Network Working Group
Request for Comments: 4094
Category: Informational

J. Manner
X. Fu

May 2005

Analysis of Existing Quality-of-Service Signaling Protocols

Status of This Memo

Copyright Notice

Abstract

This document reviews some of the existing Quality of Service (QoS) signaling protocols for an IP network. The goal here is to learn from them and to avoid common misconceptions. Further, we need to avoid mistakes during the design and implementation of any new protocol in this area.

Table of Contents

1. Introduction

This document reviews some of the existing QoS signaling protocols for an IP network. The goal here is to learn from them and to avoid common misconceptions. Further, we need to avoid mistakes during the design and implementation of any new protocol in this area.

There have been a number of historic attempts to deliver QoS or generic signaling to the Internet. In the early years, it was believed that multicast would be popular for the majority of communications; thus, both RSVP and earlier ST-II were designed in a way that is multicast-oriented.

ST-II was developed as a reservation protocol for point-to-multipoint communication. However, since it is sender-initiated, it does not scale with the number of receivers in a multicast group. Its processing is fairly complex. Since every sender needs to set up its own reservation, the total amount of reservation states is large. RSVP was then designed to provide support for multipoint-to-multipoint reservation setup in a more efficient way. However, its complexity, scalability, and ability to meet new requirements have been criticized.

YESSIR (YEt another Sender Session Internet Reservations) [PS98] and Boomerang [FNM+99] are examples of protocols designed after RSVP. Both were meant to be simpler than RSVP. YESSIR is an extension to RTCP, whereas Boomerang is used with ICMP.

Previously, a lot of work has been targeted at creating a new signaling protocol for resource control. Istvan Cselenyi suggested having a QoSSIG BOF in IETF47, for identifying problems in QoS signaling, but failed to get enough support [URL1]. Some people argued, "in many ways, RSVP improved upon ST-2, and it did start out simpler, but it resulted in a design with complexity and scalability", while others thought that "new knowledge and requirements" made RSVP insufficient. Some concluded that there is no simpler way to handle the same problem than RSVP.

Michael Welzl organized a special session "ABR to the Internet" in SCI 2001, and gathered some inputs for requesting an "ABR to the Internet" BOF in IETF#51, which was intended to introduce explicit rate-feedback-related mechanisms for the Internet (e2e, edge2edge). This failed because of "missing community interest".

OPENSIG [URL2] has been involved in the Internet signaling for years. Ping Pan initiated a SIGLITE [URL3] BOF mailing list to investigate lightweight Internet signaling. Finally, NSIS BOF was successful, and the NSIS WG was formed.

The most mature and original protocols are presented in their own sections, and other QoS signaling protocols are presented in later subsections. The presented protocols are chosen based on relevance to the work within NSIS. The aim is not to review every existing protocol.

2. RSVP and RSVP Extensions

RSVP (the Resource Reservation Protocol) [ZDSZ93] [RFC2205] [BEBH96] has evolved from ST-II to provide end-to-end QoS signaling services for application data streams. Hosts use RSVP to request a specific quality of service (QoS) from the network for particular application flows. Routers use RSVP to deliver QoS requests to all routers along the data path. RSVP also can maintain and refresh states for a requested QoS application flow.

By original design, RSVP fits well into the framework of the Integrated Services (IntServ) [RFC2210] [BEBH96] with certain modularity and scalability.

RSVP carries QoS signaling messages through the network, visiting each node along the data path. To make a resource reservation at a node, the RSVP module communicates with two local decision modules, admission control and policy control. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control provides authorization for the QoS request. If either check fails, the RSVP module returns an error notification to the application process that originated the request. If both checks succeed, the RSVP module sets parameters in a packet classifier and packet scheduler to obtain the desired QoS.

2.1. Basic Design

The design of RSVP distinguished itself by a number of fundamental ways; particularly, soft state management, two-pass signaling message exchanges, receiver-based resource reservation, and separation of QoS signaling from routing.

