Integrated Services/RSVP

Anjali Agarwal

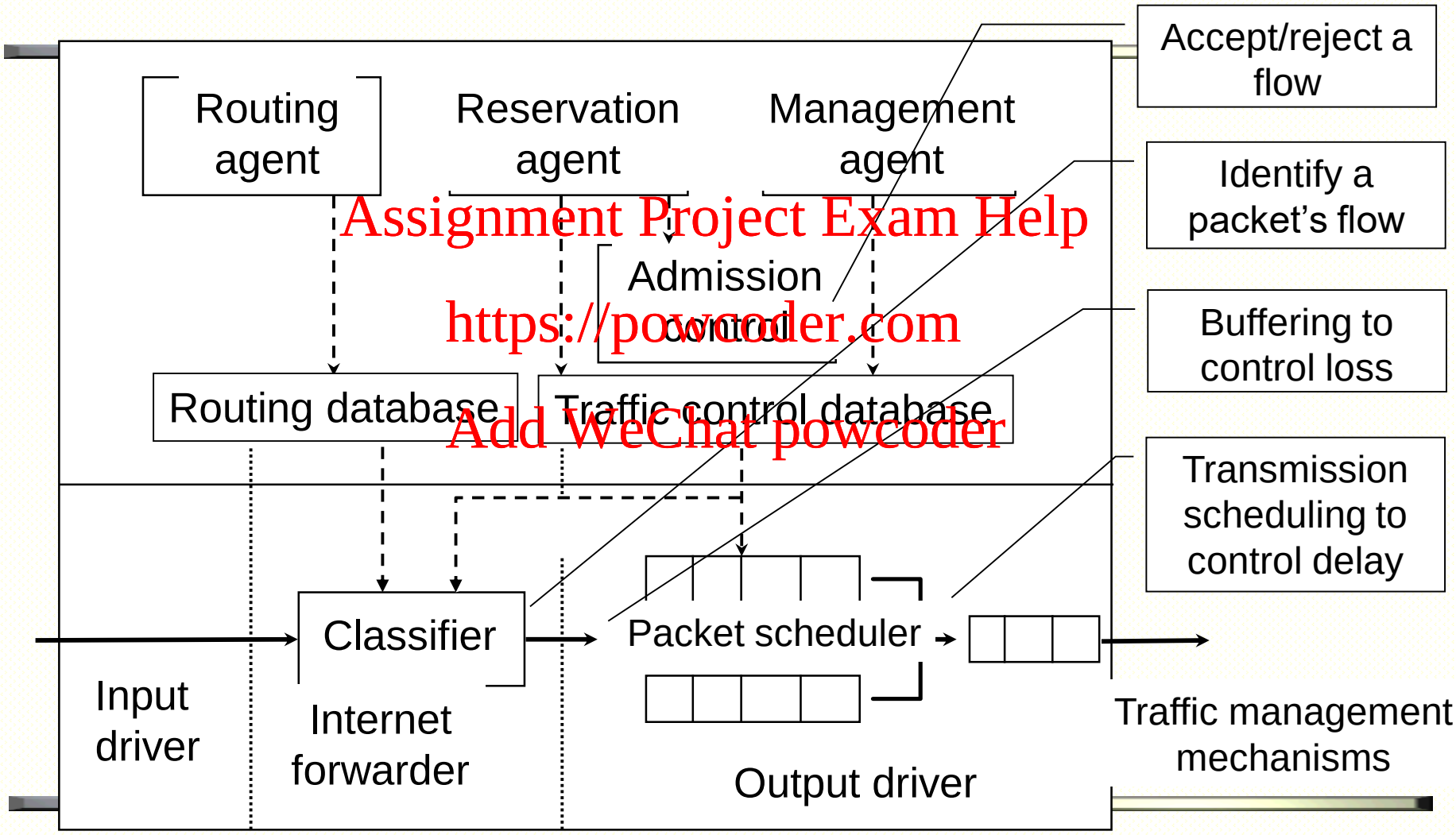
Integrated Services IP Model

- Defines a *flow* as a stream of IP packets
  - » *Generated by a sender and destined to a destination*
  - » *That requires the same QoS*
- Provides QoS to *individual flows* in the Internet
  - » *"Better than Best Effort" for some applications*
  - » *Support for real-time voice and video applications*
- Requires traffic management mechanisms to deliver appropriate QoS to each flow
  - » *Packet classification, scheduling, admission control*
- Explicit reservation of buffers and bandwidth resources for individual flows at every node
  - » Resource Reservation Protocol (RSVP) *provides means for making reservations*

Network Service Models

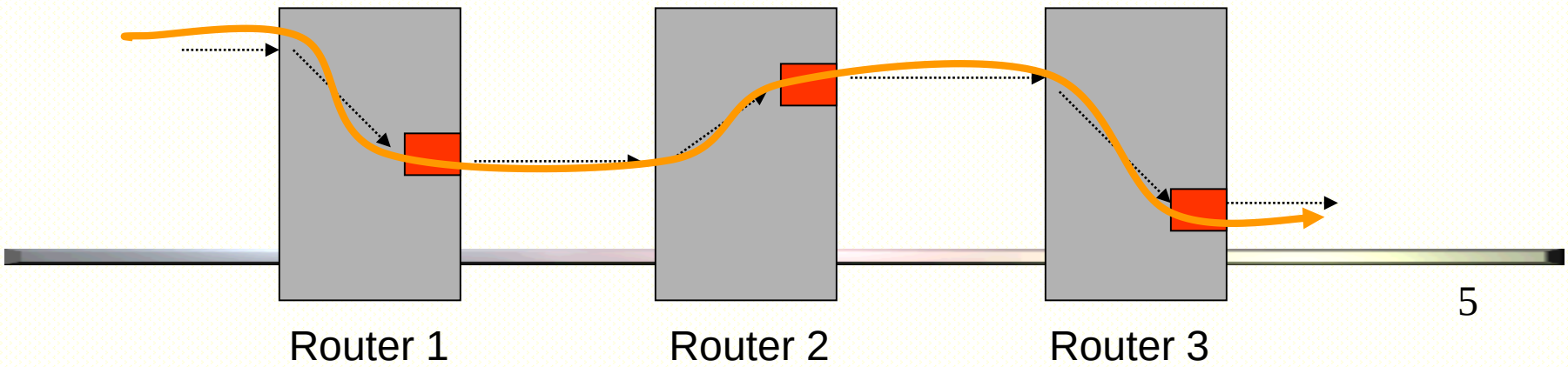
- Best effort service
  - » *No guarantees; suitable for elastic traffic*
  - » *At low loading, suitable for many traffic classes*
- Guaranteed service
  - » *bound on maximum delay*
  - » *guarantee on available bandwidth*
- Controlled load service
  - » *delay consistent with lightly loaded network*

IntServ Router Model



End-to-End Performance

- End-to-end performance for an individual flow is the result of per-switch performances
  - » *delay, jitter, loss, bandwidth*
- Per-switch performance depends on:
  - » *per-packet processing common to all packets*
  - » *specific per-connection or per-class treatment*
- Resources must be allocated by RSVP at each node *for each flow*



Admission Control

- Individual flow negotiates admission into the network
- *Flow Descriptor* has two parts

1. *Filter specification (filterspec)* provides information required by classifier to identify the packets in the flow
2. *Flow specification (flowspec)* describes traffic properties of flow and QoS requirements
   - » Traffic Specification (Tspec) *describes traffic in terms of a token bucket*
   - » Request Specification (Rspec) *describes QoS in terms of bandwidth, delay, loss*

