RSVP and other methods of QoS provisioning

Lecture for QoS in the Internet – course
S-38.180
9.10.2003 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
    • but not necessarily only in IntServ
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>IP header</th>
<th>common header</th>
<th>object header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Flags</td>
<td>Message types</td>
</tr>
<tr>
<td>Send TTL</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Class-num</td>
<td>C type</td>
</tr>
<tr>
<td></td>
<td>Object content (variable length)</td>
<td></td>
</tr>
</tbody>
</table>

Null SESSION
RSVP HOP TIME_VALUE
STYLE FLOWSPEC
FILTER_SPEC SENDER_TEMPLAT
SENDER_TSPEC ADSPEC
ERROR_SPEC POLICY_DATA
INTEGRITY SCOPE
RESV_CONFIRM

PATH RESV
PATHErr RESVErr
PATHTear RESVTear
RESVConf

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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distinct</td>
</tr>
<tr>
<td>Explicit</td>
<td>Fixed Filter</td>
</tr>
<tr>
<td>Wildcard</td>
<td>ND</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

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

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

33 units to reserve

FF (S1, 2, S2, 3, S4, 5)
FF (S1, S4, S2, 2)
FF (S4, 4)
FF (S2, 6, S4, 2, S6, 2)

Total 12 for this interface
Total 12 for this interface
Total 9 for this interface
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

- $C_{tot}$, $D_{tot}$, $C_{sum}$ and $D_{sum}$
- 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
    • EXPPLICIT_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>Mgmt required for LSP set-up and maintenance</td>
<td>Request, mapping</td>
<td>Path, Resv, Resv_Conf</td>
</tr>
<tr>
<td>Base architecture</td>
<td>Based on LDP for MPLS</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