2.1.1. Signaling Model

The RSVP signaling model is based on a special handling of multicast. The sender of a multicast flow advertises the traffic characteristics periodically to the receivers via "Path" messages. Upon receipt of an advertisement, a receiver may generate a "Resv" message to reserve resources along the flow path from the sender. Receiver reservations may be heterogeneous. To accommodate the multipoint-to-multipoint multicast applications, RSVP was designed to support a vector of reservation attributes called the "style". A style describes whether

all senders of a multicast group share a single reservation and which receiver is applied. The "Scope" object additionally provides the explicit list of senders.

2.1.2. Soft State

Because the number of receivers in a multicast flow is likely to change, and the flow of delivery paths might change during the life of an application flow, RSVP takes a soft-state approach in its design, creating and removing the protocol states (Path and Resv states) in routers and hosts incrementally over time. RSVP sends periodic refresh messages (Path and Resv) to maintain its states and to recover from occasional lost messages. In the absence of refresh messages, the RSVP states automatically time out and are deleted. States may also be deleted explicitly by PathTear, PathErr with Path_State_Removed flag, or ResvTear Message.

2.1.3. Two-Pass Signaling Message Exchanges

The receiver in an application flow is responsible for requesting the desired QoS from the sender. To do this, the receiver issues an RSVP QoS request on behalf of the local application. The request propagates to all routers in reverse direction of the data paths toward the sender. In this process, RSVP requests might be merged, resulting in a protocol that scales well when there are a large number of receivers.

2.1.4. Receiver-Based Resource Reservation

Receiver-initiation is critical for RSVP to set up multicast sessions with a large number of heterogeneous receivers. A receiver initiates a reservation request at a leaf of the multicast distribution tree, traveling toward the sender. Whenever a reservation is found to already exist in a node in the distribution tree, the new request will be merged with the existing reservation. This could result in fewer signaling operations for the RSVP nodes in the multicast tree close to the sender but could introduce a restriction to receiver-initiation.

2.1.5. Separation of QoS Signaling from Routing

RSVP messages follow normal IP routing. RSVP is not a routing protocol, but rather is designed to operate with current and future unicast and multicast routing protocols. The routing protocols are responsible for choosing the routes to use to forward packets, and RSVP consults local routing tables to obtain routes. RSVP is responsible only for reservation setup along a data path.

A number of messages and objects have been defined for the protocol. A detailed description is given in [RFC2205].

2.2. RSVP Extensions

RSVP [RFC2205] was originally designed to support real-time applications over the Internet. Over the past several years, the demand for multicast-capable real-time teleconferencing, which many people had envisioned to be one of the key Internet applications that could benefit from network-wide deployment of RSVP, has never materialized. Instead, RSVP-TE [RFC3209], a RSVP extension for traffic engineering, has been widely deployed by a large number of network providers to support MPLS applications.

There are a large number of protocol extensions based on RSVP. Some provide additional features, such as security and scalability, to the original protocol. Some introduce additional interfaces to other services, such as DiffServ. And some simply define new applications, such as MPLS and GMPLS, that are completely irrelevant from protocol's original intent.

In this section, we list only IETF-based RFCs and a limited set of other organizations' specifications. Informational RFCs (e.g., RFC2998 [RFC2998]) and work-in-progress I-Ds (e.g., proxy) are not covered here.

2.2.1. Simple Tunneling

[RFC2746] describes an IP tunneling enhancement mechanism that allows RSVP to make reservations across all IP-in-IP tunnels, basically by recursively applying RSVP over the tunnel portion of the path.

2.2.2. IPsec Interface

RSVP can support IPsec on a per-address, per-protocol basis instead of on a per flow basis. [RFC2207] extends RSVP by using the IPsec Security Parameter Index (SPI) in place of the UDP/TCP-like ports. This introduces a new FILTER_SPEC object, which contains the IPsec SPI, and a new SESSION object.

