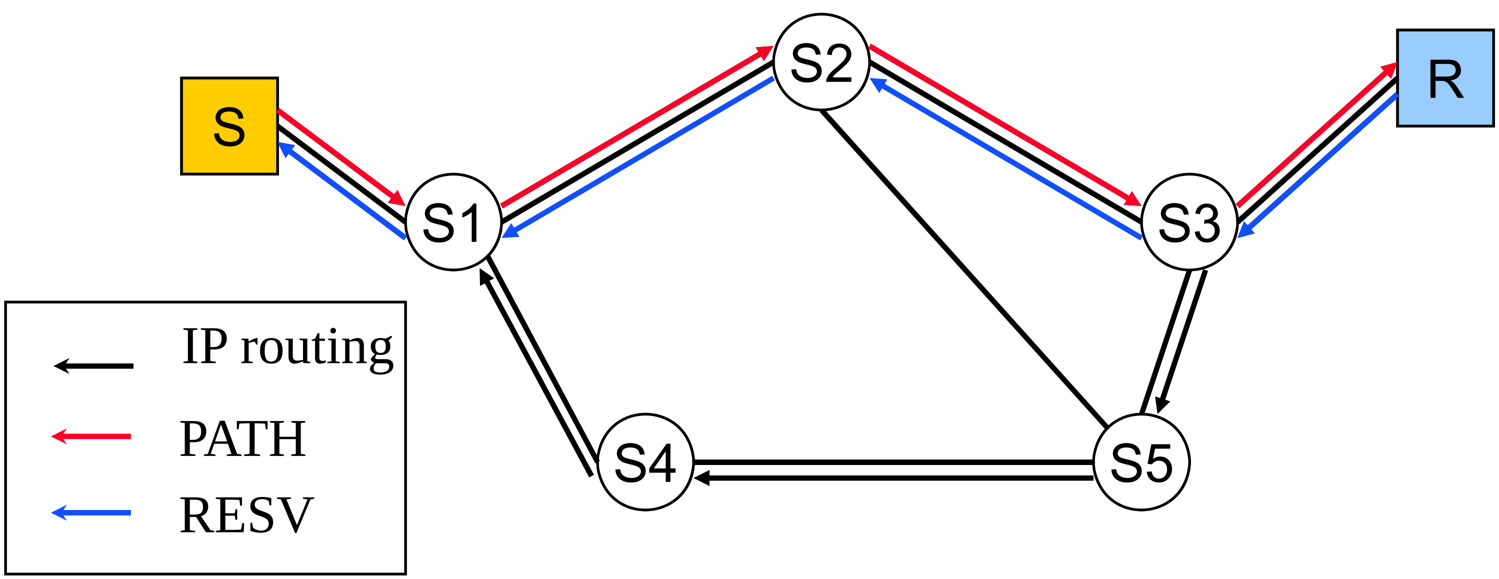
- Extends current best-effort Internet model and supports QoS over IP unicast and multicast.
- Is a signaling protocol to install and maintain reservation state information in the Integrated Services architecture.
- Carries traffic spec and path information from a sender to receivers and reservation information in the other direction.

Attributes of RSVP

- RSVP is simplex. It makes reservations for unidirectional data flows.
- RSVP is receiver-initiated. The receiver of a data flow initiates and maintains the resource reservation.
- RSVP uses soft states and adapts dynamically to changing group membership as well as changing routes.
- RSVP is not a routing protocol. It relies on a routing protocol to provide a route/tree along which it sends control messages to make reservation.
- RSVP design is independent of routing, admission control, and packet scheduling.
- RSVP provides transparent operation through routers that do not support it.

Why Path Message?

- 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



PATH Message

- PATH message via the data delivery path:
  - Flowspec: traffic specification (e.g., use token bucket)
  - F-flag: specify whether filtered reservation is allowed
- Routers store:
  - Path state, i.e., address to previous hop
  - If F-flag is set, store sender and its flowspec
  - Otherwise, just add new link to multicast tree

RESV Message

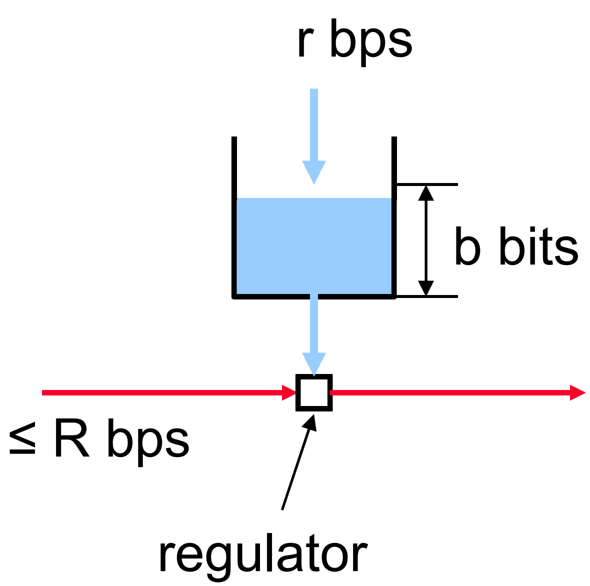
- RESV message via the reverse path established by PATH:
  - Flowspec: source traffic specification and desired QoS (e.g., queueing delay)
  - Packet filter: what packets can use the reservation
- Routers perform:
  - Admission control
  - Update reservation state on outgoing link:
    - Amount of reserved resources
    - Source filter
    - Reservation style: how senders can share reservation
    - If style dynamic filter, store reservation initiator