- Each node along path must decide whether a flow can be accepted

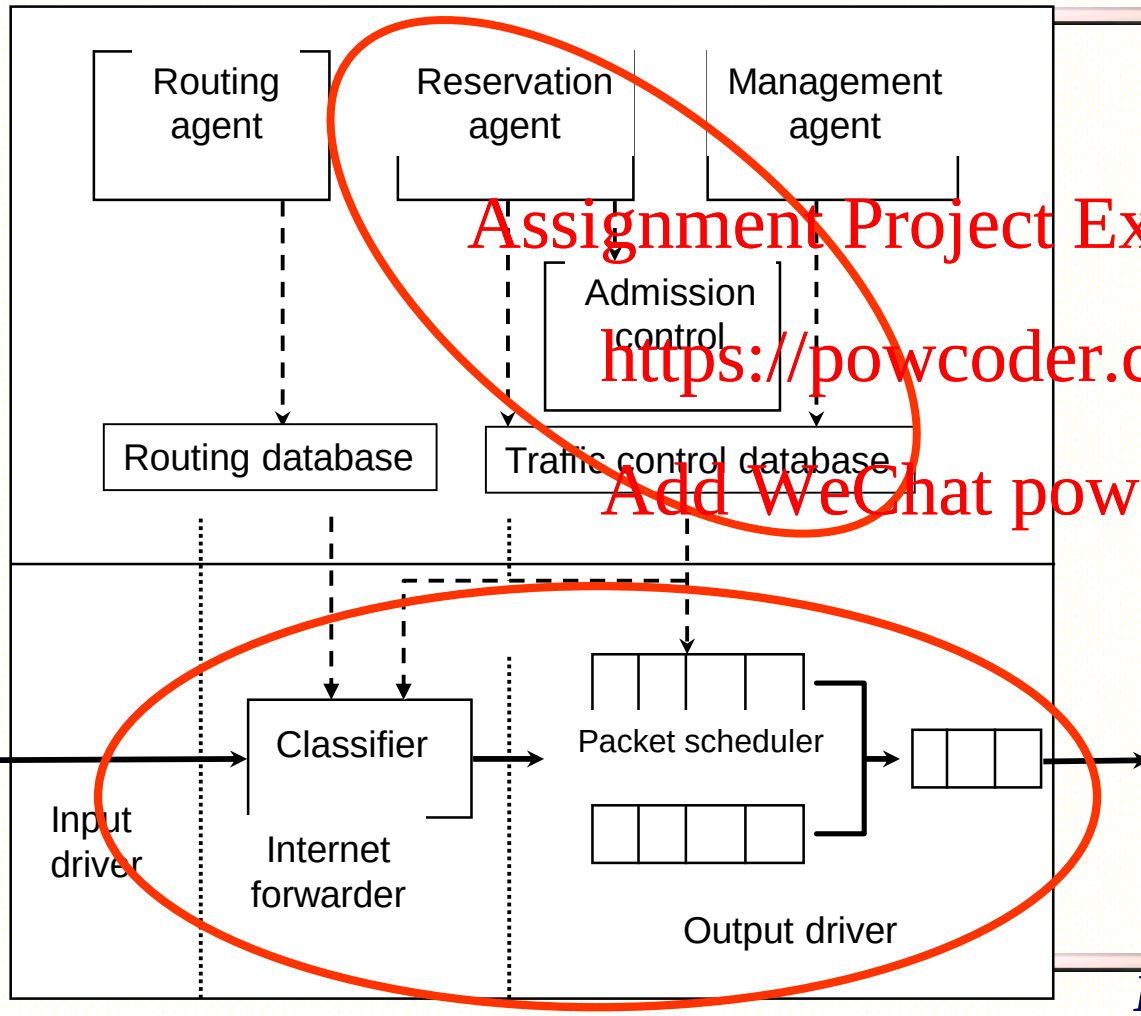
Guaranteed Service

- Intended for flows that require real-time packet delivery
- Provides a firm delay bound
  - » *Each flow is shaped by (b,r) leaky bucket*
    - – *b* token bucket size
    - – *r* token rate
  - » *Police the flow to ensure compliance*
  - » *Reserve bit rate* R>r *at every node (weighted fair queueing)*
  - » *Account for other network parameters*

Controlled Load Service

- Intended for flows that can tolerate some delay but are sensitive to traffic overload
  - » *Equivalent to "Best Effort under Light Traffic"*
  - » *Low delay and low loss, but no quantitative guarantees*
- Less complex than guaranteed service
  - » *Each flow is shaped by (r,b) leaky bucket*
  - » *Use admission control to limit volume of controlled load service*
  - » *Reserve bit rate for the entire class to ensure light traffic mode*
  - » *Police each flow to ensure compliance; Non-conforming packets accorded best effort service*

IntServ involves High Complexity



- Number of (application) flows can become extremely large
- Per-flow treatment involves high complexity
- Traffic Management
  - » *Per-flow classifier*
  - » *Per-flow queueing*
  - » *Per-flow scheduling*
  - » *Huge table sizes & high hardware complexity*
- Admission Control
  - » *Set up & maintenance of individual flows*
  - » *High processing load*

IntServ is not scalable

Why do we need RSVP

- In connectionless protocols, the network nodes (the routers) don't have any knowledge of a "flow" of information; they see only individual datagrams
- There is no mechanism in protocols like TCP/IP to specify just what quality of service (QoS) a given "flow" would require, even if one could figure out what datagrams made it up.

What is RSVP

- a network-control protocol that enables Internet applications to obtain special QoSs for their data flows
- an internetwork end-to-end protocol that reserves the necessary resources for the different classes of service by using the techniques available on each underlying network type
- not a routing protocol; works in conjunction with them to determine where it should carry reservation requests
- Routing protocols determine where packets get forwarded; RSVP is only concerned with the QoS of those packets that are forwarded in accordance with routing.
- It occupies the place of a transport protocol

Features of RSVP

- Makes resource reservations for both unicast and many-to-many multicast applications
- maintains "soft" state in routers and hosts, providing
- Adapts dynamically to changing group membership as well as to changing routes - establishes "soft" state that are built and destroyed incrementally in routers and hosts
- Because multimedia flows may be (in fact, will probably be) asymmetrical, RSVP treats data flow as one directional (simplex). It logically distinguishes the role of data sender from data receiver.
- RSVP is a receiver-based protocol; resource reservations requests are originated by the receivers of the service

Key Concepts

- Data Flows
  - » *is a sequence of messages that have the same source, destination (one or more), and QoS*
  - » *Flow specification defines the desired QoS*
  - » *Session is a set of data flows with the same unicast or multicast destinations (session may have some number of senders talking to some number of receivers)*
  - » *multicast traffic – copy of each data packet forwarded from a single sender to multiple destinations*
  - » *unicast traffic – session involving a single receiver host distinguished by its generalized destination port*
  - » *multiple senders supported for a unicast destination*

Key Concepts (cont.)

- Reservation Model
  - » *basic element is the "flow descriptor" which combines*
    - – filter spec (way to identify datagrams in a flow) with the flow spec (QoS the flow is to receive)
  - » *the filter spec is used to set parameters in the packet classifier*
  - » *flowspec is used to set parameters in the node's packet scheduler*
- Reservation Styles to fit a variety of applications

