WHAT PROBLEM ARE WE SOLVING?

1. RESERVATION PERFORMANCE

1a) *Worst-case* time to install new state

State installation can take 30 seconds (or longer) if RSVP messages are lost.

This is a *functional* problem (protocol is not robust).

RSVP spec says RSVP msgs get low-loss service; perhaps this is unrealistic.

1b) *Worst-case* time to tear down state

2. RESERVATION COST

PROBLEM...

2. RESERVATION COST

2a) CPU time for processing RSVP messages

-- Design point ~ 10,000 sessions

=> ~ 300 msgs per second

2b) Network transmission cost for RSVP messages

-- Design point ~ 10,000 sessions

=> ~ 300 msgs per second

=> ~ 300Kbps

In current RSVP, improving performance and lowering cost require refresh interval R to be shifted in opposite directions! [Berger et al]

DESIGN CONSTRAINTS

- Backwards-compatible with pure soft state
- Handle multicast sessions

  (Problem: ACKing multicast Path messages)
- Handle route changes
- Useable across non-RSVP cloud

  (Problem: No route change notification)
- Auto-configuration

AGREEMENTS

o Distinguish "trigger" from "refresh" mesages

These are identical in standard RSVP; if route changes, refresh message becomes trigger message.

TRIGGER message:

-- Install new state or change or remove state.

-- Delivered reliably -- retransmit until ACKd.

-- Hard: Multicast messages (Path, PTear)

REFRESH message:

-- Sent periodically, no ACKs.

-- Various schemes to compress/reduce refresh msgs.

Design Parameters

- Number of sessions
- Average interarrival time for new sessions
- Average length of a session