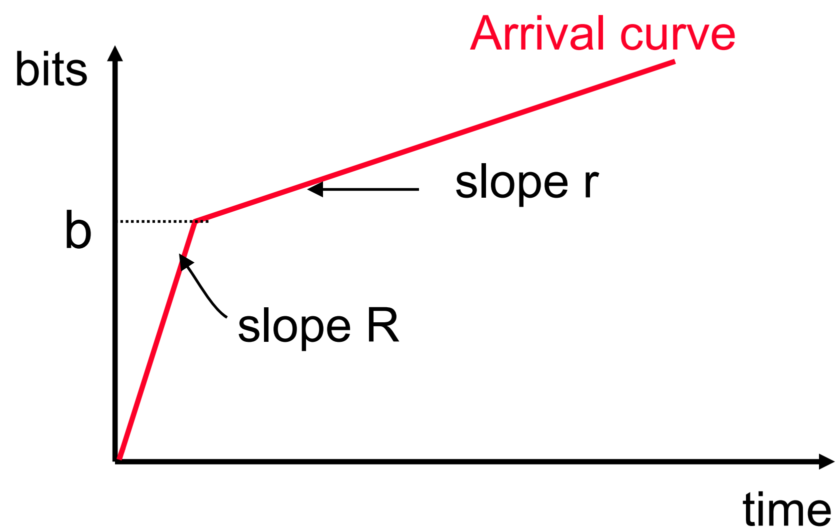
Comments

- Receiver initiated reservation
- Decouple the routing from reservation
- Two types of state: path and reservation

Flow Specification: Token Bucket

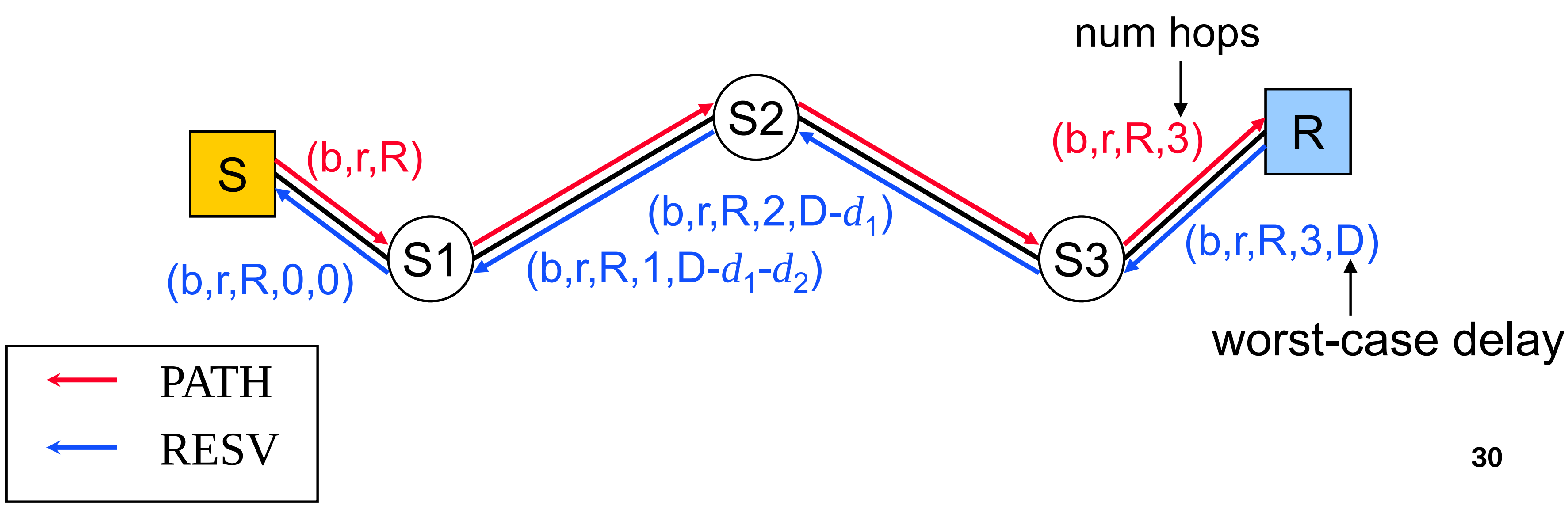
- Characterized by two parameters (r, b)
  - r – average rate
  - b – token bucket depth
- Assume flow arrival rate ≤ R bps (e.g., R link capacity)
- A bit is transmitted only when there is an available token
- Arrival curve – maximum amount of bits transmitted by time t