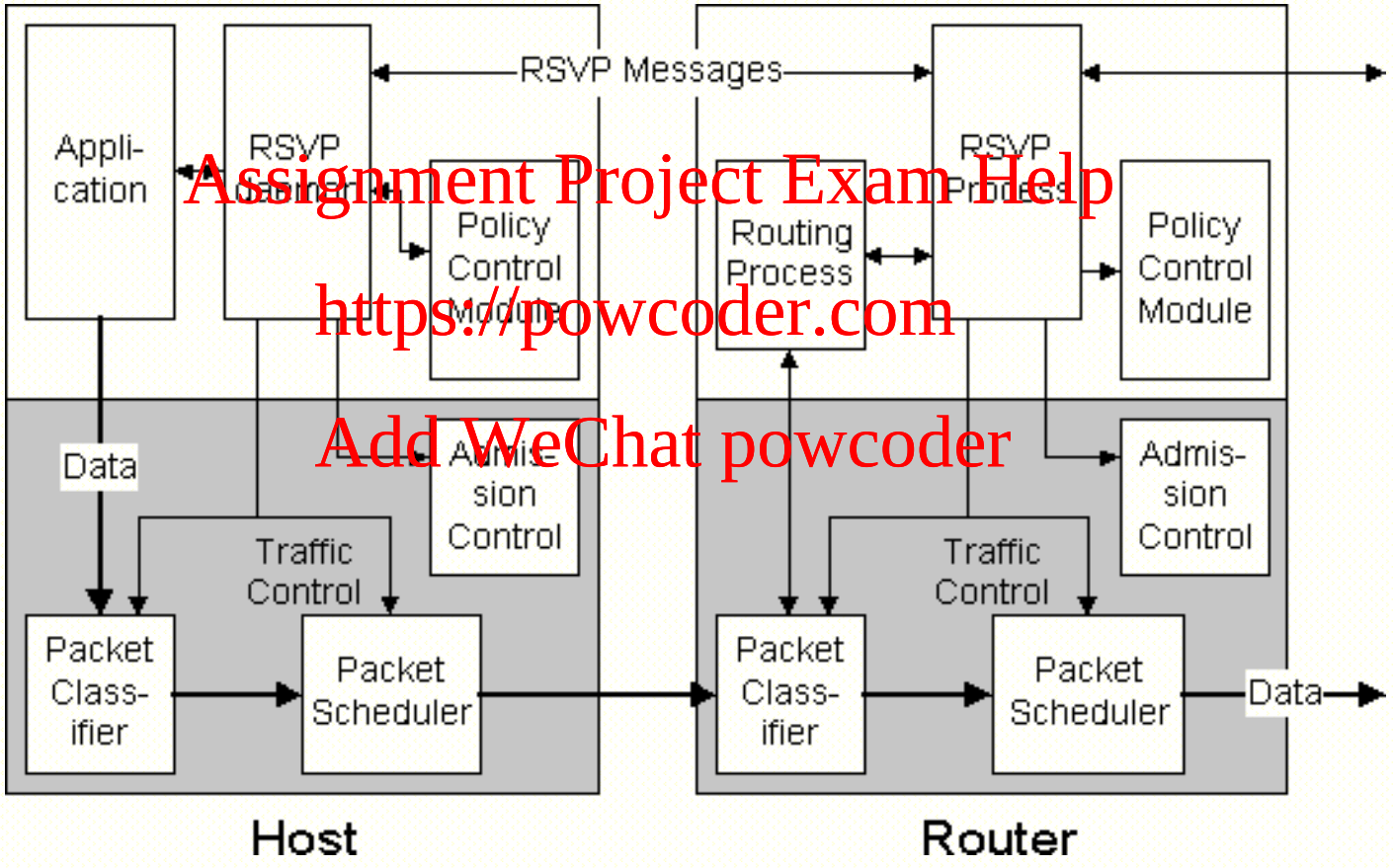
Traffic Control modules

- RSVP Deamon - the "main" module,
  - » *manages the reservation algorithm with the help of the other modules of the service*
  - » *responsible to ask the "policy control" and "admission control" for their permission to set up the reservation*
  - » *sets parameters in a "packet classifier" and "packet scheduler" to obtain the desired QoS*
- Policy Control – decision module (for administrative purposes)
  - » *responsible for who is allowed to make reservations, and what kinds of QoS he may reserve*

Traffic Control modules (cont.)

- Admission Control – decision module
  - » *determines whether the node has sufficient resources available to meet the needs of the reservation request*
- Packet Classifier –
  - » *If both admission control and policy control returned a positive decision, then RSVP deamon passes incoming data packets to a packet classifier that determines the route and the QoS class for each packet*
  - » *may be combined with the routing function*
- Packet Scheduler –
  - » *responsible for achieving the promised QoS by prioritizing queues of flows, as necessary*
  - » *key component of the architecture*
  - » *must support the distinction between different services on all nodes*

Traffic Control modules (cont.)



Reservation Styles

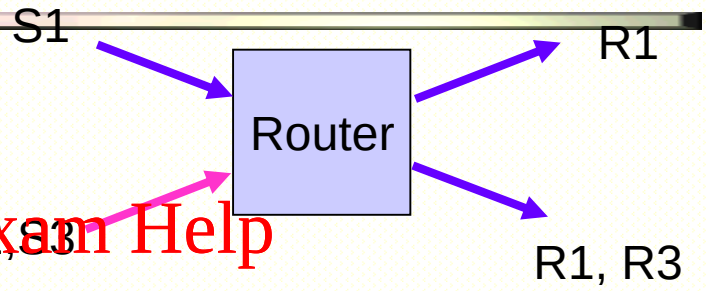
to fit a variety of applications

- fixed filter (FF) – distinct reservation
  - » *reservation is made for packets sent by exactly one sender that is specified in the filterspec*
- shared explicit (SE) – shared reservation
  - » *packets from several sources listed explicitly in the filterspec can use the reservation*
- wildcard filter (WF) – shared reservation
  - » *all sources sending to the multicast group address share the reservation*

Shared and wildcard filters are useful for applications that are self-limiting in their bandwidth needs. These include audio sessions, because usually not more than one or two participants talk at the same time.

FF is more appropriate for video signals

Reservation Styles



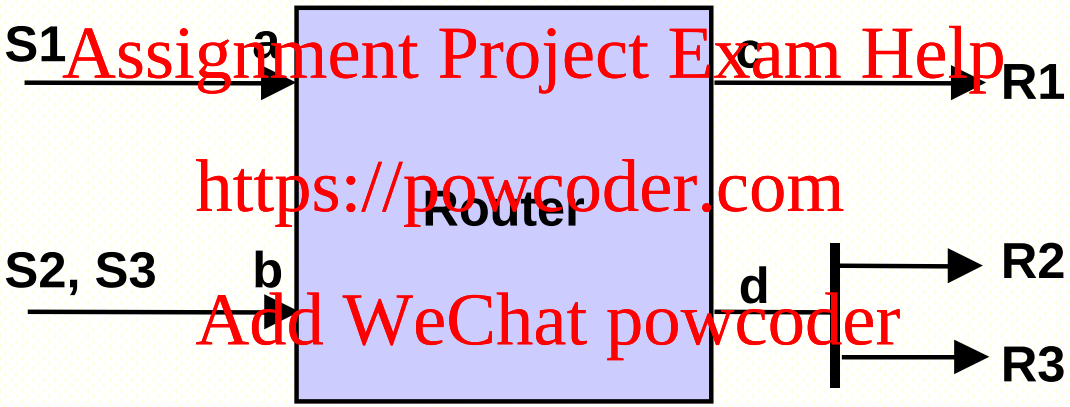
- S1, S2, S3, R1, R2, R3 belong to the same session
- Can S2 & S3 share the bandwidth reserved by S1?
  - » *Yes if application has one sender transmit at a time*
  - » *No if multiple senders transmit*
- How does router know which senders can access a reserved resource?
  - » *Explicit List*
  - » *Wildcard (Any sender in session)*

Fixed Filter
- Separate reservations
- Explicit list

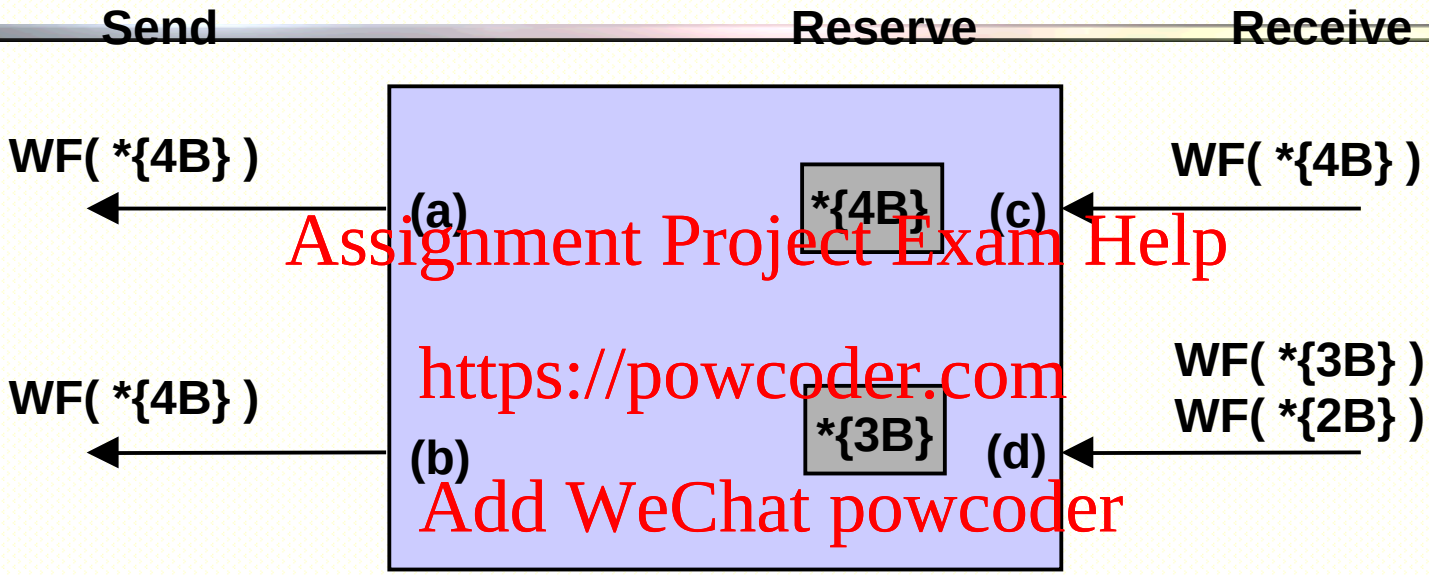
Wildcard Filter
- Shared reservations
- Wildcard (all senders)

