Agenda

- Integrated Services Principles
- Resource Reservation Protocol
- RSVP Message Formats
- RSVP in a IP Multicasting Environment
Design Intentions

• The Internet was based on a best effort packet delivery service, but nowadays the Internet carries many more different applications
• Some applications require special bandwidth and delay; real time transmissions also became important
• The Integrated Services Model
  – was introduced to guarantee predictable network behavior for these applications (RFC 1633)
• The underlying Internet architecture should not be modified to support QoS

Design Intentions

• Resources (e.g., bandwidth) must be explicitly managed
  – in order to meet application requirements for packet delay and throughput
• This implies that resource reservation and admission control are key building blocks of IntServ
Design Intentions

- **Basic idea:** Client reserves network resources on every router "upstream" to the server
  - using a signaling protocol
  - Resource ReSerVation Protocol (RSVP), RFC 2205
  - resource reservation is initiated by the receiver

- **Flow-based concept**
  - "Flow" = packets of the same session, identified by socket parameters

- **Using RSVP a client requests specific QoS parameters**
  - at each router along the upstream path
  - if possible, routers reserve resources for this flow
  - each flow then uses a single, designated path

IntServ Components

- **Generally IntServ consists of three traffic-control and one reservation mechanisms:**
  - Packet scheduler:
    - actually manages the forwarding of different packet streams
    - additionally metering and traffic policing is done at each router
    - additionally traffic shaping is done at the sender
  - Packet classifier:
    - each incoming packet must be mapped into some class to allow traffic control; all packets in the same class get the same treatment from the packet scheduler
    - a class might correspond to a broad category of flows, e.g., all video flows or all flows attributable to a particular organization
    - on the other hand, a class might hold only a single flow
IntServ Components

- Generally IntServ consists of three traffic-control and one reservation mechanisms (cont.):
  - Admission control:
    • determines whether the node has sufficient available resources to supply the requested QoS
    • that means whether a new flow can be granted the requested QoS without impacting earlier guarantees
  - Reservation setup protocol,
    • which is necessary to create and maintain flow-specific state in the endpoint hosts and in router along the path of a flow
    • protocol called RSVP (for “ReSerVation Protocol”)
- Policy Control
  - additionally needed but outside the scope of IntServ
    • determines whether the user has the administrative permission to make a reservation including authentication of request
IntServ Components

HOST

Application
RSVP Process
I need this QoS
Policy Control
Admission Control
Classifier Packet Scheduler Traffic Shaping

Can requestor made this reservation?
Do I have the resources?

HOST

RSVP

I need this QoS
Policy Control
Admission Control
Classifier Packet Scheduler Traffic Shaping

Can requestor made this reservation?
Do I have the resources?

IntServ Components

ROUTER

RSVP Routing Process RSVP Process

Can requestor made this reservation?
Do I have the resources?

Determine route and QoS class

ROUTER

RSVP Routing Process RSVP Process

Can requestor made this reservation?
Do I have the resources?

Host Data

Classifier Packet Scheduler Traffic Shaping

schedule packet and meter traffic

© 2009, D.I. Dr. Wolfgang Kampichler
IntServ Components

- **Output queue can be controlled by a token bucket model**
  - token bucket parameters can be reserved via RSVP, the Resource Reservation Protocol
    - token rate, bucket depth
    - maximum packet size, peak rate, minimum policed size
- **RSVP uses these token bucket parameters in its flow descriptor field**

Two IntServ Classes

- **“Guaranteed QoS” (RFC 2212)**
  - guarantees a maximum queuing delay, minimum of interference from best-effort traffic, isolation between reserved flows;
    - Assuming no failure of network components or changes in routing during the life of the flow
  - the datagram delivery time is bounded
- **“Controlled Load” (RFC 2211)**
  - simulates unloaded conditions only
  - better than best effort
  - network guarantees that the reserved flow will reach its destination with a minimum of interference from the best-effort traffic
The Problem with Real-Time Traffic

- The sender packetizes some data stream and sends it to the receiver.
- The receiver de-packetizes this packets into the original data stream and playbacks it.
- Receiver must smooth transmission jitter by buffering the data
  - introducing an offset delay

Real-Time Traffic

![Diagram showing the process of real-time traffic and the issues with delay and offset.](diagram.png)
Types of Real-Time Applications

- **Intolerant Real-Time Applications**
  - are very sensitive to jitter — require Guaranteed QoS
  - e.g. conference applications, telephony