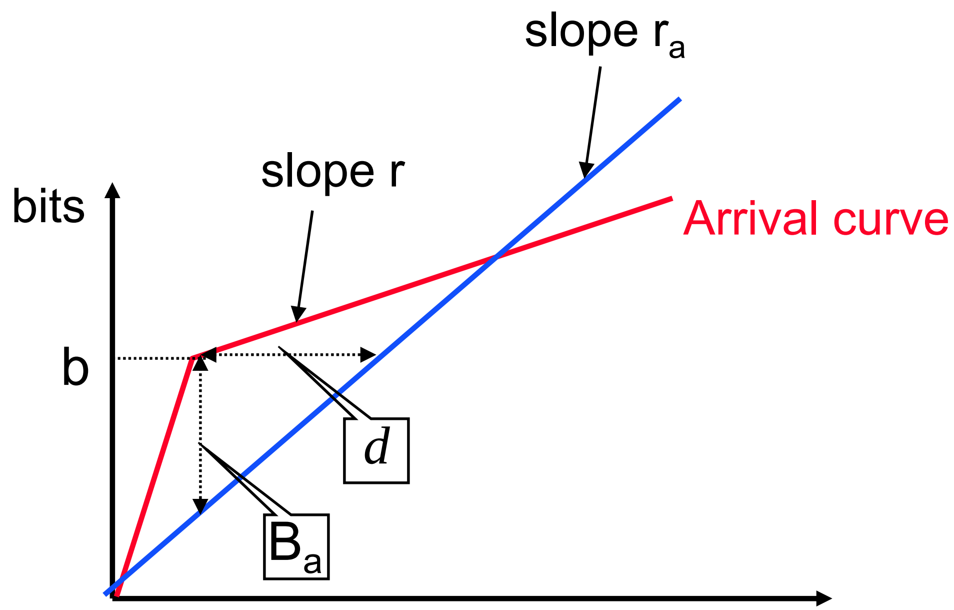
End-to-End Reservation

- 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 (no-filter): does not specify any sender; all packets from senders 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 (shared explicit): resource shared by senders that are (explicitly) specified
    - Sources can be modified over time

Reservation Styles

- Wildcard-Filter (WF) Style -- WF(*{Q})
  - Creates a single reservation shared by flows from all upstream senders belonging to the unicast (multicast) session.
  - The reservation propagates towards ALL sender hosts and automatically extend to new senders as they appear.
- Fixed-Filter (FF) Style -- FF(S{Q}), FF(S1(Q1), S2(Q2), …)
  - Creates distinct reservations for data packets from a particular sender, not sharing them with other sender' packets for the same session.
- Shared-Explicit (SE) Style -- SE((S1, S2, …) {Q})
  - Creates a single reservation shared by selected upstream senders.
  - Allows a receiver to explicitly specify the set of senders to be included.

RSVP Reservation Styles

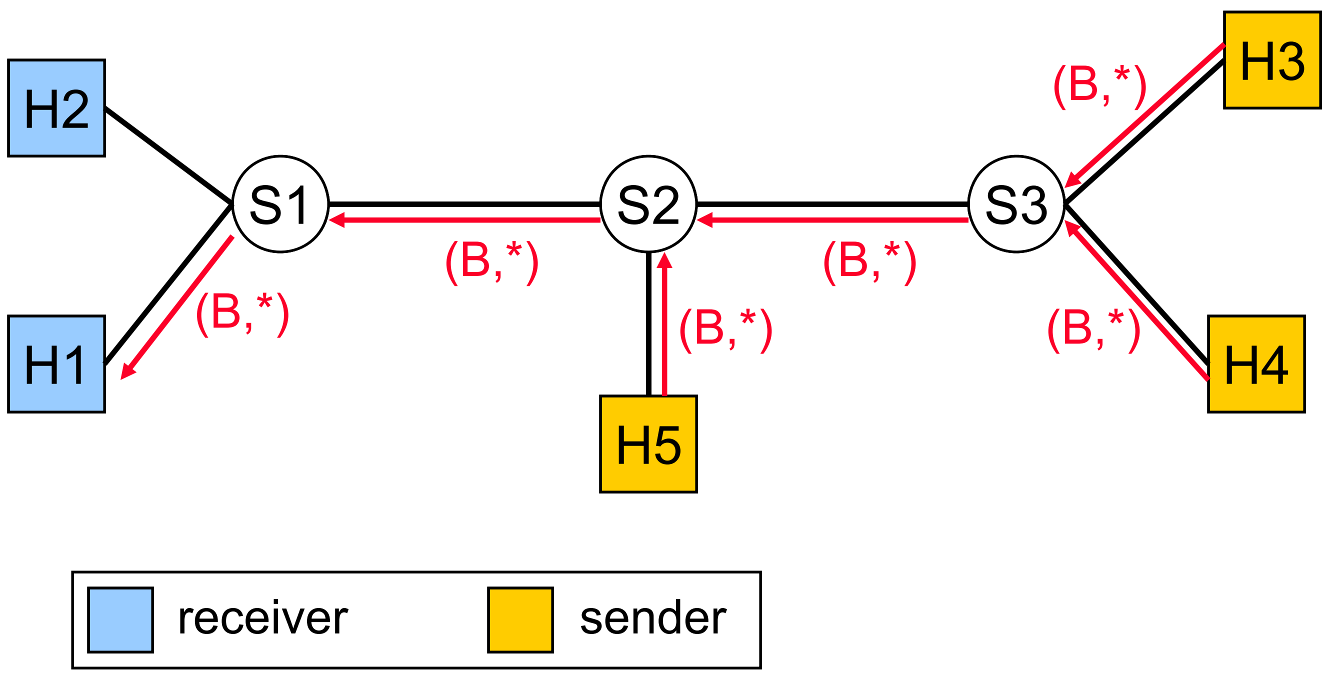
	Reservations	
Sender Selection	Distinct	Shared
Explicit	Fixed Filter	Shared Explicit
Wildcard	None	Wildcard Filter