Shared Explicit Filter
- Shared reservations
- Explicit list

Example

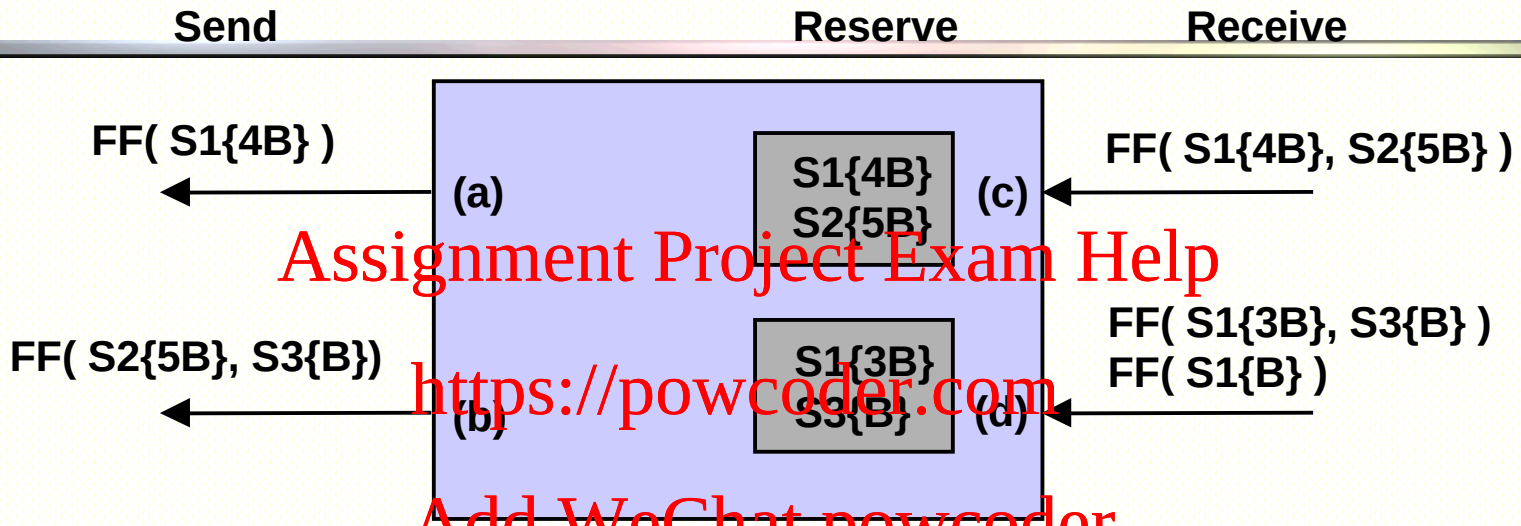


Wildcard Filter



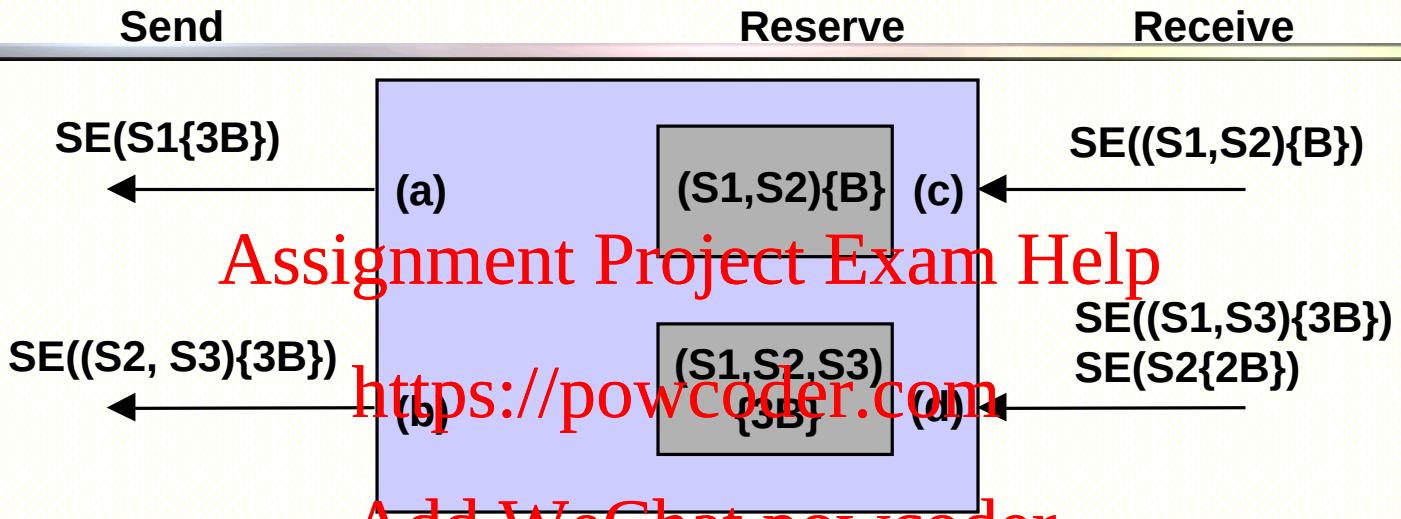
- Wildcard request for 4B from R1
- Wildcard request for 3B & 2B from R2 and R3;
- Merged into 3B request
- Inputs merge requests to 4B before upstream
- Example: audioconferencing with different bitrates

Fixed Filter



- FF request from R1 for 4B from S1, 5B from S2
- FF request from R2 for 3B from S1, B from S3
- FF request from R3 for B from S1
- Merge request to S1 for 3B
- Merge request to S1 for 4B
- Example: all-to-all videoconference

Shared Explicit

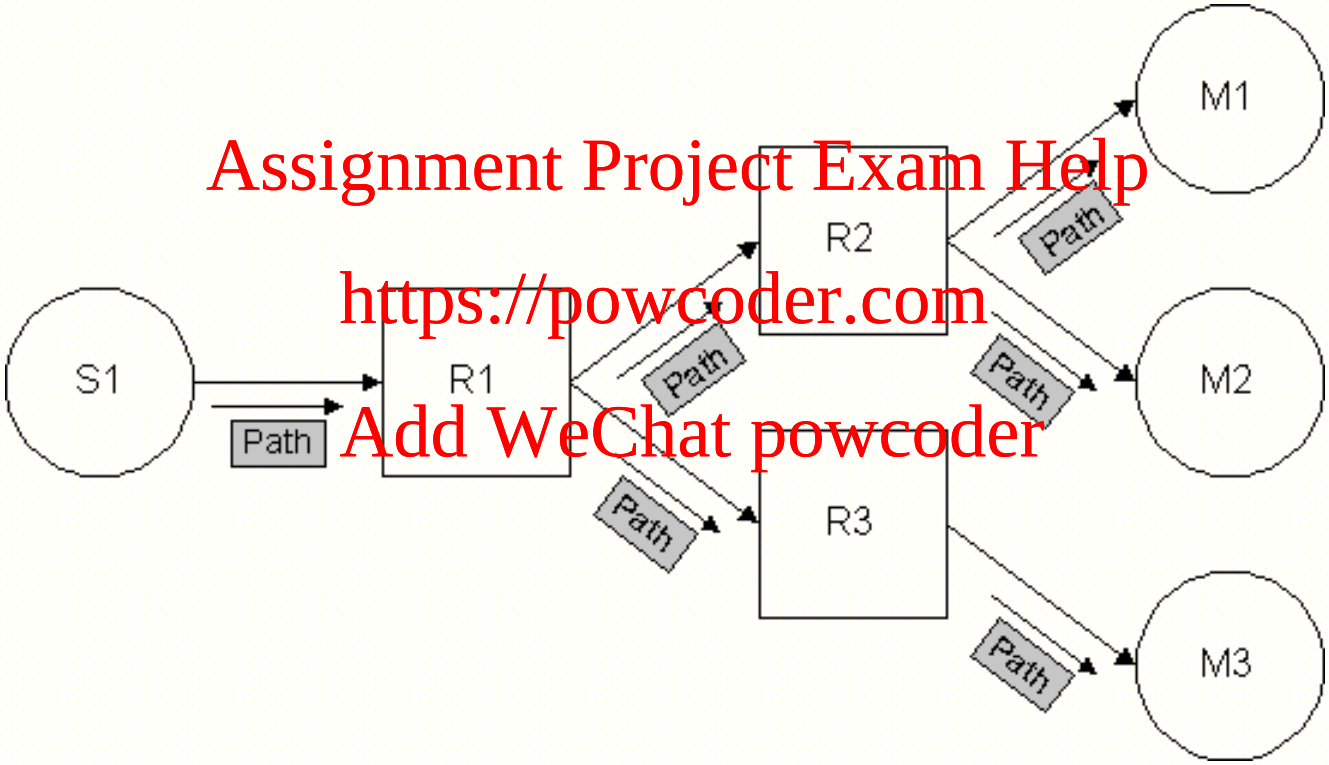


- SE request for B for S1 & S2 from R1
- SE request for 3B for S1 & S3 from R2
- SE request for 2B for S2 from R2
- Merge to union of list (S1, S2, S3) & max request, 3B
- Example: layered video

