RSVP and other methods of QoS provisioning

Lecture for QoS in the Internet – course
24.11.2004 Mika Ilvesmäki

RSVP in a nutshell

• 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
    • but not necessarily only in IntServ
  – RSVP is a state establishment and maintenance protocol
Knowledge gain for this lecture

- After this lecture you will
  - Be able to explain RSVP design goals, actual design, RSVP properties and shortcomings in the design
  - Be able to explain how RSVP functions and what different types of reservations are and how they differ from each other
  - Be able to explain alternative uses for RSVP and RSVP-TE
  - Be able to explain what CR-LDP is and how it compares to RSVP

RSVP design goals

- Heterogeneous receivers
  - Receiver oriented reservation style
- Dynamic membership
  - Data transfer is not controlled by RSVP
- Sharing of resources
  - Reservation styles (WF, FF, SE)
- Adaptation to network dynamics
  - Soft-state approach
- Independence of architectural components
  - Flow specs, admission control, packet classification, scheduling, routing
- Controllable (and modifiable) protocol overhead
  - Refresh period parameter
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

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
Method of establishing flow state
• sender sends a PATH –message to the receiver specifying the traffic characteristics (Tspec) and setting up the path (path state)
• receiver responds with RESV-message to request resources for the flow (Rspec) and sets up flow state.

Soft state
• RSVP sends (by default) PATH and RESV messages periodically
  – If states are not updated regularly they time out.
• PATH refreshes make it possible to adapt to path/(multicast distribution tree) property changes
• RESV refreshes may incorporate changes altered reservations -> adaptive QoS
  – Old reservations die out, no additional state maintenance
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>IP header</th>
<th>common header</th>
<th>object header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version Flags</td>
<td>Message types</td>
<td>RSVP checksum</td>
</tr>
<tr>
<td>Send TTL Reserved</td>
<td>RSVP length</td>
<td></td>
</tr>
<tr>
<td>Length Class-num</td>
<td>C type</td>
<td></td>
</tr>
<tr>
<td>Object content</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NULL</th>
<th>SESSION</th>
<th>PATH</th>
<th>RESV</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSVP_HOP</td>
<td>TIME.VALUE</td>
<td>PATHErr</td>
<td>RESV.Tear</td>
</tr>
<tr>
<td>STYLE</td>
<td>FLOWSPEC</td>
<td>PATH.Tear</td>
<td>RESV.Tear</td>
</tr>
<tr>
<td>FILTER_SPEC</td>
<td>SENDER_TEMPLATE</td>
<td>RESV_Confirm</td>
<td>PATH</td>
</tr>
</tbody>
</table>
PATH-message

- Sent by the source
- Includes flow identification and flow characterization
- Sets up PATH-state in the router
  - Note: RSVP does not restrict a source from transmitting data even without any receiver having made reservation setup

<table>
<thead>
<tr>
<th>PHOP</th>
<th>Sender Template</th>
<th>Sender TSpec</th>
<th>Adspec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Previous router</td>
<td>Filter Spec</td>
<td>OPWA-information (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(defines uniquely the sending host and flow)</td>
<td>Defines flow characteristics</td>
</tr>
</tbody>
</table>

RESV-message

- Sent by the receiver to reserve resources
- Contains the flow characterization and filter specification (reservation type, WF, FF, SE)
- 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></td>
<td></td>
</tr>
<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></td>
<td>Shared Explict</td>
</tr>
<tr>
<td>Wildcard</td>
<td>ND</td>
<td></td>
<td>Wildcard-Filter</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
  - Reservation messages (RESV) propagate only as far as reservation request meets an existing distribution tree with sufficient resources. (Reservation merged).
Reservations in action - FF

S1  S2
S3  S4
S5  S6

Resv message direction

FF (S1, 4)  FF (S2, 6)  FF (S3, 2)  FF (S4, 5)  FF (S5, 4)  FF (S6, 2)

33 units to reserve

Distinct reservations with explicit sender selection (set of senders have their own reservations). Allows for simultaneous sending.