•Shared reservations (WF and SE) are appropriate for multicast applications in which multiple sources are unlikely to transmit simultaneously (packetized audio for example).

•FF style makes separate reservations for each sender, and is appropriate for video applications.

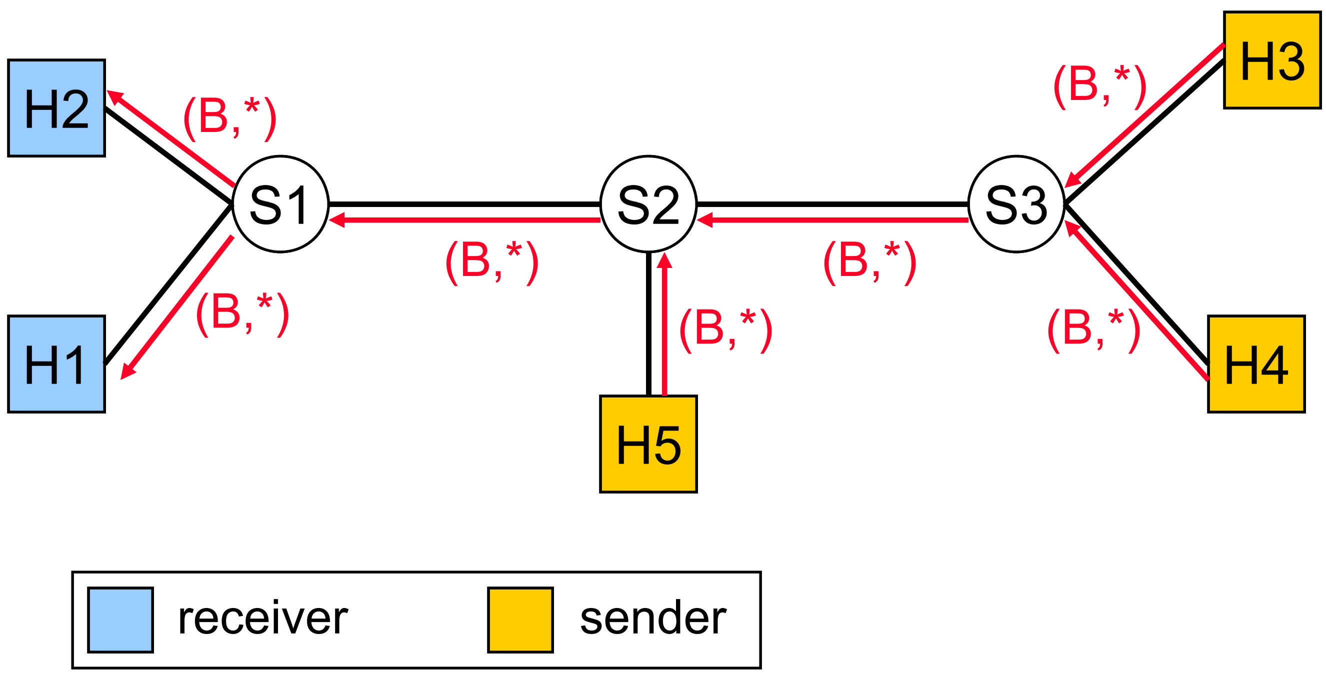
Wildcard Filter Example

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



Wildcard Filter Example

- H2 reserves B

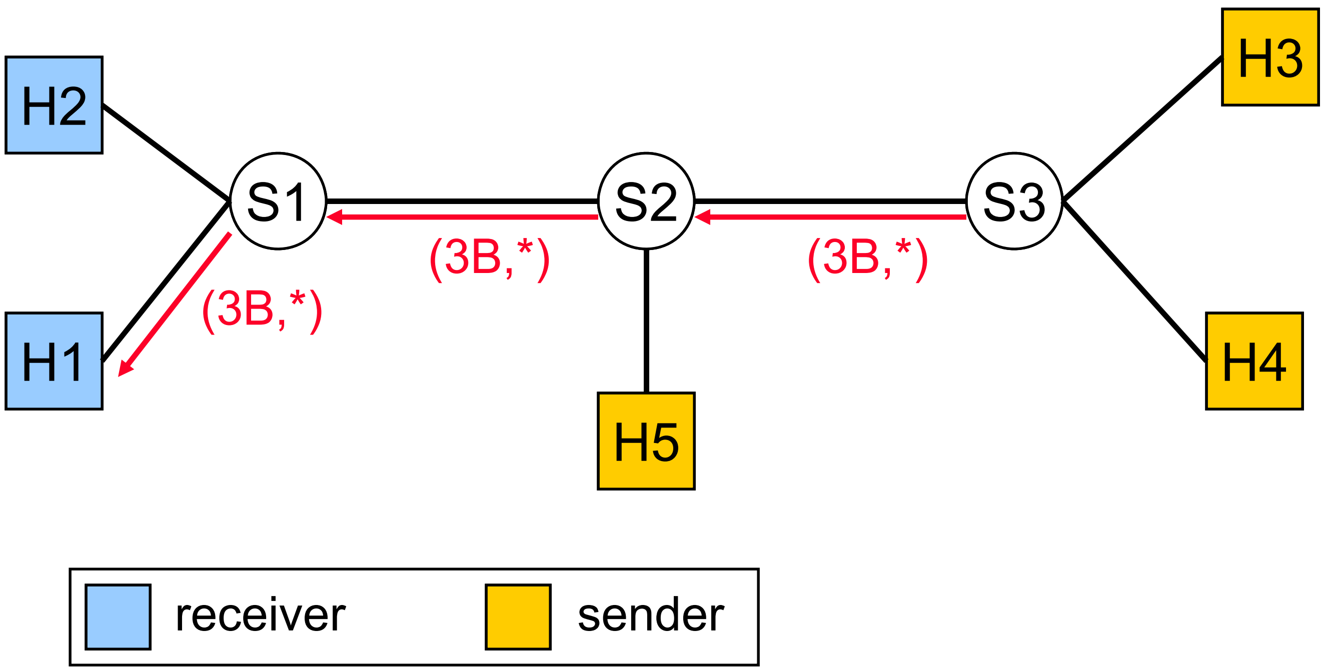