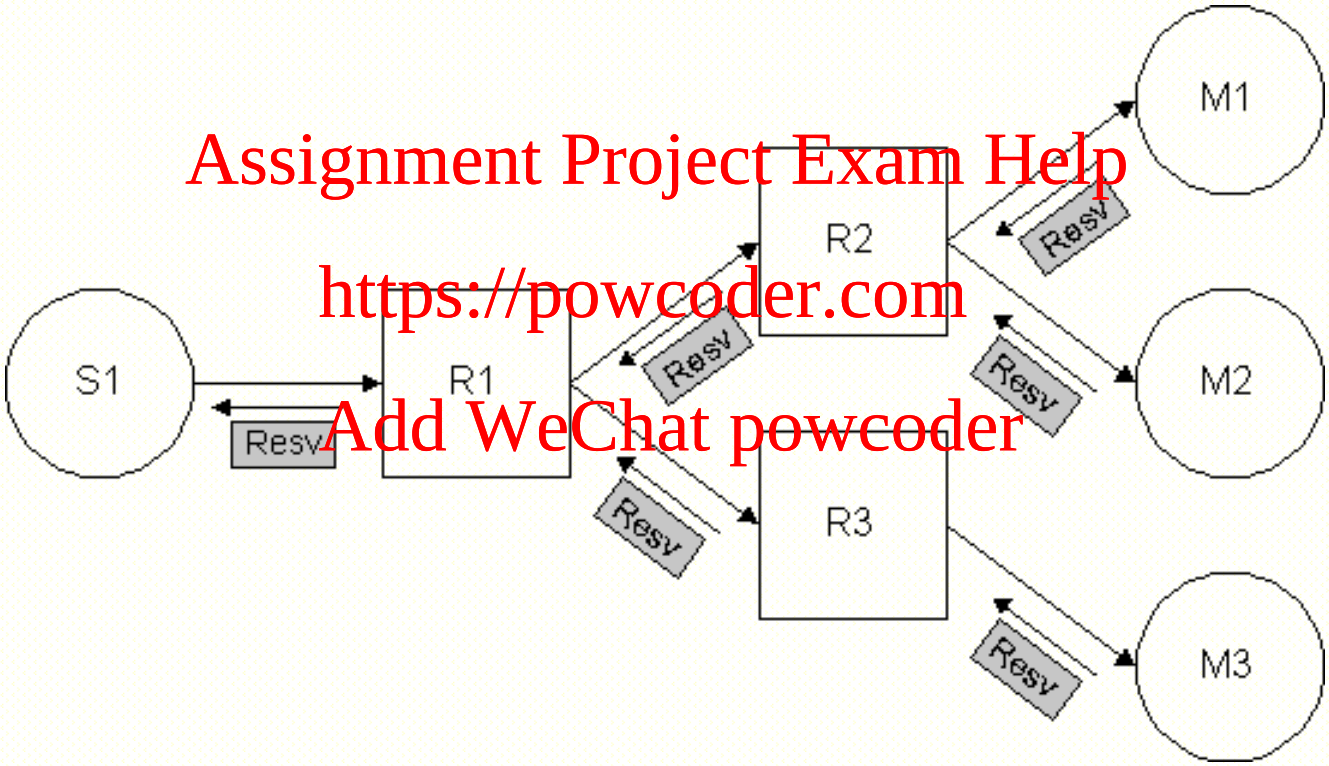
RSVP Messages

- Reservation-Request Messages
  - » *sent by each receiver host towards the senders*
  - » *follows in reverse the routes that data packets use*
- Path Messages
  - » *sent by each sender forward along the unicast or multicast routes provided by routing protocols*
  - » *used to store the path state in each node*
  - » *the path state is used to route reservation request messages in reverse direction*

Path Messages



Resv Messages



RSVP Messages (cont.)

- Error Messages
  - » *path-error messages:*
    - – result from path messages and travel toward senders
    - – routed hop-by-hop using the path state
  - » *reservation-request error messages:*
    - – result from reservation-request messages and travel toward the receiver
    - – routed hop-by-hop using the reservation state
  - » *Information in error messages include:*
    - – admission failure
    - – Bandwidth unavailable
    - – service not supported
    - – ambiguous path

RSVP Messages (cont.)

- Confirmation Messages
  - » *reservation-request acknowledge messages sent as a result of the appearance of a reservation confirmation object in a reservation-request message*
- Teardown Messages
  - » *removes the path and reservation state without waiting for the cleanup timeout period*
  - » *can be initiated by an application in an end system or a router as the result of state timeout.*
  - » *path-teardown messages – deletes the path state which in turn deletes the reservation state, routed like path messages*
  - » *reservation-request teardown messages – deleted reservation state, routed like reservation request messages*

Reservation Process

- The net's routing table defines the routes from senders to receivers

  - » *distribution tree is the directed tree carrying traffic away from a source to all destinations, with the sender being the root node, receivers being the leaves and routers being intermediate nodes*
  - » *trees may be unicast or multicast*

Reservation Process (cont.)

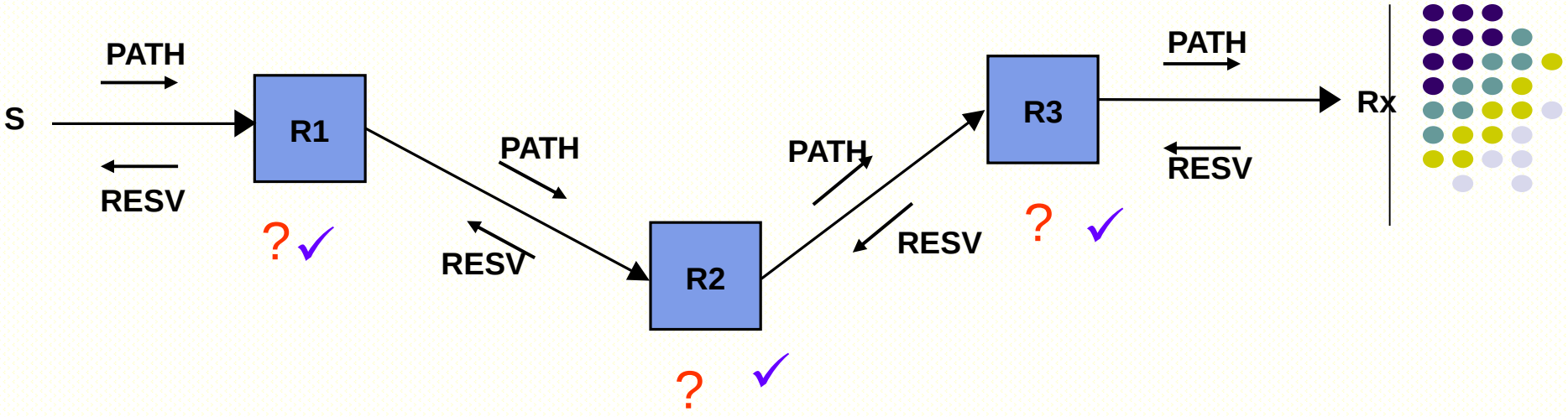
- RSVP senders periodically emit PATH message which indicates that the system is a sender and contains information required by network to route later on RESV messages up the distribution tree.

  - » *PATH messages include the following information*
    - – destination address (IP multicast address)
    - – Reservation ID
    - – Previous-hop IP address (used in forwarding RESV msgs)
    - – Templates for identifying traffic from that sender
    - – Flow specification describing the sender's output
  - » *Routers capture this message and update the previous-hop IP address before sending it on*
  - » *Path messages are used to "mark" the network*

Reservation Process (cont.)

