15-744: Computer Networking

L-19 RSVP & DiffServ

RSVP & DiffServ

- RSVP
- DiffServ architecture
- Assigned reading
  - [CF98] Explicit Allocation of Best-Effort Packet Delivery Service

Overview

- RSVP
- Differentiated services

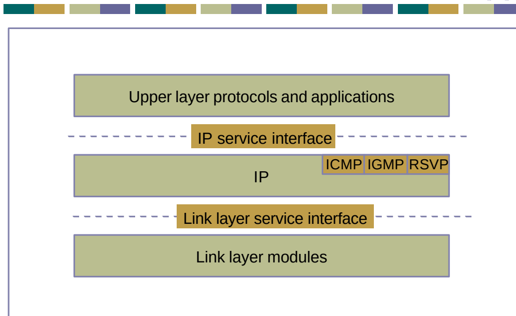
Components of Integrated Services

- Type of commitment
  - What does the network promise?
- Service interface
  - How does the application describe what it wants?
- Packet scheduling
  - How does the network meet promises?
- Establishing the guarantee
  - How is the promise communicated to/from the network
  - How is admission of new applications controlled?

Role of RSVP

- Rides on top of unicast/multicast routing protocols
- Carries resource requests all the way through the network
- At each hop consults admission control and sets up reservation. Informs requester if failure

Reservation Protocol: RSVP



RSVP Goals

- Used on connectionless networks
  - Should not replicate routing functionality
  - Should co-exist with route changes
- Support for multicast
  - Different receivers have different capabilities and want different QOS
  - Changes in group membership should not be expensive
  - Reservations should be aggregate – I.e. each receiver in group should not have to reserve
  - Should be able to switch allocated resource to different senders
- Modular design – should be generic "signalling" protocol
- Result
  - Receiver-oriented
  - Soft-state

Basic Message Types

- PATH message
- RESV message
- CONFIRMATION message
  - Generated only upon request
  - Unicast to receiver when RESV reaches node with established state
- TEARDOWN message
- ERROR message (if PATH or RESV fails)

RSVP Service Model

- Make reservations for simplex data streams
- Receiver decides whether to make reservation
- Control msgs in IP datagrams (proto #46)
- PATH/RESV sent periodically to refresh soft state
- One pass:
  - Failed requests return error messages - receiver must try again
  - No e2e ack for success

PATH Messages

- PATH messages carry sender's Tspec
  - Token bucket parameters
- Filtered or not-filtered
  - If F-Flag is set, store sender and flowspec
  - Otherwise, just add new link to tree
- Routers note the direction PATH messages arrived and set up *reverse path* to sender
- Receivers send RESV messages that follow reverse path and setup reservations
- If reservation cannot be made, user gets an error

RESV Messages

- Forwarded via reverse path of PATH
- Queuing delay and bandwidth requirements
- Source traffic characteristics (from PATH)
- Filter specification
  - Which transmissions can use the reserved resources
  - Reservation style
- Router performs admission control and reserves resources

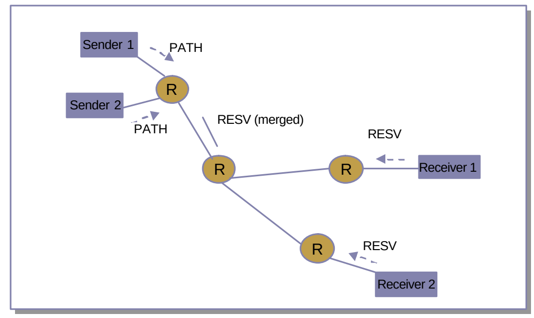
Router Handling of RESV Messages

- If new request rejected, send error message
- If admitted:
  - Install packet filter into forwarding dbase
  - Pass flow parameters to scheduler
  - Activate packet policing if needed
  - Forward RESV msg upstream