Wildcard Filter

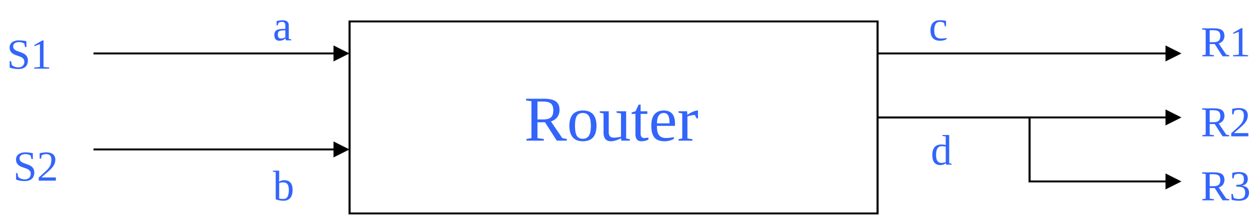
- Advantages
  - Minimal state at routers
    - Routers need to maintain only routing state augmented by reserved bandwidth on outgoing links
- Disadvantages
  - May result in inefficient resource utilization

Wildcard Filter: Inefficient Resource Utilization Example

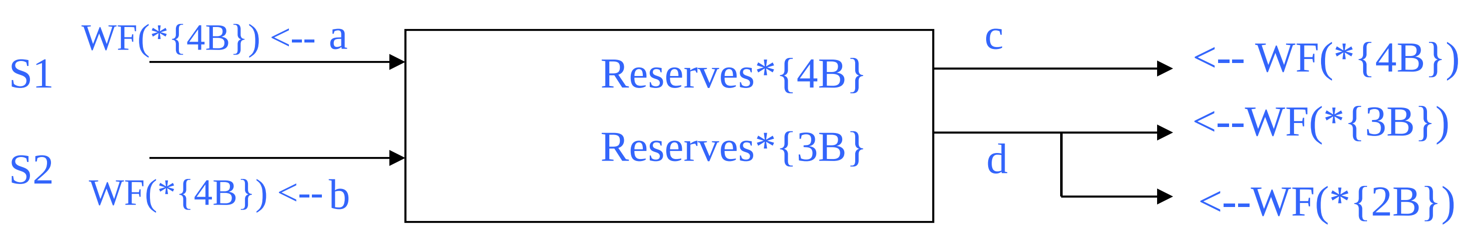
- H1 reserves 3B; wants to listen from all senders simultaneously
- Problem: reserve 3B on (S3:S2) although 2B sufficient !



