QoS and Resource Reservation Protocols

Review of IP Unicast Routing

- Routes determined by routing protocols based on minimizing metrics.
- Destination based: routers forward to next hop based on destination IP address
- Routes are sink trees: one tree per destination
- Two routing protocols are used:
  - RIP
  - OSPF

RIP in brief

- Calculates best paths using Bellman-Ford algorithm
- Slower route convergence compared to OSPF
- Single metric: hop count

OSPF in brief

- Calculates best paths using Dijkstra's shortest path first algorithm
- Designed for fast route convergence (compared to RIP)
- Can utilize multiple performance metrics
- Originally designed to support use of TOS bits in IP header

Review of IP Multicast Routing

- Protocols include:
  DVMRP, MOSPF, CBT, PIM-SM, PIM-DM
- Differ by:
  - The multicast routes established
  - State information kept in routers
  - Information exchanged among routers

QoS Support in the Internet

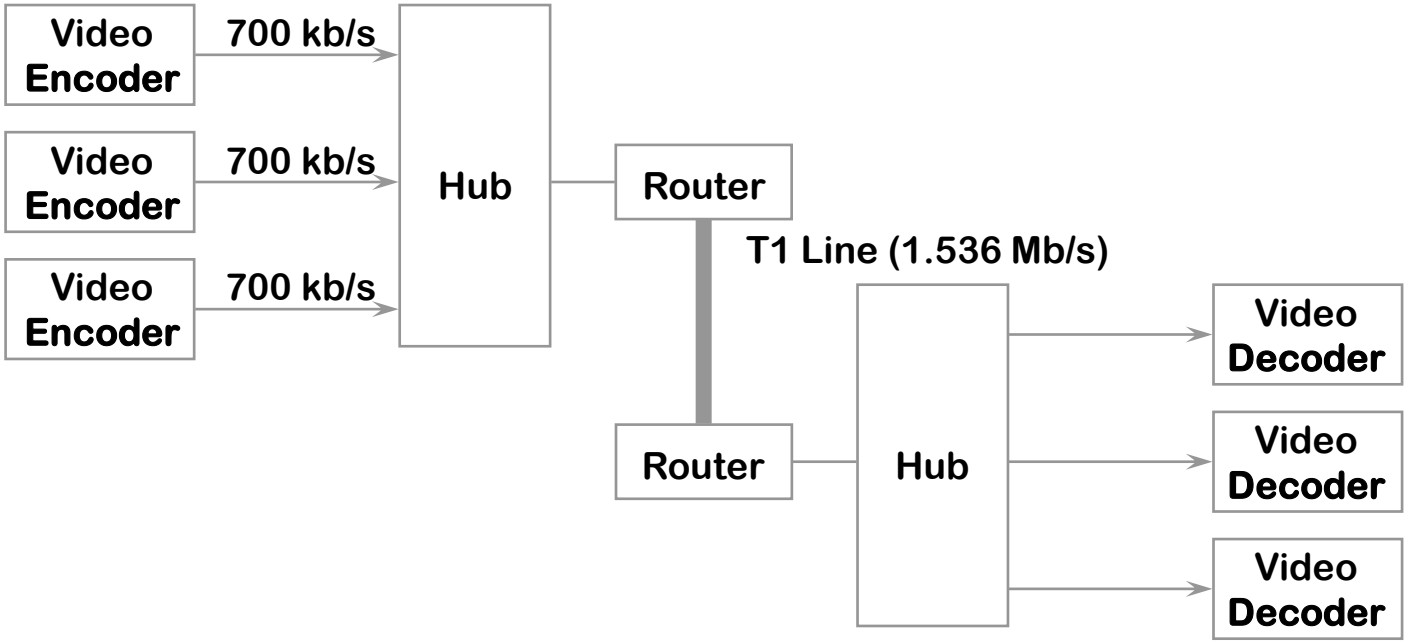
Motivation

- Some applications use network services but almost don't care about performance aspects of these services
  - Examples: file transfer, printing.
- Some applications have performance requirements on the traffic they generate:
  - ***They will not work if these requirements are not met!***
  - Example: video transmission requires a certain available bandwidth.

Why is this an issue?

- Traditional data networks have been designed based on "best effort delivery":
  - *The network will try its best to deliver your data, but it offers no guarantees.*
- Traditional data applications:
  - *use protocols such as TCP to deal with data loss and unknown network available bandwidth;*
  - *tend to generate data in "bursts";*
  - *normally do not require any kinds of guarantees from the network*
- Traditional networks ***have not*** been designed with Quality Of Service (QOS) in mind

Where do things break?



- Things break when there is a contention for resources.
- Service may be unacceptable for all

Example of a Network with QoS
(the phone network)

- Before you can communicate, you need to negotiate a channel with the network
- If the network does not have the resources, or the destination is unavailable, your call is not completed
- If your call is completed, you "own" the circuit for the duration of the call, regardless of the additional traffic in the network
- Your circuit has appropriate capacity for your voice requirements.