Reservations in action – WF

S1  S2
S3  S4
S5  S6

Resv message direction

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

33 units to reserve

Shared reservation with wildcard sender selection: Single reservation for the receiver shared by all flows (traffic) from all upstream senders.
Reservations in action - SE

Resv message direction

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

SE (S1,S2;5)
SE (S3,S4; 5)
SE (S5,S6; 4)

33 units to reserve

SE (S2,S4;5)
SE (S1,S2; 2)
SE (S4, 3)
SE (S4,S6; 2)
SE (S2,S3,S5; 4)

Total 5 for this interface
Total 3 for this interface
Total 4 for this interface

Total 5 for this interface
Total 3 for this interface
Total 4 for this interface

Shared reservation with explicit sender selection: single reservation shared by selected senders (not all as in WF).

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

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
• Hierarchical RSVP
  – reserve large pipes, classify packets to pipes at the edge.
    • reduction of reservation state, fewer choices for packet scheduling but still looking at the source and destination
• Accounting and billing need to be integrated
• Authentication issues need to be resolved
Using RSVP-TE for label distribution in MPLS

- New functions:
  - Label distribution
  - Explicit routing, rerouting, route tracking
  - Bandwidth/Resource reservation
- New objects
  - PATH-message
    - LABEL_REQUEST
    - EXPLICIT_ROUTE
    - RECORD_ROUTE
    - SESSION_ATTRIBUTE
  - RESV-message
    - LABEL
    - RECORD_ROUTE

RSVP-TE in action

- Addition of Label_request –message in RSVP PATH-message
  - Downstream label allocation
- Addition of Label –object to be carried in RSVP RESV-message
  - Labels propagate upstream in the RESV-message
- LSPs are set up with FF-reservation
Domain wide QoS

- a.k.a Constraint based routing (CR) or QoS routing (QoSR)
- Calculate the route so that multiple constraints are met and that the route is optimal for every constraint
  - Constraints: delay, bandwidth, etc. and/or administrative
- Problems: route oscillation, path capacity
- Could be used together with a signalling protocol (RSVP or CR-LDP) that has knowledge on the constraint values

CR-LDP

- LDP (label distribution protocol) is defined for distribution of labels in MPLS-networks.
  - Constraint-based Routing LDP (CR-LDP) uses information not available for routing protocols when setting up the paths.
    - Explicitly routed LSPs
- CR-LDP is simple, scalable (TLV), open and non-proprietary signalling protocol
CR-LDP and QoS

- Strict and loose explicit routing
  - Route pinning
- Specification of traffic parameters (peak rate, delay variation…)
- Use of resource classes (instead of traffic parameters)
- LSP pre-emption
  - Set-up priority better than holding priority may preempt an existing LSP

Comparing RSVP_TE and CR-LDP

<table>
<thead>
<tr>
<th>Property</th>
<th>CR-LDP</th>
<th>RSVP_TE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport mechanism</td>
<td>Transport on TCP (reliable)</td>
<td>Raw IP packets (unreliable)</td>
</tr>
<tr>
<td>State management</td>
<td>Hard state</td>
<td>Soft state; needs per-flow refresh management</td>
</tr>
<tr>
<td>Maps required for LSP set-up and maintenance</td>
<td>Based on LDP for MPLS</td>
<td>Based on RSVP, may require major changes</td>
</tr>
<tr>
<td>Base architecture</td>
<td>Based on RSVP, may require major changes</td>
<td>Based on RSVP, may require major changes</td>
</tr>
<tr>
<td>Signalling of QoS and traffic parameters</td>
<td>Can signal DiffServ and ATM traffic classes</td>
<td>Extendable, currently based on IntServ</td>
</tr>
<tr>
<td>Types of LSPs</td>
<td>Strict, loose, and loose pinned</td>
<td>Strict and loose, no pinning</td>
</tr>
<tr>
<td>Models of label distribution and LSP set-up</td>
<td>All modes</td>
<td>Only downstream on demand</td>
</tr>
<tr>
<td>Failure notification</td>
<td>Reliable procedure</td>
<td>Unreliable procedure</td>
</tr>
<tr>
<td>Loop detection/prevention</td>
<td>Employs path vector TLV to prevent Label Request –loops. Hop Count TLV used to find looping LSPs</td>
<td>May be done using Record_Route –object</td>
</tr>
</tbody>
</table>

- Both can be used to establish LSPs
- CR-LDP works over TCP, RSVP works over IP (or UDP)
- Direction of resource reservations is different