Reservation Style Examples



Router Configuration

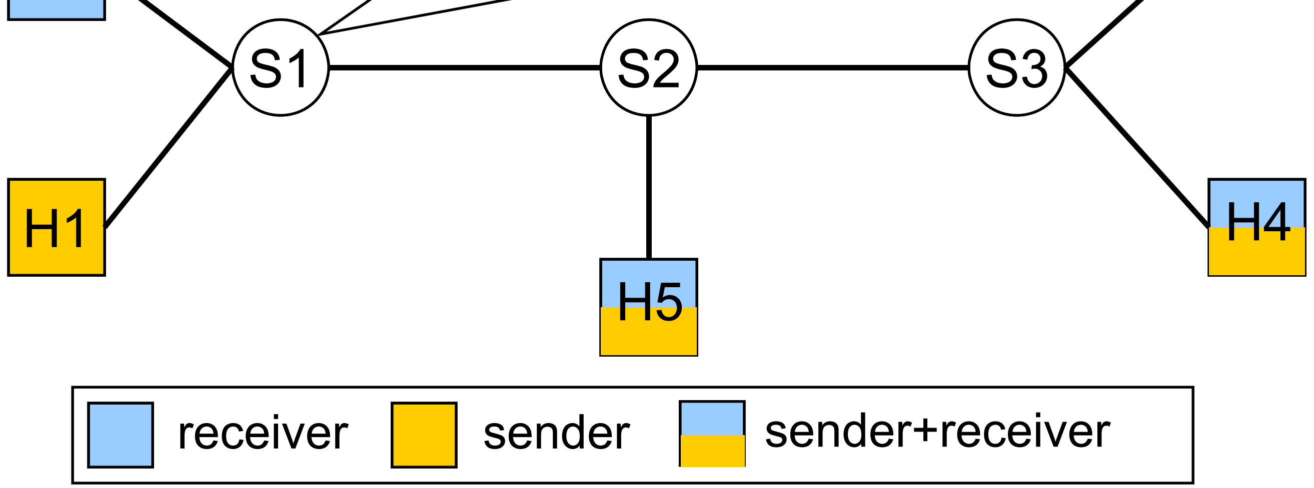


Wildcard-Filter Reservation Example

Fixed Filter Example

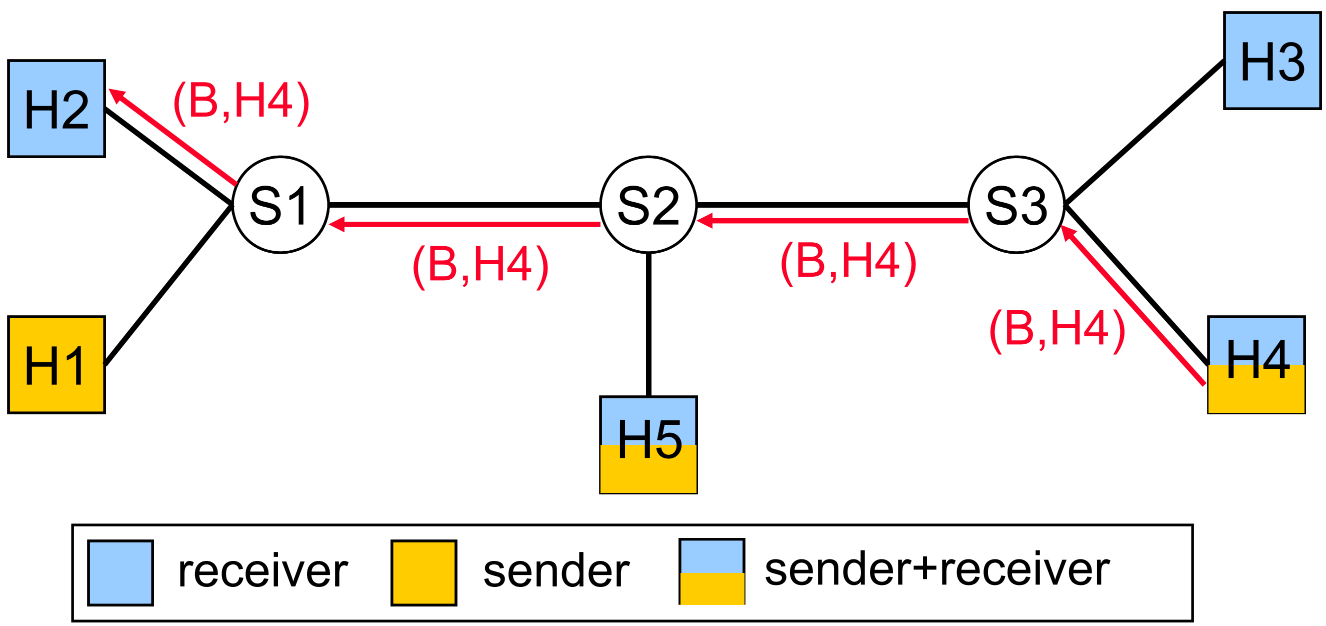
- Receivers: H2, H3, H4, H5; Sender: H1, H4, H5
- Routers maintain state for each receiver in the routing table
- Entries for receivers H3, H4, H5; and H2

NextHop	Upstreams(sources)
S2	H1(H1)
H2	H1(H1), S2(H5, H4)