QoS Components

- The nodes need to know their ***Traffic Characteristics*** and ***Traffic Requirements*** to describe them to the network.
- There must be a ***Signaling Protocol*** between the nodes and the Network Management.
- The Network must implement ***Admission Control***; connections exceeding the available bandwidth must be rejected.
- The Network may also implement ***Traffic Policing***, to make sure that the traffic emitted by the nodes conform to the agreed parameters.

IP Resource Reservation Protocols

- Provide a mechanism to allow hosts to signal their QoS requirements to the network.
- Provide a mechanism for the network to indicate whether or not to accept the reservation.
- Provide mechanisms for the network to keep state regarding the reservation
- Independent of:
  - Routing protocol; use the routes that are already there.
  - Actual traffic characteristics (opaque descriptors).
  - Policy decisions.

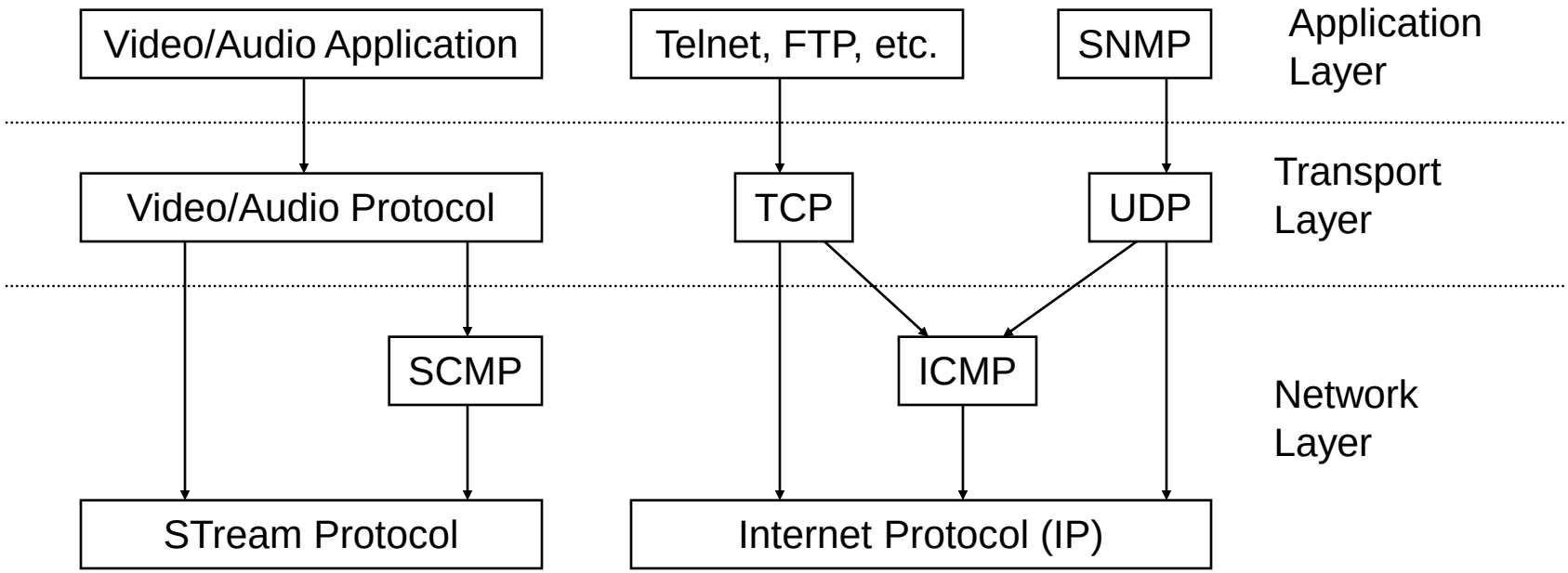
Reservation Protocols

- Internet Stream Protocol Version 2 (ST2+)
  - RFC 1819 (read sections 1, 2, 3 and 9)
- Resource ReSerVation Protocol (RSVP)
  - RFC 2205 (read sections 1 and 2)
- Suggested reading:
  - L. Zhang et al, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, vol. 7, no. 5, Sept. 1993, p.p. 8-18.
  - L. Delgrossi et al, "Reservation Protocols for Internetworks: A Comparison of ST-II and RSVP," *Fourth Int. Workshop on Network and Operating System Support for Audio and Video*, Lancaster, UK, Nov. 1993.

ST-2 Protocol History

- First version was published late ‘70s and used throughout the ‘80s for experimental transmission of video, voice and distributed simulation.
- Protocol was revised (became ST-II) and published as RFC 1190 in October 1990.
- Commercial products available.
- IETF Working Group created in 1993.
- WG cleaned up specs and and published a revised version in 1995 (ST2+), RFC 1819.

Protocol Stack



SCMP: Stream Control Message Protocol

ST-2 and IP

- ST-2 exists at the same level as IP.
- ST-2 uses the same addressing scheme as IP, standard ARP for address resolution, and the same layer 2 SAPs as IP.
- ST-2 and IP packets differ in the first four bits of the header:
  - IP headers are version 4 (0100)
  - ST-2 headers are version 5 (0101)
- ST-2 has a Control Protocol (SCMP) that is encapsulated into ST packets.