- **Tolerant Real-Time Applications**
  - agree with nominal amount of jitter — require Controlled Load Service only
  - e.g. audio and video streaming

Guaranteed QoS Class

- **Applications that have hard real time requirements, will require guaranteed service**
  - real-time multimedia applications, such as video and audio broadcasting systems that use streaming technologies, cannot use datagram’s that arrive after theirs proper playback time
Guaranteed QoS Class

- Let the application control the network’s queuing delay
  - by specifying token bucket parameters and bandwidth for requested reservation
  - token bucket depth and data rate are part of the flow descriptor
- If possible (available resources, permission) the router reserves the requested resources for this flow
  - using the flow descriptor’s parameters

Controlled Load Service Class

- Is intended to support the class of applications that are highly sensitive to overloaded conditions in the Internet
- “Best effort service under unloaded conditions”
- Application may announce an estimation of the traffic it will generate to the network
  - using a traffic specification (TSpec) as part of the flow descriptor
- A small amount of packet loss is still possible
  - service degrades quickly under overloaded conditions
    - if an application uses the Controlled Load service, the performance of a specific data flow does not degrade if the network load increases
Agenda

• Integrated Services Principles
• Resource Reservation Protocol
• RSVP Message Formats
• RSVP in a IP Multicasting Environment

RSVP

• RSVP is an Internet control protocol
  – RSVP depends on an underlying routing mechanism and IP (multicast)
  – RFC 2205
• RSVP messages are encapsulated within raw IP or UDP
• For any particular flow the receiver can reserve resources along its path to the sender
• Note: RSVP does not
  – QoS routing like ATM (PNNI routing and QoS signaling)
  – admission control and packet scheduling
  – forwarding/routing of data packets
RSVP

- RSVP makes resource reservations for both unicast and multicast applications
  - adapting dynamically to changing group membership as well as to changing routes
- RSVP is simplex
  - it makes reservations for unidirectional data flows.
- RSVP is receiver-oriented
  - the receiver of a data flow initiates and maintains the resource reservation used for that flow

RSVP

- RSVP maintains "soft" state in routers and hosts
- RSVP is not a routing protocol but depends upon present and future routing protocols
- RSVP transports and maintains traffic control and policy control parameters that are opaque to RSVP
- RSVP provides several reservation models or "styles" to fit a variety of applications
- RSVP provides transparent operation through routers that do not support it
- RSVP supports both IPv4 and IPv6
RSVP Messages

- RSVP uses seven message types
  - two required message types
    - PATH
    - RESV
  - five optional message types
    - PATH ERROR
      - Sent by the receiver or router notifying the sender when errors in PATH are found (fundamental format or integrity check fault)
    - PATH TEARDOWN
      - Are sent to multicast group with sender's source address when the PATH must be flushed from the database (e.g. link failure) or because the sender is exiting the multicast group
    - RESV ERROR
      - When errors in RESV message are found, RESV ERROR sent by sender or router informing the receiver
    - RESV CONFIRM
    - RESV TEARDOWN

RSVP Example

FTP Server
2 Mbit/s videostream
Multicast videostream
FTP session
Subscription to the sender’s multicast service using IGMP
FTP Server
FTP Client
Receiver
Subscription to the sender’s multicast service using IGMP
Sender
2 Mbit/s videostream
Normal Subscription to Multicast Service

Sender
2 Mbit/s videostream

Graft

Subscription to the sender’s multicast service

FTP Server

FTP Client

Receiver

RSVP in Action

- Sender sends RSVP **PATH** messages periodically
  - PATH messages include the IP address of the interface through which it is sent
  - these messages describe the data they intend to send

- Each RSVP router catches the PATH messages
  - it saves the previous hop address, writes its own address as the previous hop, and sends the updated PATH along the same way the application data is using

- Routers which do not implement RSVP simply forward the PATH message to the next-hop router
  - in a network which uses the IntServ model it is not a requirement that all routers implement RSVP
Demand for More Network Resources

Sender
2 Mbit/s videostream

FTP Server
FTP Client
Receiver

RSVP Functionality Announcement via PATH
RSVP in Action

- Receivers identify interesting flows by the TCP/UDP port number
- Receivers initiate a resource reservation by sending **periodically** a RESV message to the next-hop upstream to the sender
  - containing a *flow descriptor* for traffic description and reservation identification
- The RESV messages take a the reverse way of PATH messages, means they go from receiver to sender

Resource Reservation Upstream 1

Sender
2 Mbit/s videostream

FTP Server

FTP Client

Receiver

I can easily reserve additional resources for video
I have not enough resources for both FTP and video, but video comes first and FTP will get the rest.