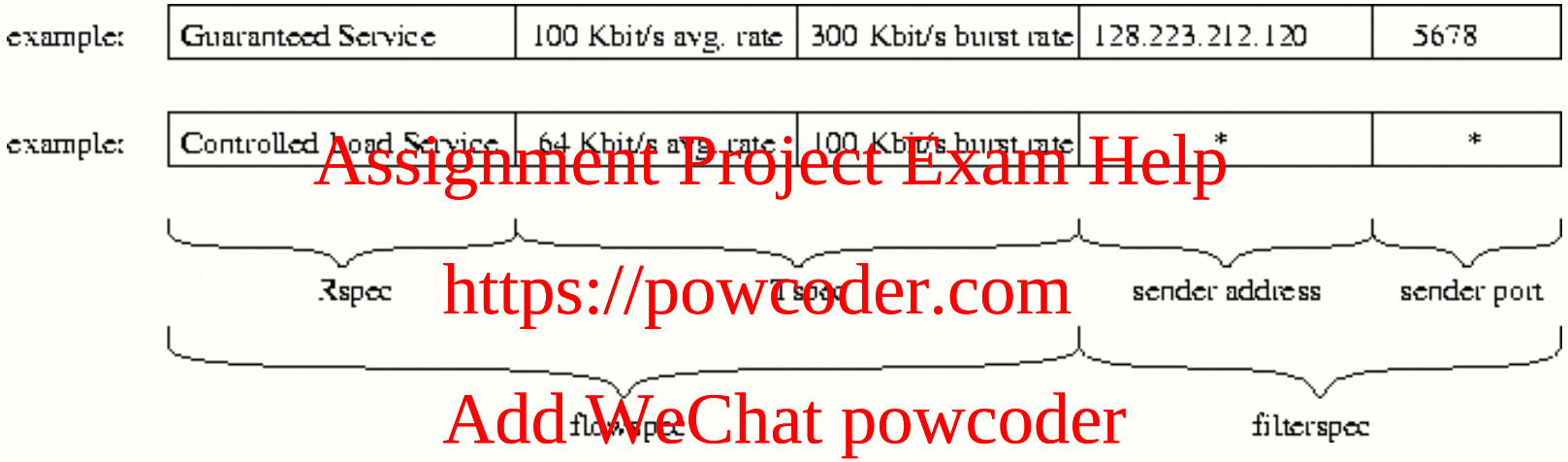
- Receivers can now ask for reservation for a session from the sender by sending RESV messages up the tree

  - *» one source may initiate several sessions (or flows).*
    - – RESV must include the session (flow) that the receiver is referring
  - *» Upon a failure, the requester would be notified that his request has been rejected.*
  - *» Where multicast flows converge, the reservations are combined, using the highest QoS specified*



- Sender multicasts PATH message that describes traffic flow
- Uses an existing routing protocol
- Each router stores address of previous RSVP router (PHOP) and inserts its address in last hop field and forwards message, establishing the path in the reverse direction
- Receiver unicasts RESV message to reserve resources (Can request confirmation from sender)
- Each router performs admission & policy control (Send PathErr message if rejected)
- Reservations may be modified or merged as RESV proceeds back to sender

Reservation Process (cont.)

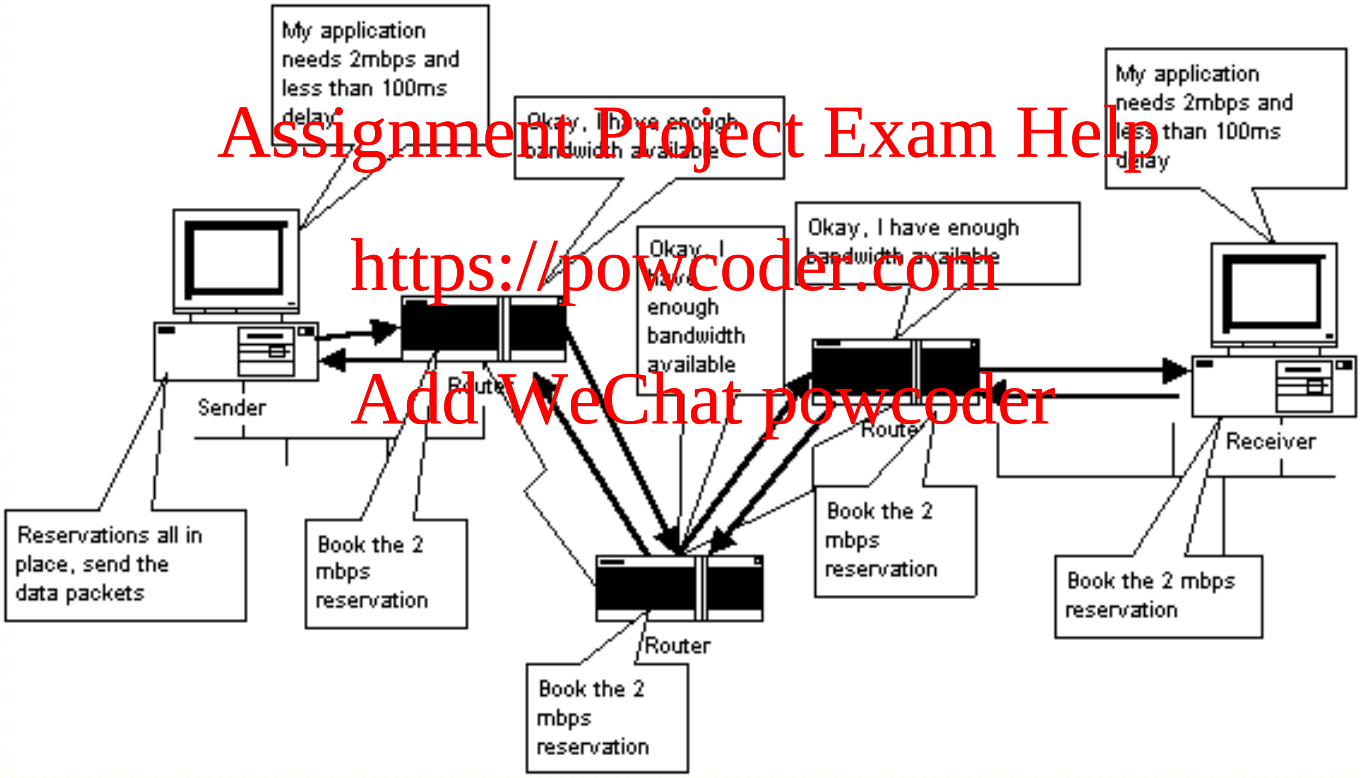


Rspec: specifies the requested service

Tspec: specifies size of the expected data flow

Filterspec: specifies which packets can use the reservation

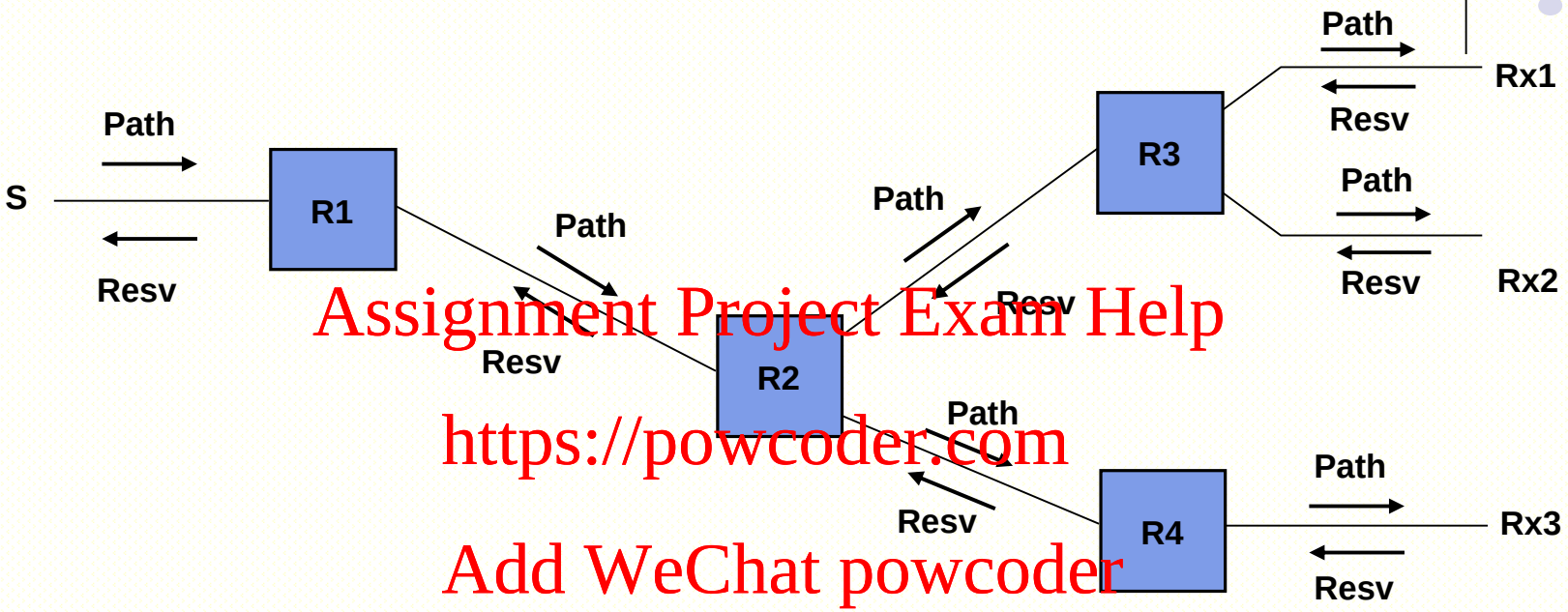
Resource Reservation example



Merging

- In order to increase the efficiency of distribution and to avoid redundancy, different RSVP control messages that reach a certain node might be merged or split as necessary before being forwarded to the next hop(s) so that only the larger reservation is passed on towards the sender
- The style of reservations combined with the requested flowspecs and with the filterspecs - determine the node's decision as to merging, splitting or simply forwarding the control messages