2.2.3. Policy Interface

[RFC2750] specifies the format of POLICY_DATA objects and RSVP's handling of policy events. It introduces objects that are interpreted only by policy-aware nodes (PEPs) that interact with policy decision points (PDPs). Nodes that are unable to interpret the POLICY_DATA objects are called policy-ignorant nodes (PINs). The

content of the POLICY_DATA object itself is protected only between PEPs and therefore provides end-to-middle or middle-to-middle security.

[RFC2749] specifies the usage of COPS policy services in RSVP environments. [RFC3181] specifies a preemption priority policy element (PREEMPTION_PRI) for use by RSVP POLICY_DATA Object. [RFC3520] describes how authorization provided by a separate protocol (such as SIP) can be reused with the help of an authorization token within RSVP. The token might therefore contain either the authorized information itself (e.g., QoS parameters) or a reference to those values. The token might be unprotected (which is strongly discouraged) or protected based on symmetric or asymmetric cryptography. Moreover, the document describes how to provide the host with encoded session authorization information as a POLICY_DATA object.

2.2.4. Refresh Reduction

[RFC2961] describes mechanisms to reduce processing overhead requirements of refresh messages, eliminate the state synchronization latency incurred when an RSVP message is lost, and refresh state without the transmission of whole refresh messages. It defines the following objects: MESSAGE_ID, MESSAGE_ID_ACK, MESSAGE_ID_NACK, MESSAGE_ID LIST, MESSAGE_ID SRC_LIST, and MESSAGE_ID MCAST_LIST objects. Three new RSVP message types are defined:

1) Bundle messages consist of a bundle header followed by a body consisting one or more standard RSVP messages. Bundle messages help in scaling RSVP to reduce processing overhead and bandwidth consumption.

2) ACK messages carry one or more MESSAGE_ID_ACK or MESSAGE_ID_NACK objects. ACK messages are sent between neighboring RSVP nodes to detect message loss and to support reliable RSVP message delivery on a per-hop basis.

3) Srefresh messages carry one or more MESSAGE_ID LIST, MESSAGE_ID SRC_LIST, and MESSAGE_ID MCAST_LIST objects. They correspond to Path and Resv messages that establish the states. Srefresh messages are used to refresh RSVP states without transmitting standard Path or Resv messages.

2.2.5. RSVP over RSVP

[RFC3175] allows installation of one or more aggregated reservations in an aggregation region; thus, the number of individual RSVP sessions can be reduced. The protocol type is swapped from RSVP to RSVP-E2E-IGNORE in E2E (standard) Path, PathTear, and ResvConf messages when they enter the aggregation region, and is swapped back when they leave. In addition to a new PathErr code (NEW_AGGREGATE_NEEDED), three new objects are introduced:

1) SESSION object, which contains two values: the IP Address of the aggregate session destination, and the Differentiated Services Code Point (DSCP) that it will use on the E2E data the reservation contains.

2) SENDER_TEMPLATE object, which identifies the aggregating router for the aggregate reservation.

3) FILTER_SPEC object, which identifies the aggregating router for the aggregate reservation, and is syntactically identical to the SENDER_TEMPLATE object.

From the perspective of RSVP signaling and the handling of data packets in the aggregation region, these cases are equivalent to that of aggregating E2E RSVP reservations. The only difference is that E2E RSVP signaling does not take place and cannot therefore be used as a trigger, so some additional knowledge is required for setting up the aggregate reservation.

2.2.6. IEEE 802-Style LAN Interface

[RFC2814] introduces an RSVP LAN_NHOP address object that keeps track of the next L3 hop as the PATH message traverses an L2 domain between two L3 entities (RSVP PHOP and NHOP nodes). Both layer-2 and layer-3 addresses are included in the LAN_NHOP; the RSVP_HOP_L2 object is used to include the Layer-2 address (L2ADDR) of the previous hop, complementing the L3 address information included in the RSVP_HOP object (RSVP_HOP_L3 address).

