QoS Path Management with RSVP

Roch Guérin   Sanjay Kamat   Shai Herzog
IBM Corp.

IBM IETF April 1997

Outline

- Motivations and goals
- General strategy and approach
- Path management rules
- Examples & discussion
- Summary
- Issues, extensions, and alternatives

Motivation and Goals

- Assumptions:
  - QoS routing with extended link state routing (OSPF)
  - Flow identification and QoS signaling using RSVP
- Implications to RSVP - QoS routing relationship: Need
  - QoS sensitive routing of PATH messages
  - Prevent route oscillations for QoS flows
    (Sticking with "satisfactory" existing path with reservation
    rather than changing to "best" current path
    $\Rightarrow$ "caching"/"pinning" of paths)
  - Prevent loops and avoid race conditions
- Goals: Propose a QoS Path management solution that
  - leverages RSVP soft state management
  - requires minimal changes to RSVP

General Strategy and Approach

- Set up QoS path as PATH message propagates
  Commit data traffic to the path after RESV succeeds

- What is "pinning"?
  Using previously obtained next hop to forward a PATH message,
  as against querying (QoS) routing each time

- What is "pinned"?
  "Path" used by an RSVP flow
  as against "route" computed by the routing algorithm

- Where is it done?
  In RSVP domain. Routing need not be aware of this pinning

- What is the duration of pinning?
  Path pinning kept *soft* by tying it to RSVP soft state

Path Management Rules

- Path (next hop) is pinned when
  - the next hop to forward PATH message is obtained by querying QOS Routing
- Path (next hop) is unpinned when
  - corresponding PATH state is removed (PATH_TEAR or time out)
  - change in PHOP or IP_TTL of the path state
  - local admission control failure during RESV processing
  - PATH_ERR with QoS_Path_failure is received
  - routing notifies RSVP of local interface failure

Path Management's Minimal Impact on RSVP

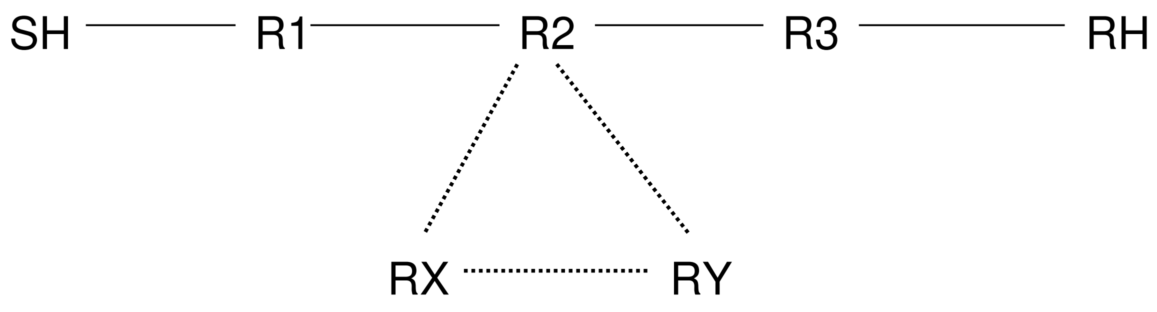
- RSVP/Routing interface:
  Currently:
  Route_Query([SrcAddress], DestAddress, Notify_flag)
  Proposed:
  Route_Query([SrcAddress], DestAddress, [TSpec], Notify_flag)
  For QOSR: TSpec is mandatory and Notify_flag is always set

- A flag in RSVP path state: set for a "pinned" path and
  updates require only minor mods to message processing rules

- An additional error code in PATH_ERR message
  (Signals upstream a QoS_Path_failure
  Needed because of potential time lag between PATH and RESV)

Examples - Loop Prevention (1)

Scenario 1



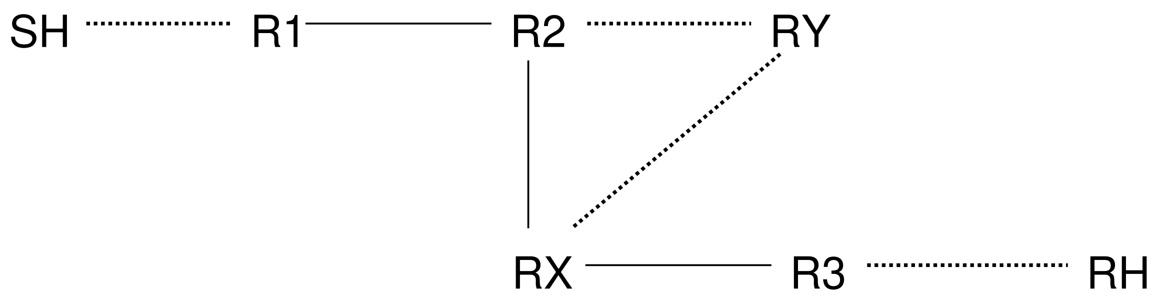
SH – Sender Host

RH – Receiver Host

- Correct QoS path: SH, R1, R2, R3, RH
  Inconsistent routing info: R2 believes RX to be next hop
- Loop detected at R2 because of change in PHOP

Examples - Loop Prevention (2)

Scenario 2

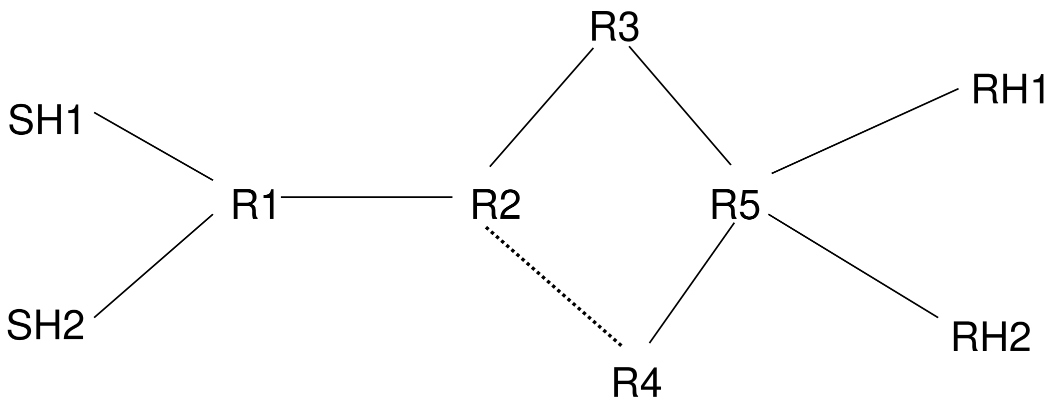


SH – Sender Host

RH – Receiver Host

- Correct QoS path: SH, R1, R2, RX, R3, RH
  Inconsistent routing info: RX believes RY to be next hop
- Loop detected at R2 because of change in IP_TTL

Examples - Preventing Race Conditions



- SH1 sends PATH message to RH1 and SH2 sends PATH message to RH2
- Relevant current QoS paths:
  (SH1, R1, R2, R3, R5, R6, RH1), (SH2, R1, R2, R3, R5, R6, RH2)
  Link R2-R3 can accommodate only one flow
- Paths for both flows get pinned but only one RESV succeeds
- Reservation failure will trigger unpinning for only one flow

Summary

- Benefits of proposed path management solution:
  - minimal impact on RSVP
  - prevents data looping, handles race conditions
  - leverages RSVP soft state mechanism
  - (QoS) routing protocol independent

Issues, Extensions, and Alternatives

- How to exploit paths with partial reservation set up?
  Possible approach: couple with policy information

- Support for source routing
  (mostly orthogonal to path management)