Reservation Merging

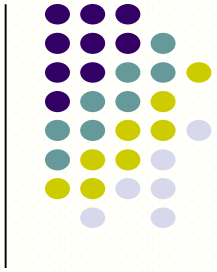


- Resources are shared among receivers up to point where paths to different receivers diverge
- RSVP process at nodes will *merge* requests at node where sufficient resources are already reserved
- Request is not forwarded beyond merge point

Soft state

- Soft state is state information that needs to be refreshed periodically, otherwise it times out – to allow new nodes to be added and existing nodes to be deleted
- Path and resv messages are periodically sent at a configurable refresh interval
- takes care of any route change occurrence
- If end-systems are unable to torn the reservations that are no longer needed, path and reservation state will time out eventually
- Short refresh periods increase control traffic overhead; Long refresh and time-out periods lead to unused capacity that is reserved but no longer needed.

RSVP Soft State

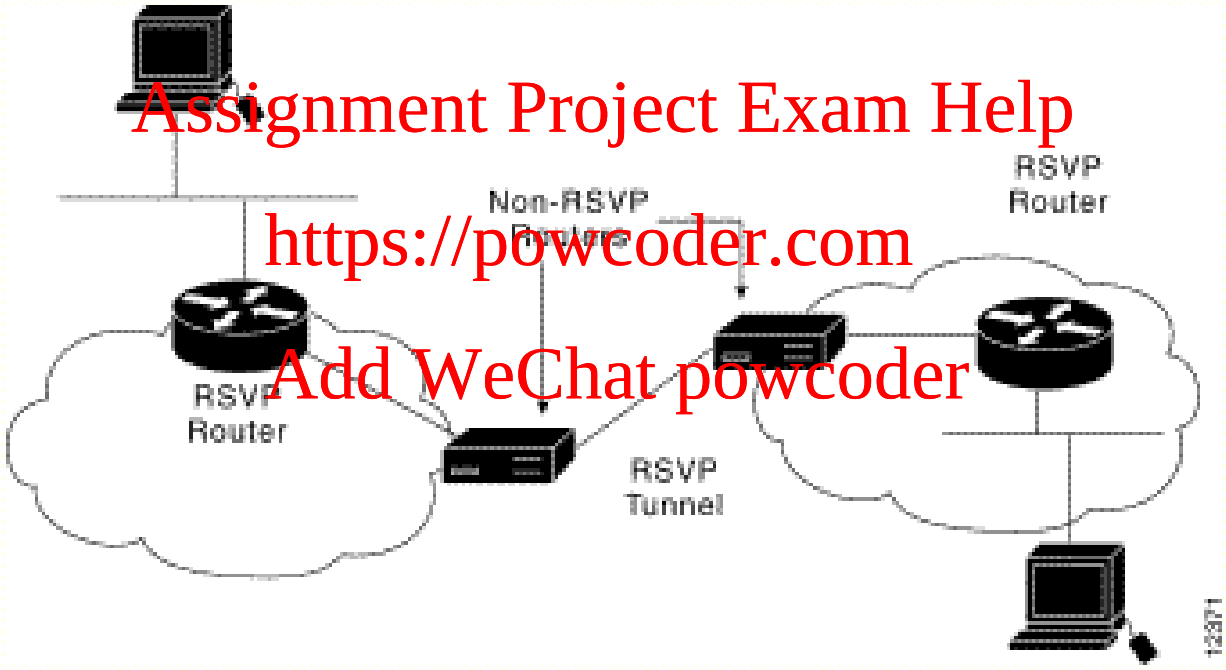


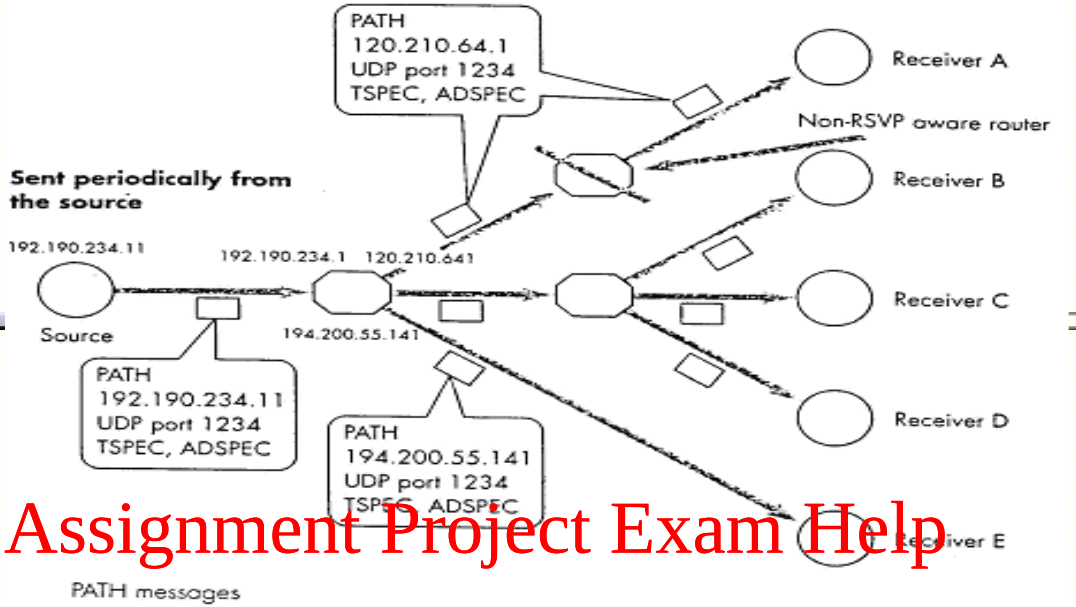
- Reservations are valid for a timeout period
- Need to “refresh” reservation state by resending PATH & RESV messages before expiry time
- Reservation removed if not refreshed by timeout
- RSVP runs directly over IP with type=46
  - message delivery is not reliable
  - Assume 1 in 3 consecutive messages gets through
- Nominal refresh rate specified by *R* (usually 30 sec)
- Refresh period for a receiver randomized from (0.5*R*, 1.5*R*) to avoid simultaneous refresh attempts
- PathTear & ResvTear messages explicitly delete reservations

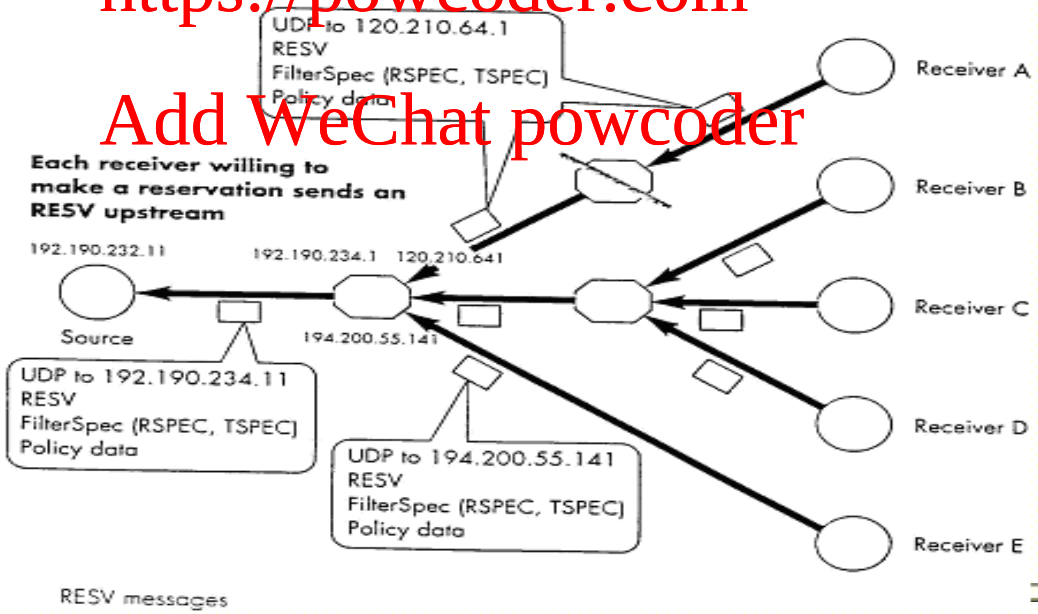
Transparent non-RSVP clouds

- To allow incremental RSVP deployment in the Internet
- No explicit tunneling is necessary, the path messages carry the IP address of the last RSVP-capable router
- Non-RSVP routers route path messages towards the receivers like any other unicast or multicast packet
- Path state is set up only in RSVP capable routers
- Resv messages are sent as unicast packets from one RSVP-capable node to the next RSVP-hop upstream
- However, the end-to-end QoS is unpredictable since there is no traffic control in non-RSVP nodes