Resource Reservation Upstream 3

FTP gets best effort service only

Resource reservation for the “Receiver” completed
RSVP in Action

- When a router receives a RESV:
  - it passes the request to its admission control function
    • to find out whether there are sufficient resources to implement the reservation request
  - it passes the request to its policy control function
    • to determine whether policy rules allow the user to make the reservation request
  - if both checks succeed,
    • it sets parameters in the packet classifier and scheduler functions to implement the requested reservation
  - it forwards the RESV message to its upstream neighbor

- The last router sends a confirmation message back to the receiver (optional RESV CONFIRM)
  - if routers cannot adjust some resources according to RESV, they will refuse the reservations and inform the receiver
  • if they can, they merge reservation requests being received and request a reservation from the previous hop router

RSVP in Action

- RSVP is often used in IP multicast environments
  - several RESV messages for the same sender can be merged together
    • for example applications carrying voice or video over IP
  - when the current reservation is equal or greater than that being requested, that RESV message need not be forwarded upstream any longer

- Non-RSVP routers within the path are weak links; service degrades to “best effort”
Removing Reservations

- Reservations can be removed by RSVP TEARDOWN messages:
  - **Sender initiated**: Using PATH-TEAR to eliminate all downstream path states and associated reservation tables
  - **Receiver initiated**: Using RESV-TEAR messages to eliminate all upstream reservation states

RSVP Soft States

- RSVP is a state protocol, which means that each router in the reservation process has to maintain a resource state for each RSVP session and update it regularly
- But IP/UDP is an insecure protocol, so it can easily be that a message, which requests to tear down the QoS connection, gets lost
RSVP Soft States (cont.)

- As a consequence the connection will stay in the net until the router memory for storing states is full.
- To avoid this situation a so called **SOFT** state is introduced:
  - in a Soft state both sender and receiver repeat their PATH/RESV messages **periodically**
  - if a message does not arrive several times one after another, the states along the route will be deleted
  - if the loss of a data packet is the reason of the packet's non-appearance then the reservation will be initialized again with the following RESV message.

RSVP Terminology

- **RSVP session is identified**
  - by the **session specification** (session ID)
    - receiver address (destination address of a flow), protocol type, destination port (UDP, TCP)
  - by a **unique flow descriptor**
    - defining the required QoS and traffic characteristics

- **A flow descriptor consist of**
  - **filter** specification (filterspec)
  - **flow** specification (flowspec)
Flow Descriptor Components

- **Filter specification**
  - identifies packets belonging to a specific flow; using sender IP address and source port
  - used by the packet classifier

- **Flow specification contains**
  - Traffic specification (TSpec)
    - describes the traffic characteristics of the requested service using a token bucket filter
  - Service request specification (RSpec)
    - specifies the requested QoS (bandwidth, max delay, max packet loss rate)

RSVP Reservation Styles

- **A reservation request (RESV) contains**
  - a set of options called the “Reservation Style”

- **Reservation styles describe**
  - **how** the receiver wants to get the data from different senders in one session
  - the receiver can establish
    - **Distinct** reservation for each upstream sender or
    - **Shared** reservation for all packets of selected senders in one session
  - senders for a reservation request are selected through
    - **Explicit** server list
      - explicit list of all selected senders
      - filterspec must identify exactly one sender
    - **Wildcard**
      - implicitly selects all the senders to the session
      - filterspec is not needed
### RSVP Reservation Styles

#### Distinct versus Shared

<table>
<thead>
<tr>
<th>Sender Selection</th>
<th>Distinct Reservation</th>
<th>Shared Reservation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit</td>
<td>Fixed-Filter (FF) Style</td>
<td>Shared-Explicit (SE) Style</td>
</tr>
<tr>
<td>Wildcard</td>
<td>(Not Defined)</td>
<td>Wildcard-Filter (WF) Style</td>
</tr>
</tbody>
</table>

#### Reservation Styles

- **Fixed-filter style (FF)**
  - one reservation for data packets from a particular sender;
  - for applications that need one pipe per source (e.g. video)
  - packets from different senders that are in the same session do not share reservations

- **Shared-explicit style (SE)**
  - a single reservation shared by selected upstream senders
  - receiver controls explicitly who can use the shared pipe
    - a sender list must be included into the reservation request from the receiver
Reservation Styles (cont.)

