RSVP Working Group

NEW ISSUES --

o UDP Encapsulation

o RSVP for IP6

Bob Braden

USC Information Sciences Institute

Dec 6, 1994

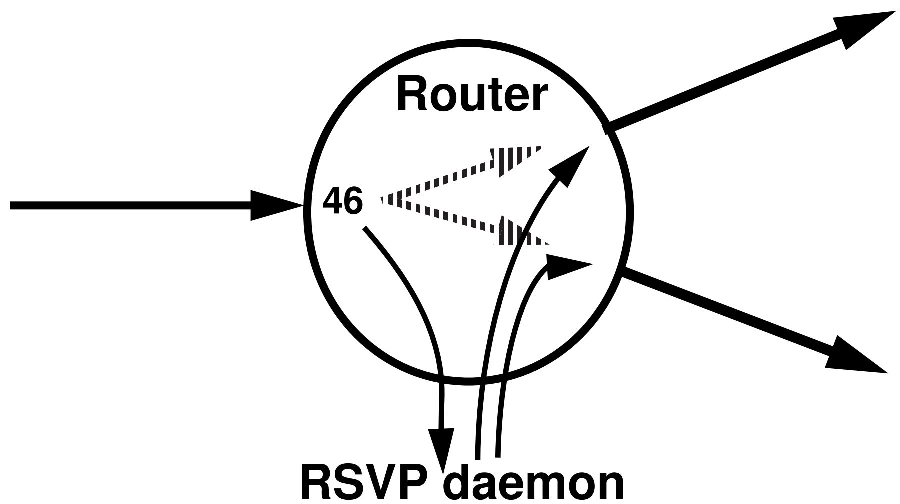
UDP Encapsulation

- **Review current RSVP message encapsulation:**

  -- RSVP msgs are sent as IP datagrams, protocol id = 46.

  -- Receiving node diverts all protocol 46 packets to RSVP

  -- RSVP suppresses routing in (multicast) sends



UDP Encapsulation

- **Rules:**

  -- Send Path msg => multicast to Gs *[= group addr of session]*

  -- Send Resv msg =>unicast to Phop *[= previous hop]*

- **Implications for host running RSVP:**

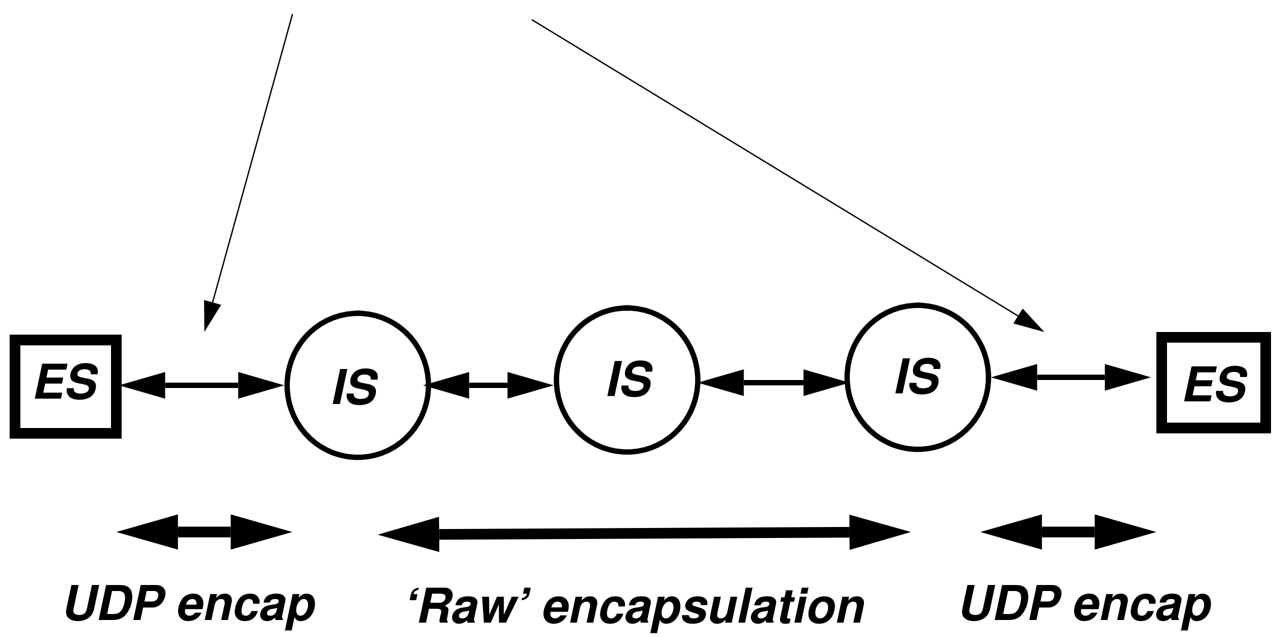
  -- RSVP must be privileged, to do “raw” network I/O.

  -- Kernel changes are required

At least in short term, these requirements impede installation of RSVP on hosts.

UDP Encapsulation

- **PROPOSE: UDP encapsulation of RSVP messages for first, last hop *(normally across LANs)***



UDP Encapsulation

a. Introduce: well-known group for RSVP: *Grsvp*.

b. UDP encapsulation =>

-- to send Path msg => multicast to *Grsvp* with TTL ~ 1;

-- to send Resv msg => unicast to *Phop*;

-- join group *Grsvp* to receive Path messages.

c. For each interface, RSVP daemon has two modes:

-- *ES mode* => UDP encapsulation

-- *IS mode* => raw RSVP (protocol 46)

UDP Encapsulation

Normally, can select modes automagically...

(1) If no protocol 46 intercept in kernel, or cannot do raw I/O
=> *ES mode* on all interfaces.

(2) If receive UDP-encap'd message on interface *I* for group *Grsvp*
=> *ES mode* on interface *I*.

UDP Encapsulation

(3) When in *ES mode* and no Path state active and

hear no other host's messages on *Grsvp*

=> send *UDP-encap* messages to *Grsvp* slowly.

UDP-encap message = Path(session 0, *Phop*)

You may ask: WHY NOT *ALWAYS* USE UDP ENCAPSULATION?

Answer: Then Path messages would flood the Internet... bad idea!

RSVP for IP6

ISSUES:

- **Bigger addresses**
- **Variable-length headers**
- **Flow IDs**

RSVP for IP6

Bigger Addresses:

- First order: trivial replacement 4 B -> 16 B
- BUT... larger size may force fragmentation/compression sooner.

Variable-Length IP Headers:

- IPv4 headers were VL in principle, but seldom in practice;

  IP6 headers are more likely to be VL [assumption].

- Filterspec representation must accommodate VL

  [probably not a *big deal*.]

RSVP for IP6

FLOW IDs:

Stifle the philosophy, while we ask:

how might IP6 flow ids fit into RSVP?

Three alternatives:

(A) Flow IDs are only a hint for routers; RSVP does not know about them.

(B) Flow IDs are carried in Path messages

(C) Flow IDs are carried in both Path and Resv messages

Flow IDs in Path Messages

SENDER **RSVP** **RECEIVER**

sender IP addr ***port*** ***flow id*** ***filter spec*** ***flowspec***

Path(S, P, Fid) ----> (path state) **<----- Resv(Filt(S,P), Q)**

Match

Install in Classifier:

(S,Fid) -> Q

Flow IDs in Path and Resv Messages

SENDER | **RSVP** | **RECEIVER**

sender IP addr *flow id* *filter spec* *flowspec*

Path(S, Fid) -------> (path state) **<----- Resv(Filt(S, Fid), Q)**

Match

Install in Classifier:

(S,Fid) -> Q