RSVP: Technical Issues

Lixia Zhang

Xerox Palo Alto Research Center

March 31, 1994

- **Route flapping -> route pinning**
- **A new proposal to filterspec usage**
- **How to compare flowspec**

I. Route Flapping

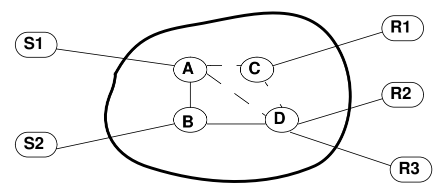
The Problem:

route flapping -->unstable service

3 suggested approaches:

- **Ask a question first: How badly do routes flap?**
- (as temp patch)**Maintain redundant reservations.**
- **Pin down the route for each reservation.**

Redundant Reservation



Route flapping between A-D and A-C-D, make reservation over both

When first time the route flaps from A-D to A-C-D ?

If the link C-D does not have adequate BW?

Red. Reservation: Pro’s & Con’s

Pro’s:

- **Simple**

Con’s:

- **Waste of resources.**
- **When one path has adequate resources and another doesn’t, reservation breaks unexpectedly.**

Route Pinning

- **Route stability is routing protocol’s problem.**
- **Expect that routing protocols will eventually support reservations well** (e.g. fix route flapping)
- **For the time being stablize the routes by adding a vanilla layer between routing and RSVP**

Route Pinning: The Basic Scheme

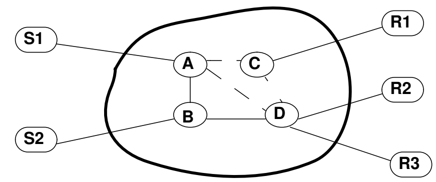
- **Path msgs: forward along both pinned and mcast routes**
- **Resv msgs**

  Pin down the route by the first reservation request (to a sender) until the reservation torndown.

  Forward all subsequent Resv msgs along pinned routes
- **Data packets are forwarded along the pinned routes only**

  (or treated as no reservation if no pinned route)

Route Pinning



How route flapping is handled : ignored by Resv msgs.
How new members are handled : by path msgs following mcast routing new resv msgs traverse the new branches
How topological change is handled: timeout the pinned route

Possible Extension to the Basic Scheme

Goal: Un-pin routes promptly to speed up recovery from topological changes

- **Setting a watchdog timer for each interface.**
- **Getting signal from the underneath routing protocol**
- **Running our own hop-by-hop Hello msg exchanges.**

Route Pinning: Pro's & Con's

Pro's:

- **Provide stable service**
- **Save resources (**as compared to red. reservation)

Con's

- **Added complexity**
- **Running our own hop-hop Hello replicates routing protocol function.**

II. RSVP Filterspec

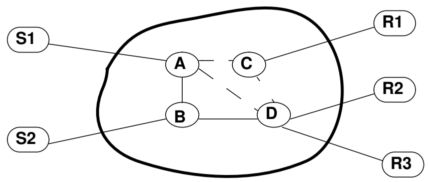
- **RSVP separates reservation from usage**
- **Function of the filterspec: determines which packets can use the reserved resources.**
- **Current filterspec format: [mask, value] pair.**
- **Typical filterspec usage** (as we know today)
  1. distinguish packets from different sources (e.g. to support dynamic filter reservation).
  2. distinguish substreams of packets from the same source (e.g. generated by layered audio.video

Recently Suggested Changes

Using separate mutlicast groups to distinguish substreams

- **heterogenous receivers, not all of them receive all the substreams.**
- **Instead of filter, this can be supported by using separate mcast group for each substream.**
- **Still use filters to distinguish different soruces (in dynamic-filter reservation)**

An Example



S1 and S2 each send 2 substreams of video
R2 receives both streams from both sources
R3 receives the baseband signal only

Assumption:
All sources use the same encoding scheme (?)

Using Diff Multicast Group: Pro's & Con's

Pro's:

- **Can simplify RSVP filterspec--only need to identify individual sources (e.g. by addr+port)**

Con's:

- **Still need filterspec**

  (may not save much processing cost)

- **Eliminating heterogenous sources?**