ST-2 Communication Model

- Two-step process:
  - First step: data channels are built and resources are reserved; no real-time constraints.
  - Second step: actual real-time data transfer.
- Architectural modules:
  - ***Data transfer protocol (ST)***
  - ***Setup protocol (SCMP)***
  - ***Flow specification, to indicate traffic characteristics and requirements (ST2+ FlowSpec)***
  - Routing function
  - Local resource manager in each node
- The highlighted modules are provided by ST-2.

ST-2 Modules

- Data Transfer Protocol (ST)
  - Defines the packet format.
  - Packets contain a globally-unique stream identifier.
  - No defined error control procedures.
- Setup Protocol (SCMP)
  - Responsible for establishing, maintaining and releasing real-time streams.
  - Request-response protocol, with retransmissions for reliability.
  - Relies on the underlying routing function to select paths.

ST-2 Modules, cont.

- Flow Specification
  - Expresses the traffic characteristics and requirements from the source.
  - Includes some negotiable parameters, and some "measured" parameters along the route.
  - Transported by the setup protocol (SCMP).
  - The FlowSpec is opaque to SCMP (i.e., it does not care about the actual contents of the FlowSpec, it just transports it).
  - RFC 1819 defines a FlowSpec to ensure interoperability, but other FlowSpecs can be used.

The Routing Function

- The routing function is not specified by ST-2.
- ST-2 expects an unicast routing capability that can find paths from the source to each of the individual destinations; SCMP messages use these routes.
- The routing function is called in a hop-by-hop basis.
- Once a route is established, it persists for the life of the stream.

The Local Resource Manager

- The Local Resource Manager is not specified by ST-2. It provides the following functions:
  - Local resource management: CPU, code memory, buffer space, network adapters, outbound bandwidth.
  - Stream admission control (are there enough resources?)
  - QoS computation: figure out what the QoS will be.
  - QoS enforcement on flows.
- Optional functions:
  - Data regulation (traffic smoothing)
  - Policing
  - Stream preemption based on priorities.

General Protocol Operation

- Resources for streams are allocated using SCMP messages.
- The source of the stream is responsible for allocating resources.
- SCMP messages:
  - CONNECT and ACCEPT are used to build a stream.
  - DISCONNECT and REFUSE are used to close a stream.
  - CHANGE is used to modify the QoS parameters once the stream is established.
  - JOIN is used by a target to request that it be added to a stream.
  - STATUS is used to inquire about streams.
  - NOTIFY messages inform nodes of significant changes
  - HELLO messages detect unreachable or failed nodes.

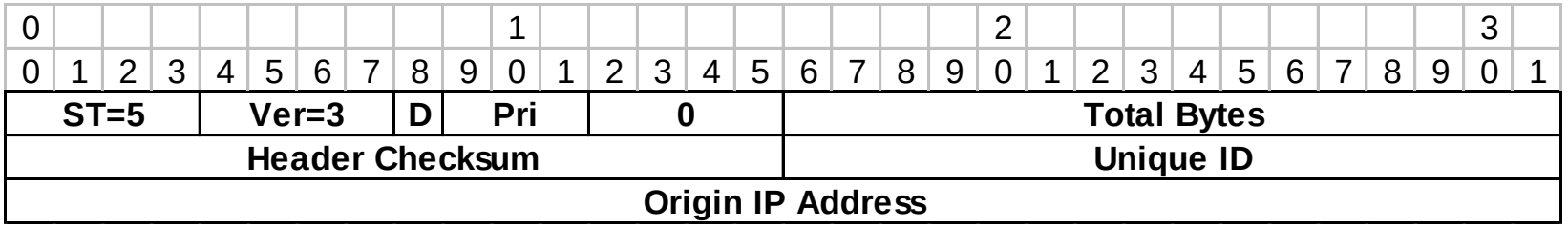
ST-2+ FlowSpec

- QoS Class
- Precedence
- Basic QoS Parameters:
  - Message (Packet) Size
  - Message Rate (messages/second)
  - End-to-End Delay (milliseconds)
- Basic QoS Parameters are specified three times:
  - The ***desired*** values are assigned by the application and never modified in transit.
  - The ***limit*** values are the minimum QoS the application is prepared to accept; they are never modified.
  - The ***actual*** values are changed hop-by-hop, but must be between the desired and the limit.

ST-2 FlowSpec, cont.

- QoS Classes:
  - PREDICTIVE: The negotiated QoS can be violated for short periods of time.
  - GUARANTEED: The negotiated QoS can never be violated and must be met always; support for this class is optional.
- Precedence:
  - Relative importance of the connection being established.
  - Higher precedence streams can take over resources allocated to lower precedence streams.