Reservation Styles

- How filters are used
- Three styles
  - Wildcard/No filter – does not specify a particular sender for group
  - Fixed filter – sender explicitly specified for a reservation
  - Dynamic filter – valid senders may be changed over time
- Receiver chooses but sender can force no-filter by setting F-Flag

PATH and RESV Messages



Changing Reservation

- Receiver-oriented approach and soft state make it easy to modify reservation
- Modification sent with periodic refresh

Routing Changes

- Routing protocol makes routing changes
- In absence of route or membership changes, periodic PATH and RESV msgs refresh established reservation state
- When change, new PATH msgs follow new path, new RESV msgs set reservation
- Non-refreshed state times out automatically

Packet Classifying and Scheduling

- Each arriving packet must be:
  - **Classified**: associated with the application reservation
    - Fields: source + destination address, protocol number, source + destination port
  - **Scheduled**: managed in the queue so that it receives the requested service
    - Implementation not specified in the service model, left up to the implementation

RSVP and Multicast

- Reservations from multiple receivers for a single sender are merged together at branching points
- Reservations for multiple senders may not be added up:
  - Audio conference, not many talk at same time
  - Only subset of speakers (filters)
  - Mixers and translators

Overview

- RSVP
- Differentiated services

DiffServ

- Analogy:
  - Airline service, first class, coach, various restrictions on coach as a function of payment
- Best-effort expected to make up bulk of traffic, but revenue from first class important to economic base (will pay for more plentiful bandwidth overall)
- Not motivated by real-time! Motivated by economics and assurances

Basic Architecture

- Agreements/service provided within a domain
  - Service Level Agreement (SLA) with ISP
- Edge routers do traffic conditioning
  - Perform per aggregate shaping and policing
  - Mark packets with a small number of bits; each bit encoding represents a class or subclass
- Core routers
  - Process packets based on packet marking and defined per hop behavior
- More scalable than IntServ
  - No per flow state or signaling

Per-hop Behaviors (PHBs)

- Define behavior of individual routers rather than end-to-end services – there may be many more services than behaviors
- Multiple behaviors – need more than one bit in the header
- Six bits from IP TOS field are taken for Diffserv code points (DSCP)

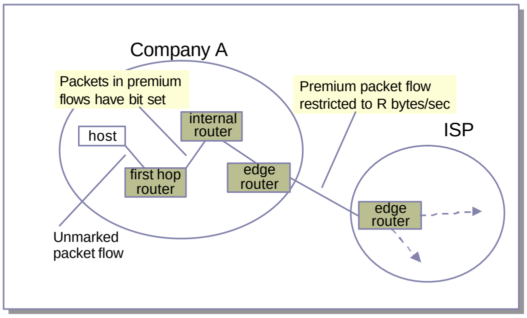
Per-hop Behaviors (PHBs)

- Two PHBs defined so far
- Expedited forwarding aka premium service (type P)
  - Possible service: providing a virtual wire
  - Admitted based on peak rate
  - Unused premium goes to best effort
- Assured forwarding (type A)
  - Possible service: strong assurance for traffic within profile & allow source to exceed profile
  - Based on expected capacity usage profiles
  - Traffic unlikely to be dropped if user maintains profile
  - Out-of-profile traffic marked

Expedited Forwarding PHB

- User sends within profile & network commits to delivery with requested profile
  - Signaling, admission control may get more elaborate in future
- Rate limiting of EF packets at edges only, using token bucket to shape transmission
- Simple forwarding: classify packet in one of two queues, use priority
  - EF packets are forwarded with minimal delay and loss (up to the capacity of the router)

Expedited Forwarding Traffic Flow



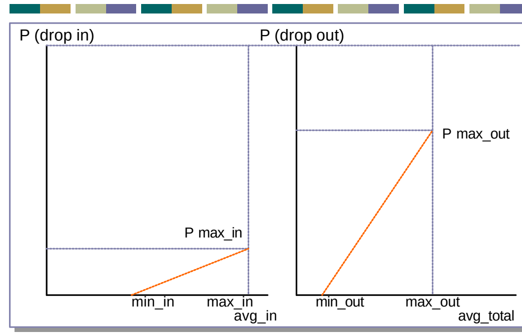
Assured Forwarding PHB

- User and network agree to some traffic profile
  - Edges mark packets up to allowed rate as “in-profile” or low drop precedence
  - Other packets are marked with one of 2 higher drop precedence values
- A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value
  - Implemented using RED with In/Out bit