To provide sufficient information for debugging or resource management, RSVP diagnostic messages (DREQ and DREP) are defined in [RFC2745] to collect and report RSVP state information along the path from a receiver to a specific sender.

2.2.7. ATM Interface

[RFC2379] and [RFC2380] define RSVP over ATM implementation guidelines and requirements to interwork with the ATM (Forum) UNI 3.x/4.0. [RFC2380] states that the RSVP (control) messages and RSVP associated data packets must not be sent on the same virtual circuits (VCs), and that an explicit release of RSVP associated QoS VCs must be performed once the VC for forwarding RSVP control messages terminates. Although a separate control VC is also possible for forwarding RSVP control messages, [RFC2379] recommends creating a best-effort short-cut first (if one does not exist), which can allow setting up RSVP-triggered VCs to use the best-effort end-point. (A short-cut is a point-to-point VC where the two end-points are located in different IP subnets.) For data flows, the subnet senders must establish all QoS VCs, and the RSVP-enabled subnet receiver must be able to accept incoming QoS VCs. RSVP must request that the configurable inactivity timers of VCs be set to "infinite". If it is too complex to do this at the VC receiver, RSVP over ATM implementations are required not to use an inactivity timer to clear any received connections. For dynamic QoS, the replacement of VC should be done gracefully.

2.2.8. DiffServ Interface

RFC2996 [RFC2996] introduces a DCLASS Object to carry Differentiated Services Code Points (DSCPs) in RSVP message objects. If the network element determines that the RSVP request is admissible to the DiffServ network, one or more DSCPs corresponding to the behavior aggregate are determined, and will be carried by the DCLASS Object added to the RESV message upstream toward the RSVP sender.

2.2.9. Null Service Type

For some applications, service parameters are specified by the network, not by the application; e.g., enterprise resource planning (ERP) applications. The Null Service [RFC2997] allows applications to identify themselves to network QoS policy agents using RSVP signaling, but does not require them to specify resource requirements. QoS policy agents in the network respond by applying QoS policies appropriate for the application (as determined by the network administrator). The RSVP sender offers the new service type, 'Null Service Type', in the ADSPEC that is included with the PATH message. A new TSPEC corresponding to the new service type is added to the SENDER_TSPEC. In addition, the RSVP sender will typically include with the PATH message policy objects identifying the user, application and sub-flow, which will be used for network nodes to manage the correspondent traffic flow.

2.2.10. MPLS Traffic Engineering

RSVP-TE [RFC3209] specifies the core extensions to RSVP for establishing constraint-based explicitly routed LSPs in MPLS networks using RSVP as a signaling protocol. RSVP-TE is intended for use by label switching routers (as well as hosts) to establish and maintain LSP-tunnels and to reserve network resources for such LSP-tunnels.

RFC3209 defines a new Hello message (for rapid node failure detection).

RFC3209 also defines new C-Types (LSP_TUNNEL_IPv4 and LSP_TUNNEL_IPv6) for the SESSION, SENDER_TEMPLATE, and FILTER_SPEC objects. Here, a session is the association of LSPs that support the LSP-tunnel. The traffic on an LSP can be classified as the set of packets that are assigned the same MPLS label value at the originating node of an LSP-tunnel.

The following 5 new objects are also defined:

1) EXPLICIT_ROUTE object (ERO), which is incorporated into RSVP Path messages, encapsulating a concatenation of hops that constitutes the explicitly routed path. Using this object, the paths taken by label-switched RSVP-MPLS flows can be pre-determined independently of conventional IP routing.

2) LABEL_REQUEST object. To establish an LSP tunnel, the sender can create a Path message with a LABEL_REQUEST object. A node that sends a LABEL_REQUEST object MUST be ready to accept and correctly process a LABEL object in the corresponding Resv messages.

3) LABEL object. Each node that receives a Resv message containing a LABEL object uses that label for outgoing traffic associated with this LSP tunnel.

