Resource reSerVation Protocol
RSVP

Lecture for S-38.180 QoS in the Internet
26.9.2002 Mika Ilvesmäki

Purpose
• In IntServ applications have to set up a reservation before transmitting traffic
  – RSVP is a signaling protocol for applications to reserve resources by setting up state in hosts and routers
RSVP properties

• End-to-end
  – requests from applications

• Per-flow method of signaling
  – fine granularity

• Originally intended for IP multicast
  – receiver-oriented setup
  – reservations are one-way only

RSVP design

• Not a routing protocol
  – designed to operate with current and future routing protocols

• Policy independent
  – RSVP is independent of the service architecture

• Soft state
  – times out unless state is refreshed
  – allows for state modification (original and refresh messages identical)

• Transparent operation through Non-RSVP clouds

• Reservations may be shared or not
Method of establishing flow state

- sender sends a PATH -message to the receiver specifying the traffic characteristics (Tspec) and setting up the path
- receiver responds with RESV-message to request resources for the flow (Rspec)

RSVP messages

- Sent either as raw IP (protocol 46) or in UDP
- PATH
  - sent downstream along the data path installing path state
- RESV
  - reservation requests sent by the receivers
**RSVP message format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td>Message types</td>
<td></td>
</tr>
<tr>
<td>RSVP checksum</td>
<td></td>
</tr>
<tr>
<td>Send TTL</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>RSVP length</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Class-num</td>
<td></td>
</tr>
<tr>
<td>C-type</td>
<td></td>
</tr>
<tr>
<td>Object content</td>
<td>(variable length)</td>
</tr>
</tbody>
</table>

**IP header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td>Message types</td>
<td></td>
</tr>
<tr>
<td>RSVP checksum</td>
<td></td>
</tr>
<tr>
<td>Send TTL</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>RSVP length</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Class-num</td>
<td></td>
</tr>
<tr>
<td>C-type</td>
<td></td>
</tr>
<tr>
<td>Object content</td>
<td>(variable length)</td>
</tr>
</tbody>
</table>

**PATH-message**

- Sent by the source
- Includes flow identification and flow characterization
- Sets up PATH-state in the router
RESV-message

- Sent by the receiver to reserve resources
- Contains the flow characterization and filter specification
- Sets up RESV-state in the router
- Flowspec may include
  - Tspec (both Guaranteed and Controlled-load)
  - Rspec (only in Guaranteed service)

<table>
<thead>
<tr>
<th>Flowspec</th>
<th>Filter Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines flow characteristics that will be requested from the routers</td>
<td>Defines flow id (or sender/senders)</td>
</tr>
</tbody>
</table>

Reservation types

- Three reservation types are defined
  - Wild-card filter
  - Fixed-Filter
  - Shared-explicit
- WF and SE are designed for multicast

<table>
<thead>
<tr>
<th>Sender selection</th>
<th>Reservations</th>
<th>Distinct</th>
<th>Shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit</td>
<td>Fixed Filter</td>
<td>Shared Explicit</td>
<td></td>
</tr>
<tr>
<td>Wildcard</td>
<td>ND</td>
<td>Wildcard-Filter</td>
<td></td>
</tr>
</tbody>
</table>
Reservation merging

- Reservations may be shared or merged
  - Depending on the reservation type and possible only within same type
  - router calculates the filterspec and flowspec to be sent to previous hop(s) according to reservation type

Reservations in action - FF

Sender selection
Explicit
Wildcard
Distinct
Reservations
Shared
Shared Explicit
Wildcard-Filter

S1 S2
S3 S4
S5 S6

Resv message direction

33 units to reserve

Total 12 for this interface
FF (S1, 4)
FF (S2, 6)
FF (S3, 2)
FF (S4, 5)
FF (S5, 4)
FF (S6, 2)
FF (S1,2,S2,3,S4, 5)
FF (S1,4,2,2)
FF (S4,4)
FF (S2,6,S4,2,S6,2)
FF (S2,3, S3,2,S5,4)