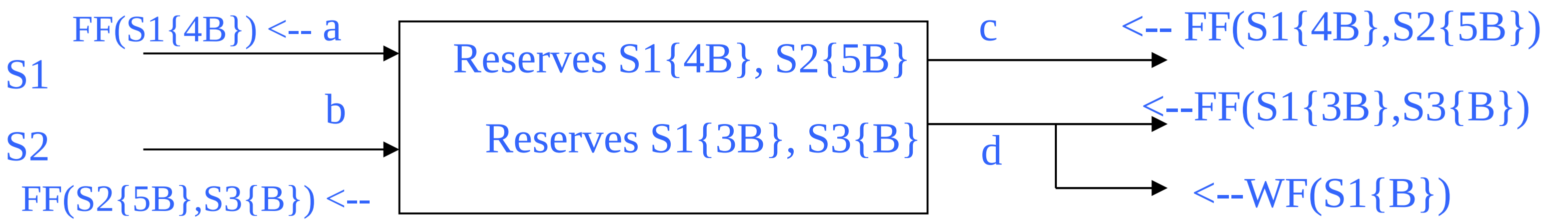


Fixed Filter Example

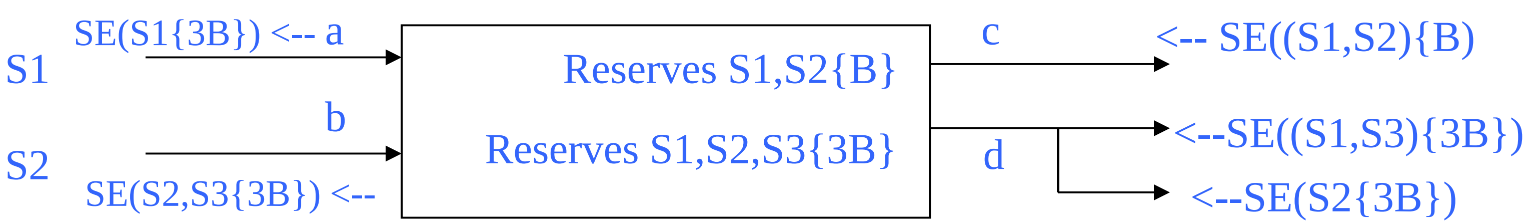
- H2 wants to receive B only from H4



Reservation Style Examples (Continued)