ST-2 Packet Header



- **ST** is the IP version number (5); **Ver** indicates the version (3 for ST2+).
- **D** differentiates between Data (D=1) and control (D=0, SCMP).
- **Pri** is the packet drop priority; 000 is the lowest, 111 is the highest.
- **Total Bytes** is the total number of bytes in the packet, including the header.
- **Header Checksum** covers only the header, not the data.
- **Unique ID** is locally unique at the source and assigned by it.
- **Origin IP Address** is the IP address of the source

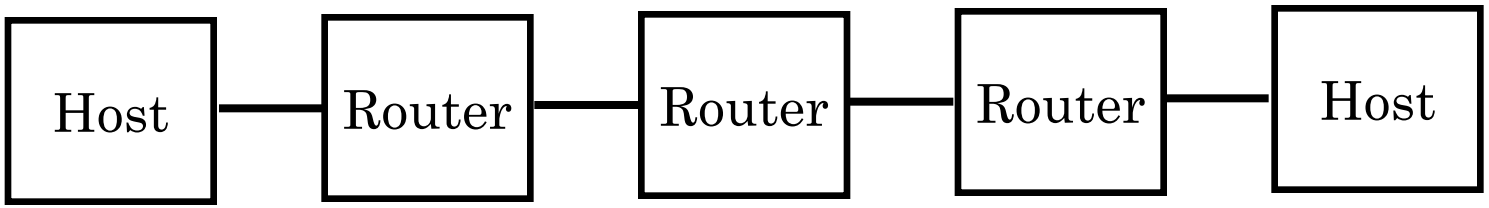
  (each stream is uniquely identified by UniqueID+OriginIPAdddress)

Resource ReSerVation Protocol (RSVP)

RFC 2205 - Version 1 Functional Specification

RSVP

- RSVP is a signaling protocol, used by a host to request a specific QoS from the network for particular data streams or flows.
- RSVP is also used by routers
  - to deliver QoS control requests to all nodes along the paths of the flows and
  - to establish and maintain state to provide the requested service



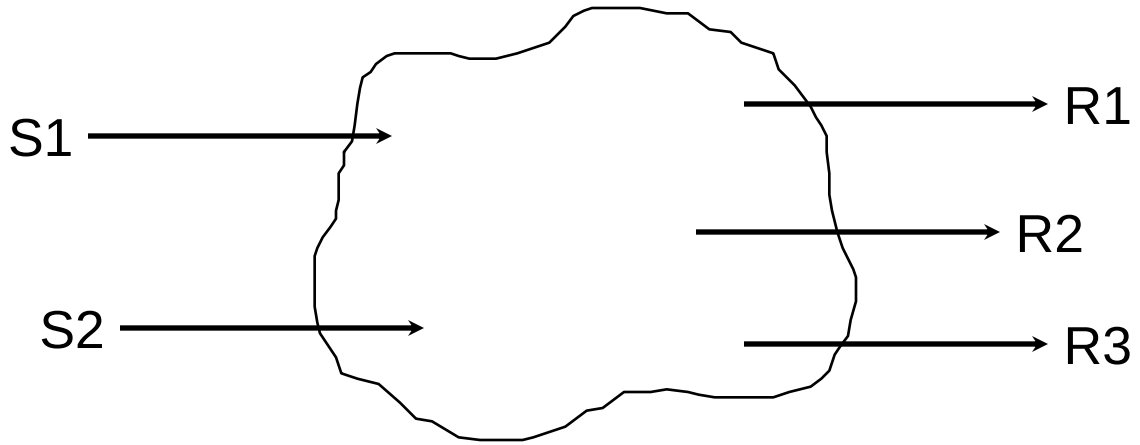
RSVP Requests

- RSVP requests resources for simplex flows (i.e. in only one direction)
- RSVP requests will generally, but not necessarily, result in resources being reserved in each node along the data path.

RSVP Mechanisms

- RSVP protocol mechanisms provide a general facility for creating and maintaining distributed reservation state across a mesh of multicast or unicast delivery paths
- RSVP transfers and "manipulates" QoS control parameters as opaque data, passing them to the appropriate traffic control modules for interpretation
- Structure and contents of the QoS parameters are documented in specifications developed by the integrated Services Working Group

Modes of Communications



RSVP is designed to support all modes of communications

Point to point
Point to multipoint
Multipoint to multipoint
Multipoint to single point

RSVP and IP

- RSVP is a network layer control protocol, like ICMP, IGMP or routing protocols
- RSVP messages are encapsulated into IP datagrams with protocol number 46.
- RSVP is designed to operate with current and future unicast and multicast routing protocols:
  - An RSVP process consults the local routing database to obtain routes

Basic Protocol Operation

- Sources send "Path" messages along with the stream data packets; these messages ***do not*** reserve any resources.
- The "Path" messages mark the routed path between the source and receiver(s), and collect info about the QoS viability of each router along the path.
- If a target wants to receive a stream, it will send a reservation message ("Resv") back towards the source. As this message travels, it reserves resources along the way.
- Routers can "merge" downstream reservations to the same stream.
- State is maintained as long as "Path" and "Resv" messages flow.