Total 9 for this interface
Reservations in action – WF

Resv message direction

33 units to reserve

WF (*, 5) 
WF (*, 5) 
WF (*, 5)

WF (*, 5) Total 5 for this interface
WF (*, 2) Total 2 for this interface
WF (*, 3) Total 3 for this interface
WF (*, 2) Total 2 for this interface
WF (*, 4) Total 4 for this interface

Reservations in action - SE

Resv message direction

33 units to reserve

SE (S1,S2;5) Total 5 for this interface
SE (S2,S4;5) Total 5 for this interface
SE (S3,S4;5) Total 3 for this interface
SE (S4,3) Total 3 for this interface
SE (S4,S6;2) Total 2 for this interface
SE (S2,S3,S5;4) Total 4 for this interface

Reservations in action - SE

Resv message direction

33 units to reserve

SE (S1,S2;5) Total 5 for this interface
SE (S2,S4;5) Total 5 for this interface
SE (S3,S4;5) Total 3 for this interface
SE (S4,3) Total 3 for this interface
SE (S4,S6;2) Total 2 for this interface
SE (S2,S3,S5;4) Total 4 for this interface

Sender selection
<table>
<thead>
<tr>
<th>Distinct</th>
<th>Shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit</td>
<td>Shared</td>
</tr>
<tr>
<td>Wildcard</td>
<td>ND</td>
</tr>
</tbody>
</table>
Adspec

• optional object in the PATH-message
• Consists of
  – default general parameters
  – Guaranteed Service fragment
  – Controlled Load Service fragment
• advertise receivers the characteristics of the end-to-end path

Adspec – Default general parameters

• Minimum Path Latency
• Path bandwidth
• Global break bit
  – cleared when Adspec is created by the sender
• IntServ Hop Count
• PathMTU
Adspec – Guaranteed Service fragment

- Ctot, Dtot, Csum and Dsum
- Guaranteed Service break bit
- Guaranteed Service General Parameters
  - overrides the values in default general parameters

Adspec – Controlled load service fragment

- Controlled-load service break bit
- Controlled-load service general parameters
  - overriding those presented in default general parameters
**OPWA**

- One pass with advertise
  - Sender includes Adspec in the PATH-message
  - with the aid of Ctot and Dtot the receiver is able to determine the path characteristics and form a more accurate RESV-message
  - receiver includes R and S (the slack term) in the RESV-message Rspec
    - Rspec includes also reservation type, filter specification, flow specification with Tspec and Rspec
- Without Adspec we have OP (One pass) and the RESV-message includes only the Tspec

---

**Slack term**

- Indicates the difference between the desired delay and the actual delay obtained with current R (bandwidth reservation)
- Allows the reservations some flexibility
  - balance between queue usage and service rate

![Diagram](image-url)
Confused?

• PATH(Tspec) describes how the traffic will behave
  – PATH will also establish the route
• The receiver calculates (maybe based on Adspec) what kind of reservations have to be made and puts this reservation request into RESV(Rspec)
  – RESV will make the reservations on the route

RSVP problems

• Implementation
  – RSVP is somewhat vague in its definitions and therefore difficult to implement consistently
    • RSVP API found in latest MS Windows APIs
    • compatibility between operating systems
  – For IntServ to function every node on the path must implement the IntServ functionality
    • especially true for the Guaranteed service
Alternative uses of RSVP and future issues

• RSVP-TE
  – RSVP with traffic engineering extensions
  – Will be presented in the MPLS-lecture

• Accounting and billing need to be integrated

• Authentication issues need to be resolved

Future of IntServ (and RSVP)

• In core there might be large amount of reservations to be update, so you have to:
  – not isolate individual flows
  – map flows into fixed number of service classes
  – don’t bother RSVP messages
  – keep state on the edges
  – > DiffServ