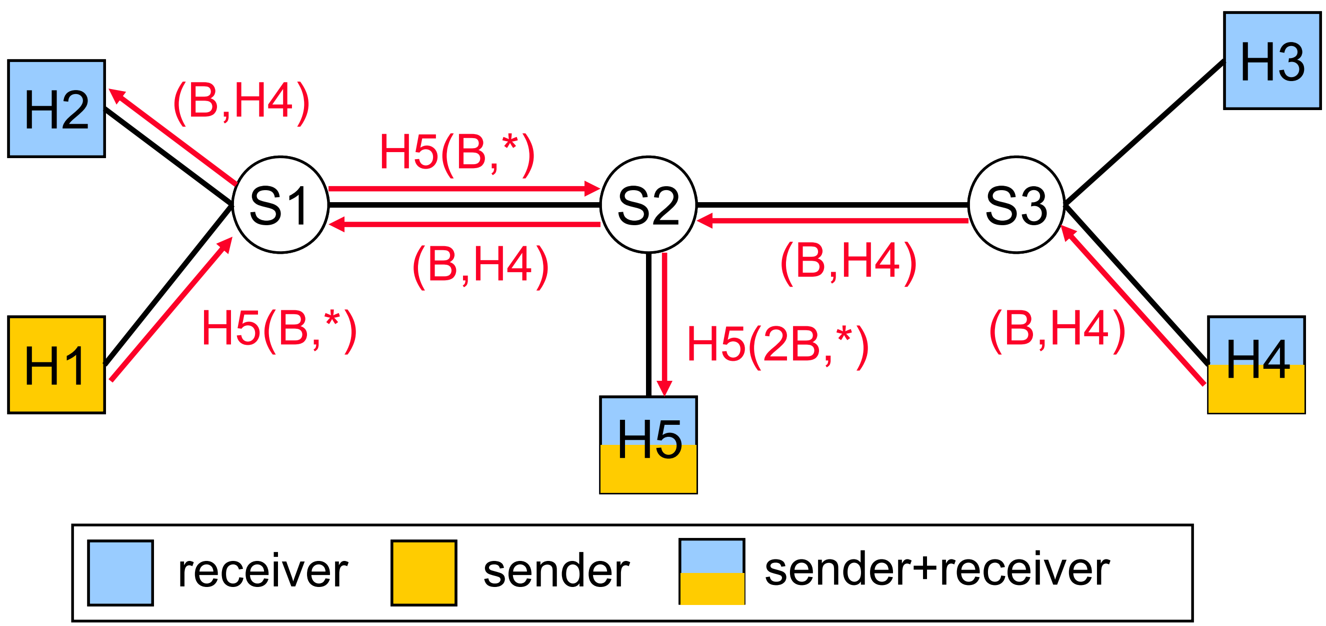
Fixed-Filter Reservation Example



Shared-Explicit Reservation Example

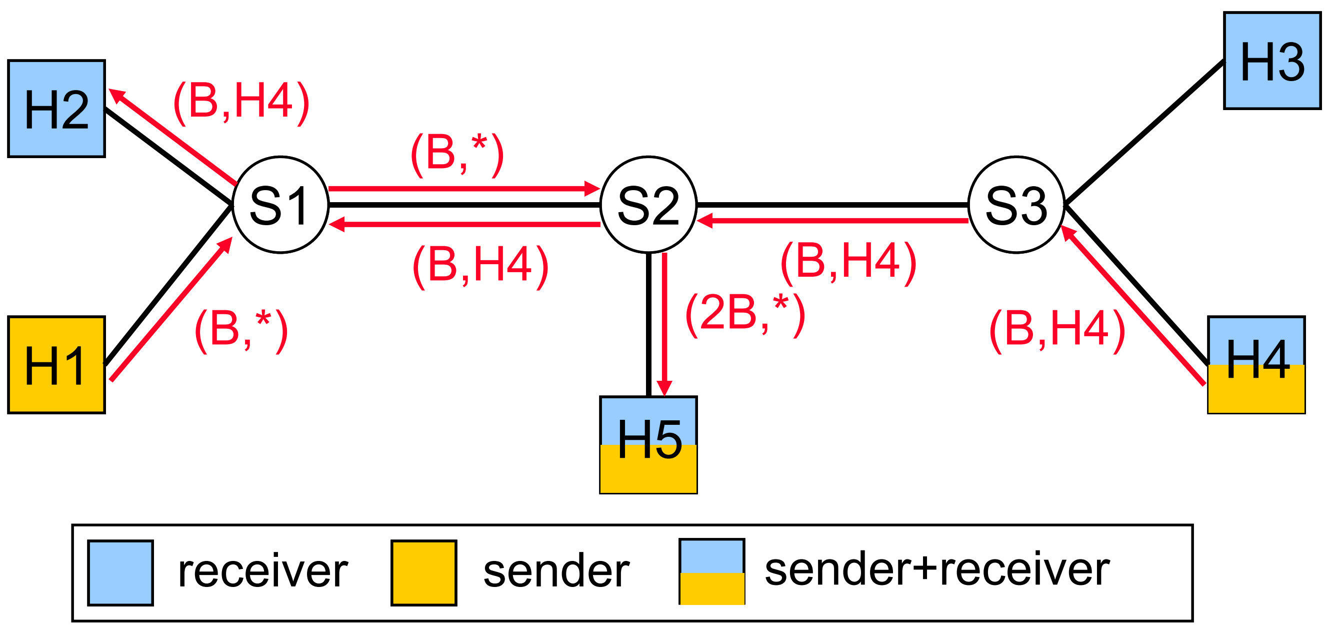
Dynamic Filter (Shared Explicit) Example

- H5 wants to receive 2B from any source



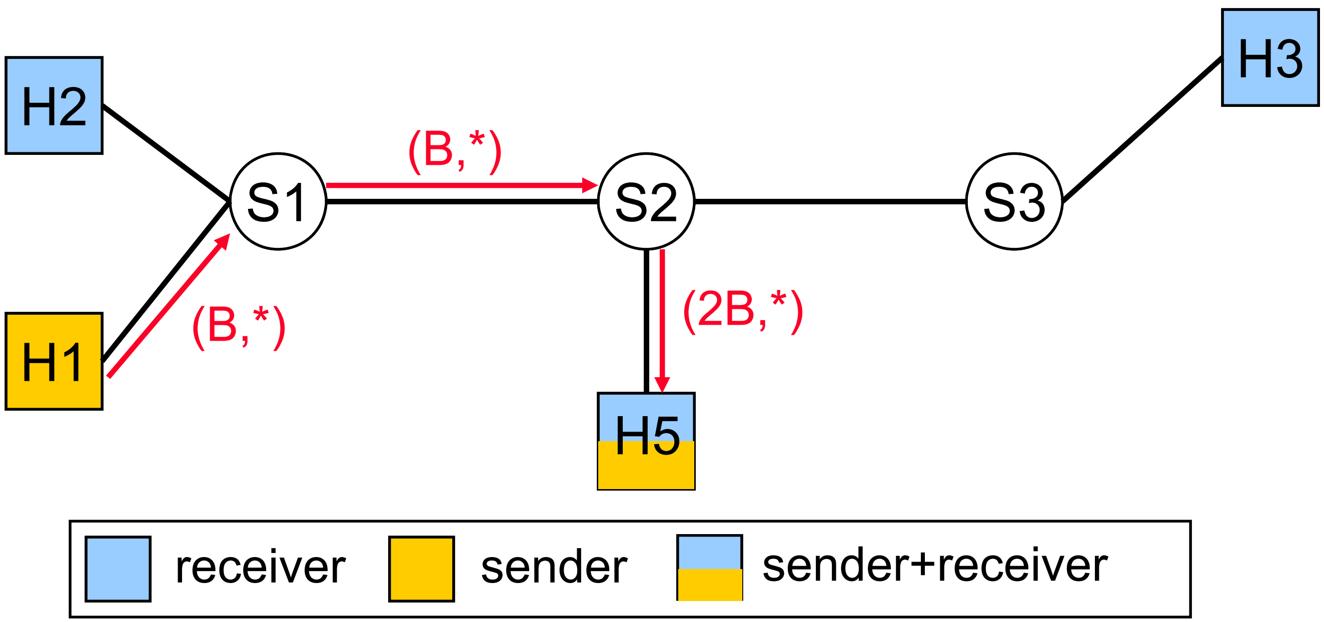
Tear-down Example

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



Tear-down Example

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



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 tear-down 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
  - 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