- **Wildcard-filter style (WF)**
  - a single reservation for all sources sending to the same RSVP session
    - reservations from different senders are merged together along the path so only the biggest reservation request reaches the senders
    - supports self-limiting applications (e.g. audio conference)
  - if new senders appear in the session the reservation is extended to these new senders

---

Agenda

- Integrated Services Principles
- Resource Reservation Protocol
- RSVP Message Formats
- RSVP in a IP Multicasting Environment
RSVP Types

- **Native RSVP**
  - encapsulated within IP only
  - using protocol number 46
- **UDP-encapsulated RSVP**
  - exceptionally, for some end-systems not able to use raw IP network I/O service
  - ports 1698 and 1699

Native RSVP

- All RSVP messages consist of the same header format and a body
- The body consists of several **objects**
  - containing necessary information for the resource reservation
  - e.g. flow descriptor and reservation style
**Header Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>RSVP protocol number; current version is 1</td>
</tr>
<tr>
<td>Flags</td>
<td>No flags defined yet</td>
</tr>
<tr>
<td>Message Type</td>
<td>PATH, RESV, PATH_ERR, RESV_ERR, PATH_TEAR, RESV_TEAR, RESV_CONF</td>
</tr>
<tr>
<td>Checksum</td>
<td>Used by receivers to detect RSVP message errors</td>
</tr>
<tr>
<td>Send TTL</td>
<td>Contains the IP TTL value the message was sent with</td>
</tr>
<tr>
<td>RSVP Length</td>
<td>Total length of message including header and body</td>
</tr>
</tbody>
</table>

**Object Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Object length in bytes; must be a multiple of four</td>
</tr>
<tr>
<td>Class-Number</td>
<td>Identifies the object class (see below)</td>
</tr>
<tr>
<td>C-Type</td>
<td>Specifies the object type within the class number; IPv4 and IPv6 uses different object types</td>
</tr>
<tr>
<td>Object Contents</td>
<td>Up to 65528 bytes long</td>
</tr>
</tbody>
</table>
### RSVP Class-Numbers

<table>
<thead>
<tr>
<th>Object-class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
<td>object content ignored by receiver</td>
</tr>
<tr>
<td>Session</td>
<td>destination’s IP address, port number and IP protocol ID</td>
</tr>
<tr>
<td>RSVP_Hop</td>
<td>IP address of node that sent this message</td>
</tr>
<tr>
<td>Time_Values</td>
<td>refresh period for PATH and RESV messages</td>
</tr>
<tr>
<td>Style</td>
<td>reservation style</td>
</tr>
<tr>
<td>Flowspec</td>
<td>specifies required QoS in RESV messages</td>
</tr>
<tr>
<td>Filterspec</td>
<td>which data-packets take advantage of QoS</td>
</tr>
<tr>
<td>Sender_Template</td>
<td>sender’s IP address and additional ID-information</td>
</tr>
<tr>
<td>Sender_Tspec</td>
<td>defines traffic characteristics of the sender’s data flow</td>
</tr>
<tr>
<td>Adspec</td>
<td>advertising information for the traffic-control modules</td>
</tr>
<tr>
<td>Error_Spec</td>
<td>specifies error (PathErr or ResvErr message) or confirmation (ResvConf)</td>
</tr>
<tr>
<td>Policy_Data</td>
<td>to support decisions of policy-modules</td>
</tr>
<tr>
<td>Integrity</td>
<td>cryptographic data to authenticate the origin node and message-contents</td>
</tr>
<tr>
<td>Scope</td>
<td>explicit list of server hosts (this object appears in RESV* messages)</td>
</tr>
<tr>
<td>Resv_Confirm</td>
<td>IP address of client who requests confirmation for its reservation</td>
</tr>
</tbody>
</table>

### Object-Order of PATH Messages

**Prescribed order**

- Common Header
  - (Integrity)
- Session
- RSVP_Hop
- Time_Values
  - (Policy_Data)
- Sender_Template
- Sender_Tspec
  - (Adspec)

**Recommended object order in PATH messages following RFC 2205**
Object-Order of RESV Messages

Recommended object order in RESV messages following RFC 2205

RSVP Usage

- RSVP provides the highest level of IP QoS but is the most complex solution
  - only suitable for private and corporate networks
- Each additional data flow requires an additional RSVP session
  - to be handled by the RSVP process inside the router; routing performance decreases!
RSVP Benefits and Drawbacks

- **Benefits**
  + Explicit resource admission control (end to end)
  + Per-request policy admission control (authorization object, policy object)
  + Signaling of dynamic port numbers

- **Drawbacks**
  - Continuous signaling due to stateless architecture
  - Not scalable