RSVP Example

- Example: (Multicast from a single source)
  - Host sends IGMP messages to join a multicast group
  - Source sends RSVP PATH messages to multicast address on a regular basis
  - Routing protocols determine where packets get forwarded
  - Host sends RSVP Reservation message to reserve resources along the delivery path(s) of that group

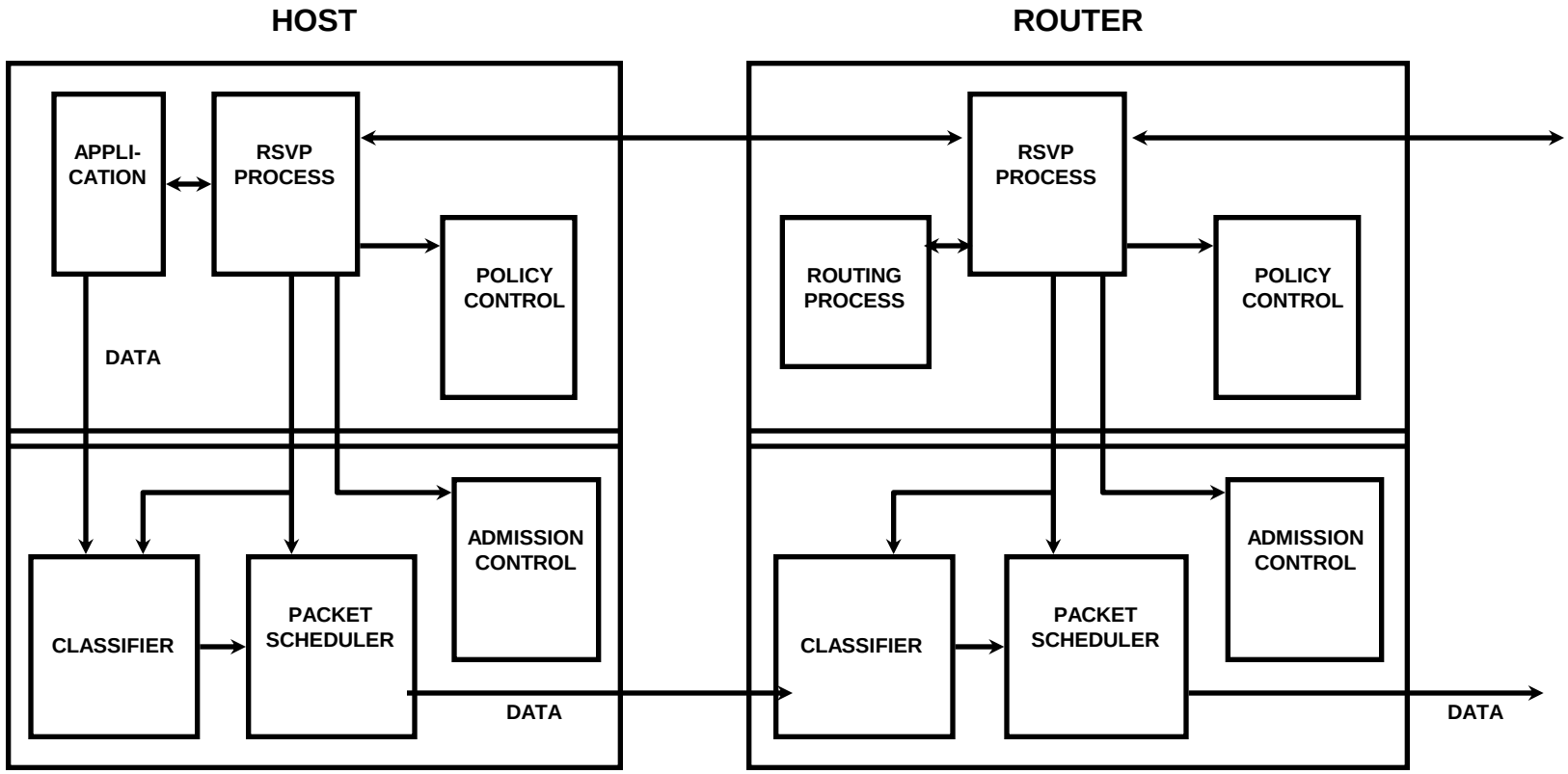
RSVP: Receiver driven

- Receivers are responsible for requesting QoS control
  - efficiently accommodates
    - large groups
    - dynamic group membership
    - heterogeneous receiver requirements

Soft State

- For RSVP to work well with large and dynamic group memberships, router state information is created, modified and removed incrementally, i.e. RSVP uses *soft state*
- RSVP sends periodic refresh messages to maintain the state along the reserved path(s); in absence of refreshes, the state will automatically time-out and be deleted

RSVP modules in Hosts and Routers



RSVP Process Module
Reserving Resources

- A receiver host application makes a QoS control request
- Request is passed to local RSVP process
- RSVP process uses RSVP protocol to carry the request to all nodes (routers and hosts) along the reverse data path(s) to the data source(s)

Packet Processing Modules
Packet classifier and packet scheduler

- Each node capable of QoS control passes data packets through a *packet classifier* which determines for each packet
  - the route
  - QoS class
- On each outgoing interface, a *packet scheduler* makes forwarding decisions for every packet to achieve the promised QoS on the particular link-layer medium used by the interface.