Transparent non-RSVP clouds



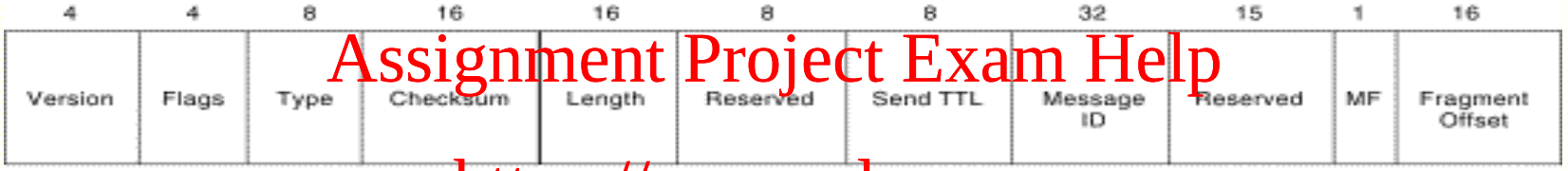




RSVP Packet Format

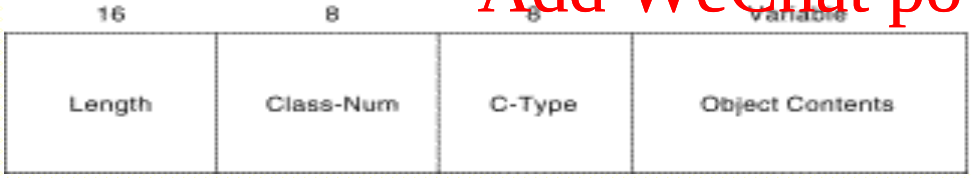
RSVP Message Header Fields

Field Length, in Bits



RSVP Object Fields

Field Length, in Bits



Value	Message Type
1	Path
2	Reservation-request
3	Path-error
4	Reservation-request error
5	Path-teardown
6	Reservation-teardown
7	Reservation-request ack

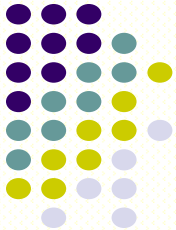
RSVP Message Header

- Length: of this RSVP packet in bytes including the common header and the variable-length objects that follow.
- Send TTL: IP time-to-live (TTL) value with which the message was sent
- Message ID: 32-bit field providing a label shared by all fragments of one message from a given next/previous RSVP hop
- More Fragments (MF) Flag: MF is set on for all but the last fragment of a message
- Fragment Offset---24-bit field representing the byte offset of the fragment in the message

RSVP Object Fields

- Length: 16-bit field containing the total object length in bytes (must always be a multiple of 4 and be at least 4)
- Class-Num: Identifies the object class. Each object class has a name.
- C-Type: Object type, unique within Class-Num
  - » Class-Num and C-Type fields can be used together as a 16-bit number to define a unique type for each object
- Object Contents: The Length, Class-Num, and C-Type fields specify the form of the object content.

RSVP Message Objects

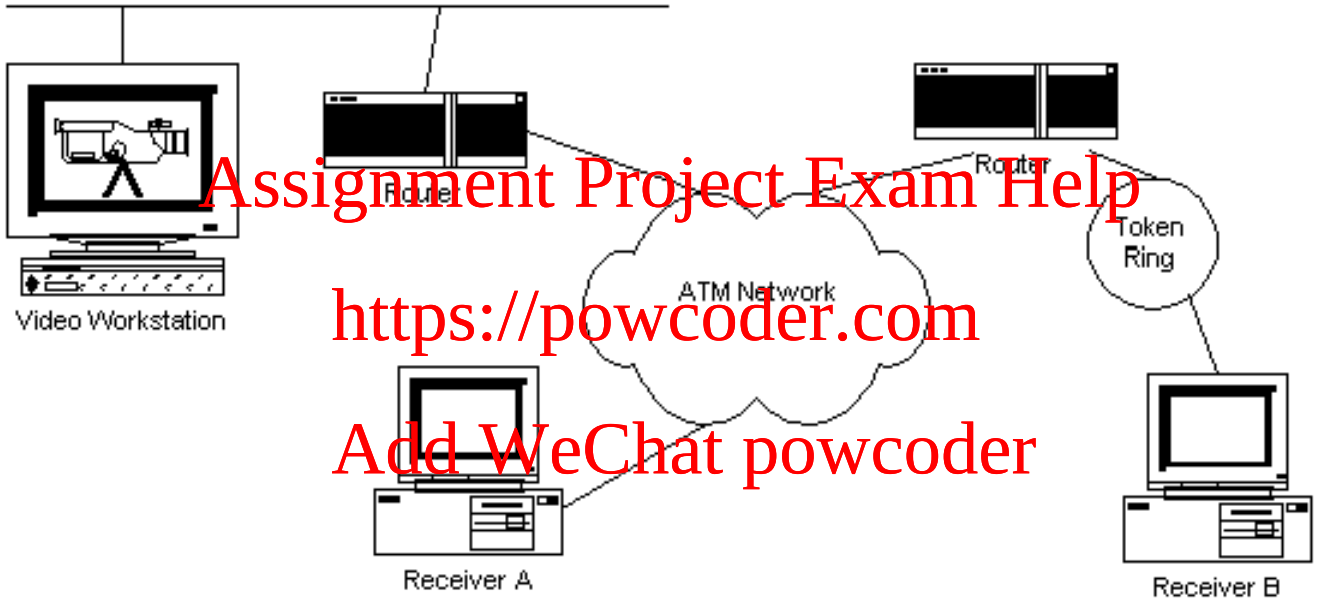


SESSION: IP destination address, IP protocol number, and destination port #
RSVP_HOP: IP address of RSVP-capable router that sent this message
TIME_VALUES: refresh period R.
STYLE: reservation style information not in flowspec or filterspec objects
FLOWSPEC: desired QoS in a Resv message.
FILTER-SPEC: set of packets that receive desired QoS in a Resv message.
SENDER_TEMPLATE: IP address of the sender in Path message.
SENDER_TSPEC: sender's traffic characteristics in Path message.
ADSPEC: carries end-to-end path information in Path message.
ERROR_SPEC: specifies errors in PathErr and ResvErr; confirmation in ResvConf.
POLICY_DATA: enables policy modules to determine whether request is allowed
INTEGRITY: cryptographic and authentication information to verify RSVP message
SCOPE: explicit list of senders that are to receive this message.
RESV_CONFIRM: receiver IP address that is to receive the confirmation.

RSVP Issues

- receiver-initiated –
  - » *need PATH messages since receiver does not know which path the data packets are taking*
  - » *advantage - sender does not need to know the number and specifics of reservations or the location of the receivers*
  - » *different receivers might request and receive different QoS*
- Sender-initiated –
  - » *sender had to maintain a reservation for each receiver*
  - » *the protocol would not scale for large multicast groups*
- Deployment of RSVP is encouraged in intranets where access control, scalability and security are not critical issues

Receiver-initiated reservations



Throughput limit for fast Ethernet on sender side: 100 Mbit/sec

Token Ring network limit: 16 Mbit/sec

Bandwidth requirement for standard video stream: 30 Mbit/sec

Scaling Issues of RSVP

- the control traffic and reservation state within a single large multicast session should be limited - solved by the RSVP design
  - » *path messages also sent as multicast messages, thus minimizing traffic*
  - » *reservation requests merged*
  - » *aggregation of reservations for flows with similar QoS requirements for the same destination*
- managing the reservation state for a large number of sessions
  - » *Information about thousands of reservations needs to be stored, accessed and changed*
  - » *degrades router performance*
  - » *the state required for RSVP grows with the bandwidth of the links, since more flows can be served and more state information needs to be managed*

Scaling Issues of RSVP

- enforcement of reservations
  - » *cost of classifying is proportional to the number of packets going through the router where classifier has to look into the network and transport layer headers of each packet*
  - » *cost of packet scheduling depends on the number of different service classes the router supports*