Agenda

- Integrated Services Principles
- Resource Reservation Protocol
- RSVP Message Formats
- RSVP in an IP Multicasting Environment
RSVP Procedure 1

- **RSVP sender-host**
  - transmits PATH downstream along the unicast / multicast routes provided by the routing protocol
    - PATH contains
      - RSVP sender IP address, UDP/TCP sender port
      - traffic specification (Tspec)
    - PATH is sent with same source and destination IP addresses as the data
  - **PATH stores path state in each node along the way**
    - state includes unicast IP address of previous hop node
    - router learns upstream RSVP neighbor for each sender
    - states are used later to carry RESV hop-by-hop in the reverse direction

PATH Message Downstream

Sender-host (multicast A) sends PATH message downstream. Each node along the path stores path state, including the unicast IP address of the previous hop node. Receiver-hosts learn upstream RSVP neighbors and store path states. The reverse path for RESV forwarding is established, allowing RESV messages to be sent hop-by-hop in the reverse direction.
RSVP Procedure 2

- **RSVP receiver-host**
  - sends RESV upstream towards the sender following the reverse of the path the data packet will use

- **if reservation is rejected by a node**
  - RESV_ERR is sent to the receiver-host by the corresponding node

- **if reservation can be accepted**
  - RESV is forwarded upstream
  - a reservation state is stored in the node

- **RESV is finally delivered to the sender-host itself**

---

**RSV Message Upstream**

[Diagram illustrating the RSVP message upstream process]
RSVP Procedure 3

- **In case of multicasting**
  - RESV messages are carried along the reverse path to the data source but only as far as the node where the receivers data path joins the current multicast reservation tree
    - Merge procedure

- **QoS is implemented by traffic control**
  - Packet classifier
    - E.g. based on session ID, filter specification
  - Admission control
    - E.g. based on available resources
  - Packet scheduler
    - E.g. based on weighted fair queuing

---

**RESV Message Upstream - Merge**

sender-host (multicast A)

reservation states established

receiver-host (multicast A)

receiver-host (multicast A)

receiver-host (multicast A)

receiver-host (multicast A)

receiver-host (multicast A)

reservation states established

receiver-host (multicast A)

receiver-host (multicast A)

receiver-host (multicast A)

reservation states established

receiver-host (multicast A)
RESV Message Upstream - Merge

- Path states and reservation states
  - are refreshed periodically by corresponding PATH and RESV messages
  - are deleted if no matching refresh messages arrive before expiration of a cleanup timeout (soft state)
- When a route changes
  - next PATH will initialize path state on the new route
  - next RESV messages will establish reservation state on the new route
  - it's possible that a receiver-host - which was granted QoS for a session before - will be refused on the new route caused by lack of network resources
  - rerouting will offer at least best effort service
Topology Change

- **sender-host (multicast A)**
- **receiver-host (multicast A)**
- **path and reservation states expire downstream**
- **receiver-host losses session**

New PATH Message Downstream

- **sender-host (multicast A)**
- **receiver-host (multicast A)**
- **new PATH message downstream**
- **path states established**
- **receiver-host losses session**
- **reverse path for RESV forwarding**
- **receiver-host (multicast A)**
New RESV Message Upstream

sender-host (multicast A)

reservation states established

receiver-host (multicast A)

New RESV Message Upstream - Merge

sender-host (multicast A)

reservation states established

receiver-host (multicast A)
RSVP Procedure 5

- **RSVP sender host**
  - Can tear down a session by PATH_TEAR
    - PATH_TEAR travels downstream towards all receivers
    - All corresponding path states as well as reservation states are cleared

- **RSVP receiver host**
  - Can tear down a session by RESV_TEAR
    - RESV_TEAR travels upstream towards the sender
    - All corresponding reservation states are cleared

RESV_TEAR Message

[Diagram showing the process of tear down a session with RSVP sender and receiver hosts and RESV_TEAR message]
**Path_TEAR Message**

Sender-host (multicast A) sends a PATH_TEAR message to remove path states. The receiver-hosts (multicast A) are informed about the path states being removed.

**RSVP Problems**

- **Routing difficulties**
  - path reserved over a long route, but data follow a shorter route
  - self healing by refreshing, but there could be problems with stability

- **QoS based routing**
  - no resources on shortest path but available resources on longer path

- **Transition through areas not supporting RSVP**
  - hopefully: over-provisioning of network resources

- **Scalability**
  - number of soft-states (remedy: flow aggregation)