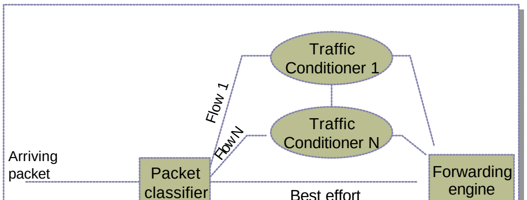
Red with In or Out (RIO)

- Similar to RED, but with two separate probability curves
- Has two classes, “In” and “Out” (of profile)
- “Out” class has lower Min_{thresh}, so packets are dropped from this class first
  - Based on queue length of all packets
- As avg queue length increases, “in” packets are also dropped
  - Based on queue length of only “in” packets

RIO Drop Probabilities

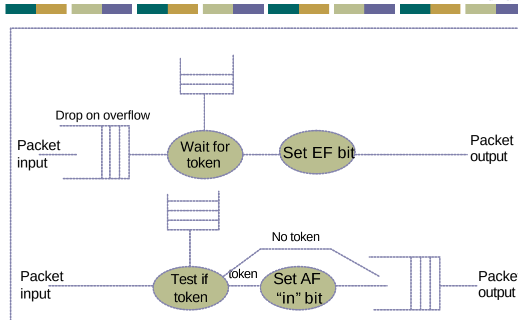


Edge Router Input Functionality



classify packets based on packet header

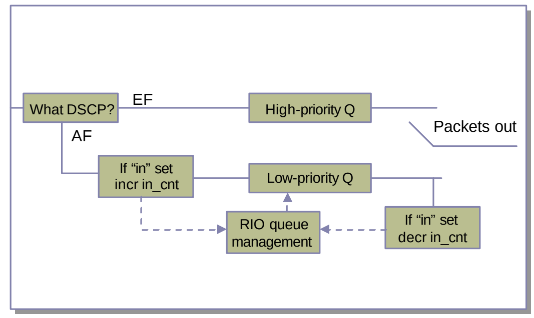
Traffic Conditioning



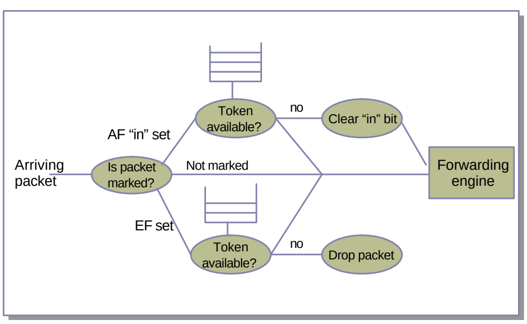
Output Forwarding

- 2 queues: EF packets on higher priority queue
- Lower priority queue implements RED "In or Out" scheme (RIO)

Router Output Processing



Edge Router Policing



Comparison

	Best-Efforts	Diffserv	Intserv
Service	• Connectivity • No isolation • No guarantees	• Per aggregation isolation • Per aggregation guarantee	• Per flow isolation • Per flow guarantee
Service Scope	• End-to-end	• Domain	• End-to-end
Complexity	• No set-up	• Long term setup	• Per flow setup
Scalability	• Highly scalable • (nodes maintain only routing state)	• Scalable (edge routers maintains per aggregate state; core routers per class state)	• Not scalable (each router maintains per flow state)

Next Lecture: Application Networking

- HTTP
- IPSec
- Firewalls
- Assigned reading
  - [BSR99] An Integrated Congestion Management Architecture for Internet Hosts
  - [PM95] Improving HTTP Latency