Local Decision Modules
Admission and Policy Control Modules

- At each node, an RSVP QoS control request is passed to two local decision modules:
  - **admission control module**: determines whether the node has sufficient available resources to satisfy the requested QoS
  - **policy control module**: determines whether the user has administrative permission to make the reservation

Local Decision Modules, cont.

- If both checks succeed, parameters are set in the packet classifier and in the scheduler, to obtain the desired QoS
- If either check fails, the RSVP program returns an error notification to the application that originated the request
- Packet scheduler and admission control components implement QoS service models defined by the Integrated Services Working Group

Reservation Request Specifications

- RSVP reservation request consists of:
  - a filter specification
  - a flow specification

Filter Spec

- defines the set of data packets (the flow) to receive the QoS defined by the *flow spec*
- used to set parameters in the *packet classifier*
- allows selection of subsets of the packets in a given session, defined in terms of
  - senders (sender IP address and generalized source port)
  - higher level protocol
  - any fields in any protocol header (e.g. different subflows in a hierarchically encoded signal)
- today's restrictions (sender IP address, optionally, UDP/TCP port number SrcPort)

Flow Spec

- specifies a desired QoS
- used to set parameters in the node's *packet scheduler*
- includes:
  - service class
  - Tspec (traffic descriptor)
  - Rspec (desired QoS parameters)

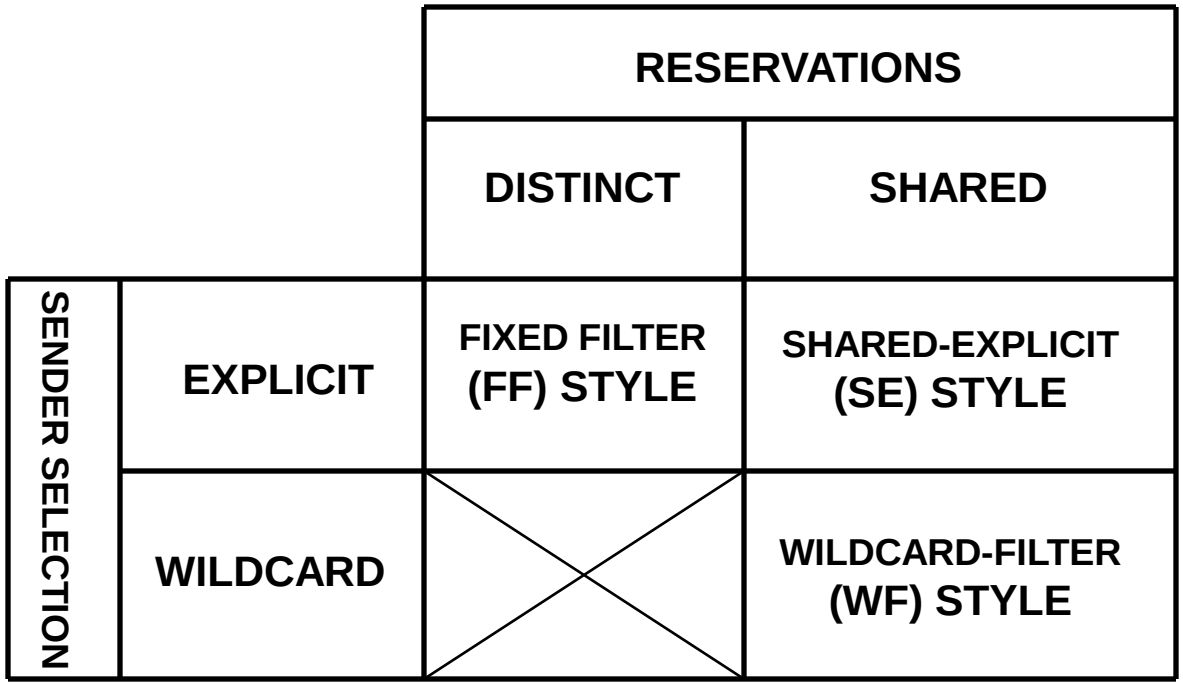
Handling Requests

- Forwarding requests upstream
  - towards appropriate senders (the "scope" of request)
  - merging of requests
- Confirmation request
  - if requested, confirmation is sent back if reservation is successful

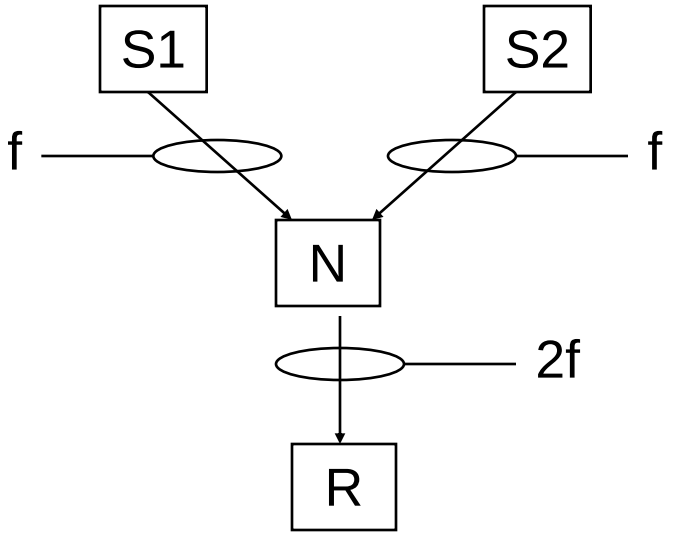
Reservation Styles

- Reservation Option with respect to:
  - reservation to be made
    - distinct for each upstream sender
    - shared among all packets of selected senders
  - selection of senders
    - explicit list of all selected senders (*filter spec*)
    - wildcard (no *filter spec*)

Reservation Styles (cont.)

