IETF Integrated Services

- Specification of Guaranteed Quality of Service (RFC 2212)
- Resource Reservation Protocol (RFC 2205)
  - Example of a real-time connection establishment protocol.
- The Use of RSVP with IETF Integrated Services.

Specification of Guaranteed Quality of Service (RFC 2212)

- The “fluid model” of service
- The traffic specification (TSPEC)
- The desired service specification (RSPEC)
- Specifying a service module (subnet, switch, trunk, …)
- Policing *vs.*reshaping

Introduction

- Guaranteed QoS is independent from connection establishment protocol or flow identification mechanism
  - RSVP
  - manual configuration
  - SNMP
- However: Guaranteed QoS only possible if every service element supports in the path supports it.
- Guaranteed service guarantees:
  - End-to-end delays
  - Queue overflows
- Guaranteed service does not guarantee:
  - Jitter
- Guaranteed service as extreme form of delay control for networks.

Fluid Service Model

- Definition: The fluid model at service rate *R* is the service that would be provided by a dedicated wire of bandwidth *R* between the source and the receiver.
- Note: In the fluid model, the flow's service is completely independend of that of any other flow!
- Algorithms and implementations:
  - Weighted Fair Queueing (WFQ) [Demers, Keshav, Shenker]
  - Jitter EDD [Verma, Zhang, Ferrari]
  - Virtual Clock [L. Zhang]
- General Definition [Goyal, Lam, Vin, NOSSDAV'95] :

$$GRC^i(p_f^0) = 0$$

$$CRC^i(p_f^j) = \max\{A^i(p_f^j), GRC^i(p_f^{j-1})\} + \frac{l_f^j}{r_f} \qquad j \geq 1$$

$$d_f^j \leq GRC^K(p_f^j) + \alpha^K - A^1(p_f^j)$$

Delays in the Fluid Service Model

- Observation: The delay of a flow bounded by a token bucket (r,b) and being served by a line with bandwidth R is bounded by b/R, as long as $R >= r$.
- Problem: Guaranteed service at rate R (R now is a share of overall bandwidth) approximates behavior of line with bandwidth R.
- Network element must ensure that local packet delay is less than $b/R+C/R+D$, where
  - C: rate-dependent error term.
    - Delay a datagram may experience due to the rate parameters of the flow.
    - Example: Serialization of datagram into ATM cells, with cells sent at frequency $1/r$.)
  - D: rate-independent error term (mostly occasional gaps in service)
    - Example: How long does a flow's data have to wait in a slotted network, once the data is ready.

Traffic Specification (TSPEC)

- TSPEC has form of token bucket plus a peak rate, a minimum policed unit, and a maximum datagram size.

  (b,r) : token bucket with bucket depth b and token rate r.

  p: maximum rate at which bursts can be injected into network.

  m: minimum policed unit. All datagrams smaller than m will be counted as having size m for policing purposes.

  M: maximum datagram size. Flow is rejected if its maximum datagram size is larger than MTU of link.

Desired Service Spec (RSPEC)

- *R*: rate
  - *R* must be greater or equal *r*
  - larger *R* reduces queueing delays
- *S*: slack term
  - Difference between the desired delay and the delay obtained by using a reservation level *R*.
  - Can be used by network element to reduce resource reservation.

Exported Information

- Network element's implementation of guaranteed service is characterized by the two error terms:
  - *C*: rate-dependent. (function of transmission rate)
  - *D*: rate-independent
- End-to-End sums of *C* and *D* (C_{tot} and D_{tot}) can be used in endnodes to compute maximal queueing delays.
- Partial sums C_{sum} and D_{sum} from most recent reshaping point downstream can be used to determine buffer requirement to assure no datagram loss.

Policing / Reshaping

- Policing:
  - at edge of network
  - traffic may exceed TSPEC
  - policing makes sure that $b(I) <= M + min(pI, rI+b-M)$
  - non-conforming datagrams should be treate as best-effort datagrams. (how?)

- Reshaping:
  - inside the network
  - delay non-conformant datagrams until they are within their TSPEC
  - amount of buffering required: $b + Csum + (Dsum * r)$

Resource **R**e**S**er**V**ation **P**rotocol (RSVP) (RFC 2205)

- RSVP as an Internet control protocol.
- RSVP itself <u>not</u> a routing protocol.
- RSVP supports <u>unicast</u> and <u>many-to-many multicast</u> applications.
- RSVP makes reservations for unidirectional data flow.
- RSVP is designed to handle large multicast groups, dynamic group membership, and heterogeneous receiver requirements => receiver-initiated QoS requests.
- “Soft” state
- Reservation setup = admission control + policy control
- Reservation “styles”

Reservation Model

- Reservation request: flow-descriptor = flow-spec + filter-spec
  - flow-spec specifies the QoS
    - RSPEC
    - TSPEC
  - filter-spec defines the set of data packets (the “flow”) to receive the QoS specified by flow-spec
    - generally: arbitrary subset of packets in given session
    - presently: filter spec defined in terms of sender IP address and port number SrcPort.
    - Problems:
      - segmentation (?)
      - IPv6 headers
      - IP-level security

RSVP Requests

- RSVP request messages originate at receivers and are passed to senders.
- Each intermediate node performs the following two operations:
- 1. Make a reservation on link. (admission control and policy control)
  - if fails, return error message to appropriate receiver.
  - details of admission control are link-layer technology specific.
- 2. Forward the request upstream.
  - Propagate request to appropriate senders.
  - Requests may be merged (remember heterogeneous requirements!)
- Basic reservation model is “one-pass”
  - Receiver sends request upstream, and each node in path either accepts or rejects.
  - Problem: no easy way for a receiver to find out the resulting end-to-end service.
- Solution: One-Pass-With-Advertising (OPWA)

Reservation Styles

- Reservation request includes a set of options that are collectively called reservation "style".
- Treatment of reservations for different senders: shared vs. distinct.
- Explicit list of selected senders vs. "wildcards".
- Shared reservations appropriate for multicast applications where multiple data sources are unlikely to transmit simultaneously.

Protocol Mechanism

- Two fundamental messages: RESV and PATH.
- RESV messages flow from receiver hosts to senders.
  – Create and maintain "reservation state" in each node.
- Each RSVP sender host transmits PATH messages downstream along unicast/multicast routes provided by routing subsystem.
  – PATH message contains:
  – previous hop address
  – sender template: describes format of packets that sender will originate
  – sender TSPEC
  – ADSPEC for OPWA: may be passed to local admission control.
- PATH messages sent with same source/destination addresses as data (for routing through non-RSVP clouds).

Merging Flow Specs; Teardown

- RESV message carries “largest” flow spec requested by all hops downstream.
- Flowspecs are opaque to RSVP: rules for comparing flowspecs are outside of RSVP.
- PATHTEAR vs. RESVTEAR
- teardown messages not transmitted reliably

Soft State

- RSVP maintains “soft state” in routers and hosts.
- Soft state is created and periodically refreshed by PATH and RESV messages.
  - State is deleted if no new matchin refresh messages arrive.
  - State can also be deleted with “teardown” messages.
- PATH and RESV messages are idempotent.
- Route change: PATH message will initialize state on new route, and future RESV messages will initialize reservation state there.
  - State on old route will eventually time out.
- Periodic retransmission to offset non-reliability of IP.
- Propagation of retransmitted control messages only if modify state.