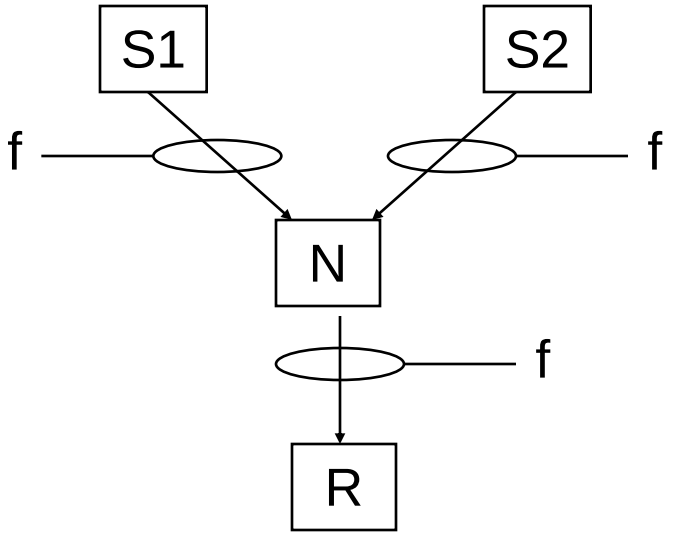


Fixed-Filter Reservation



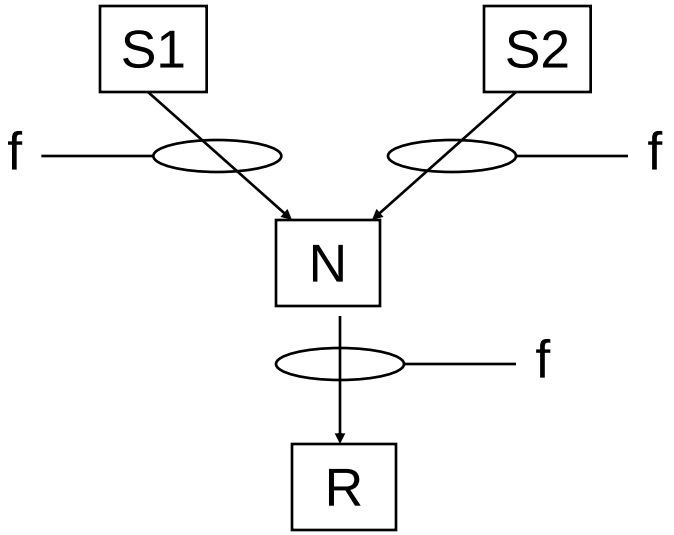
- Receivers make separate reservations for different senders.
- All senders can be active at the same time.

Wildcard-Filter Reservation



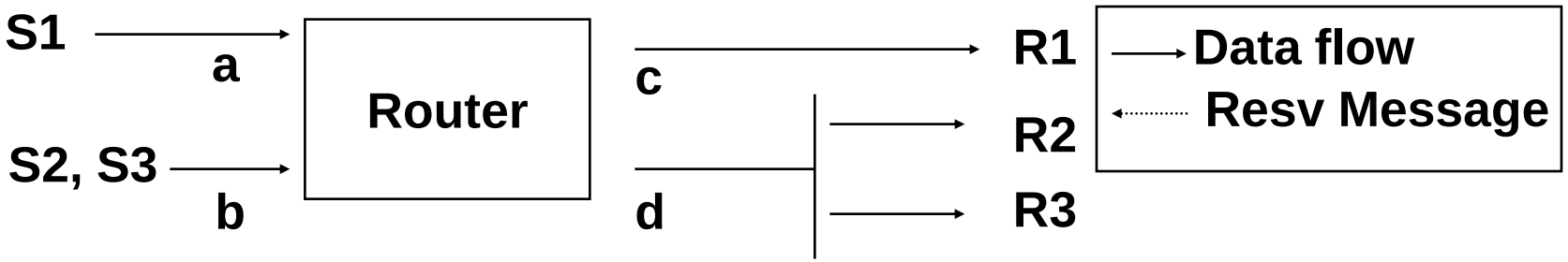
- Receivers make single reservation shared by all senders.
- Only one sender can be active at a time.
- Automatically extends to new senders as they appear.

Shared-Explicit Reservation



- Receivers make single reservation shared by all senders.
- Only one sender can be active at a time.
- The receiver explicitly indicates which senders can use the reservation.

Reservation Styles



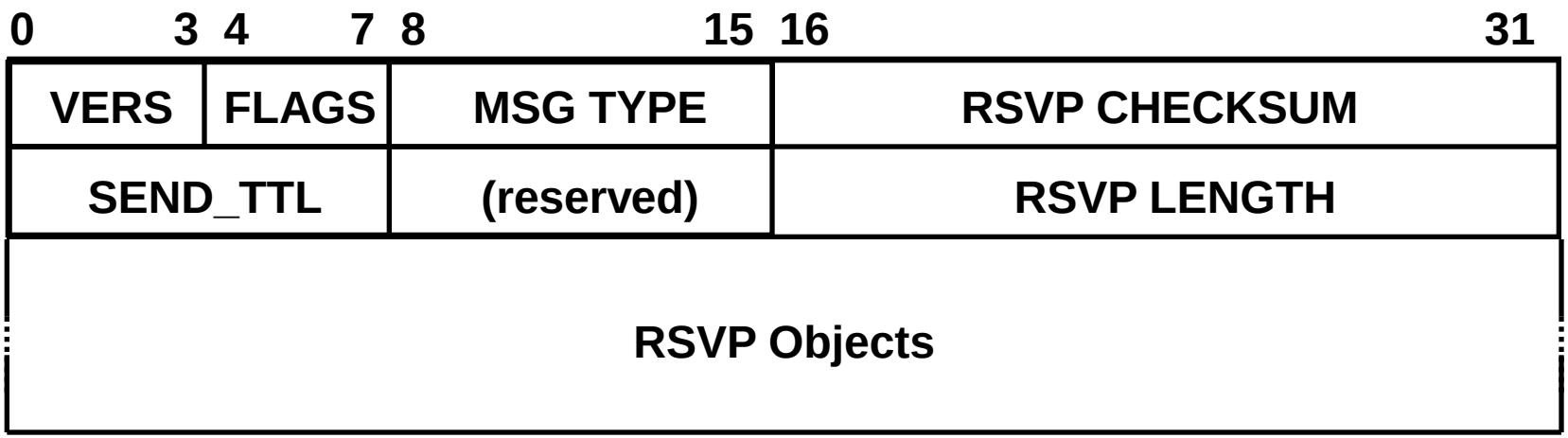
Reserved

WF(*(4B)) ← a	c	*(4B)	c ← WF(*(4B))
WF(*(4B)) ← b	d	*(3B)	d ← WF(*(3B)), WF(*(2B))
FF(S1(4B)) ← a	c	S1(4B) S2(5B)	c ← FF(S1(4B), S2(5B))
FF(S2(5B), S3(B)) ← b	d	S1(3B) S3(B)	d ← FF(S1(3B), S3(B)), FF(S1(B))
SE(S1(3B)) ← a	c	(S1,S2)(B)	c ← SE((S1,S2)(B))
SE((S2,S3)(3B)) ← b	d	(S1,S2, S3)(3B)	d ← SE((S1,S3)(3B)), SE(S2(2B))

RSVP Attributes

- RSVP
  - makes resource reservations for both unicast and multicast traffic
  - adapts to dynamically changing group membership as well as changing traffic
  - is simplex (reservation is for unidirectional flows)
  - is receiver-oriented
  - provides several reservation models or "styles"
  - maintains soft state in routers
  - transports and maintains opaque state information for traffic and policy control
  - is not a routing protocol but depends on routing protocols (present and future)
  - provides transparent operation through routers that do not support it
  - supports IPv4 and IPv6

RSVP Message Format



Header Fields

- Version - always 1
- FLAGS - none defined yet
- MSG TYPE
  - 1 = Path
  - 2 = Resv
  - 3 = PathErr
  - 4 = ResvErr
  - 5 = PathTear
  - 6 = ResvTear
  - 7 = ResvConf

Header Fields (cont.)

- RSVP CHECKSUM
  - computed for the entire message
- SEND_TTL
  - TTL used when the message was sent
- RSVP LENGTH
  - length of the entire RSVP message

Object Classes (1)

- NULL
- Session
  - Used to identify a flow
  - Destination IP address, IP protocol id, destination port
- RSVP_HOP
  - IP address of the RSVP-capable node which sent this message
- TIME_VALUES
  - Value for the refresh period used by the message creator

Object Classes (2)

- Style
  - Reservation style for Resv messages
- Flowspec
  - Defines the desired QoS
- Filter Spec
  - Subset of session data packets which should receive the desired flowspec
- Sender Template
  - Sender's IP address and possibly additional sender identification

Object Classes (3)

- Sender Tspec
  - Traffic characteristics of the data flow
- Adspec
  - OPWA data (measures downstream QoS)
- Error Spec
  - Specifies the error in a PathErr or ResvErr message
  - Specifies the confirmation in a ResvConf message
- Policy Data
  - Used to determine if the reservation is administratively permitted

OPWA=One Pass With Advertising

Object Classes (4)

- Integrity
  - Cryptographic authentication
- Scope
  - Explicit list of sender hosts
  - Used to prevent loops with wildcard sender selections
- Resv Confirm
  - Indicates a confirmation is requested
  - IP address of the receiver who requested confirmation