4) SESSION_ATTRIBUTE object, which can be added to Path messages to aid in session identification and diagnostics. Additional control information, such as setup and holding priorities, resource affinities, and local-protection, are also included in this object.

5) RECORD_ROUTE object (RRO). The RECORD_ROUTE object may appear in both Path and Resv messages. It is used to collect detailed path information and is useful for loop detection and for diagnostics.

Section 5 of [RFC3270] further specifies the extensions to RSVP to establish LSPs supporting DiffServ in MPLS networks, introducing a new DIFFSERV Object (applicable in the Path messages), and using pre-configured or signaled "EXP<-->PHB mapping" (e.g., [RFC3270]).

RSVP-TE provides a way to indicate an unnumbered link in its Explicit Route and Record Route Objects through [RFC3477]. This specifies the following extensions:

- An Unnumbered Interface ID Subobject, which is a subobject of the Explicit Route Object (ERO) used to specify unnumbered links.

- An LSP_TUNNEL_INTERFACE_ID Object, to allow the adjacent LSR to form or use an identifier for an unnumbered Forwarding Adjacency.

- A new subobject of the Record Route Object, used to record that the LSP path traversed an unnumbered link.

2.2.11. GMPLS and RSVP-TE

GMPLS RSVP-TE [RFC3473] is an extension of RSVP-TE. It enables the provisioning of data-paths within networks supporting a variety of switching types including packet and cell switching networks, layer two networks, TDM networks, and photonic networks.

It defines the new Notify message (for general event notification), which may contain notifications being sent, with respect to each listed LSP, both upstream and downstream. Notify messages can be used for expedited notification of failures and other events to nodes responsible for restoring failed LSPs. A Notify message is sent without the router alert option.

A number of new RSVP-TE (sub)objects are defined in GMPLS RSVP-TE for general uses of MPLS:

- a Generalized Label Request Object;

- a Generalized Label Object;

- a Suggested Label Object;

- a Label Set Object (to restrict label choice);

- an Upstream_Label object (to support bidirectional LSPs);

- a Label ERO subobject;

- IF_ID RSVP_HOP objects (IPv4 & IPv6; to identify interfaces in out-of-band signaling or in bundled links);

- IF_ID ERROR_SPEC objects (IPv4 & IPv6; to identify interfaces in out-of-band signaling or in bundled links);

- an Acceptable Label Set object (to support negotiation of label values in particular for bidirectional LSPs)

- a Notify Request object (may be inserted in a Path or Resv message to indicate where a notification of LSP failure is to be sent)

- a Restart_Cap Object (used on Hello messages to identify recovery capabilities)

- an Admin Status Object (to notify each node along the path of the status of the LSP, and to control that state).

2.2.12. GMPLS Operation at UNI and E-NNI Reference Points

The ITU-T defines network reference points that separate administrative or operational parts of the network. The reference points are designated as:

- User to Network Interfaces (UNIs) if they lie between the user or user network and the core network, or

- External Network to Network Interfaces (E-NNIs) if they lie between peer networks, network domains, or subnetworks.

GMPLS is applicable to the UNI and E-NNI without further modification, and no new messages, objects, or C-Types are required. See [OVERLAY].

2.2.13. MPLS and GMPLS Future Extensions

At the time of writing, MPLS and GMPLS are being extended by the MPLS and CCAMP Working Groups to support additional sophisticated functions. This will inevitably lead to the introduction of new C-Types for existing objects, and to the requirement for new objects (CNums). It is possible that new messages will also be introduced.

Some of the key features and functions being introduced include the following:

- Protection and restoration. Features will be developed to provide
  - end-to-end protection
  - segment protection
  - various protection schemes (1+1, 1:1, 1:n)
  - support of extra traffic on backup LSPs
- Diverse path establishment for protection and load sharing.
- Establishment of point-to-multipoint paths.
- Inter-area and inter-AS path establishment with
  - explicit path control
  - bandwidth reservation
  - path diversity
- Support for the requirements of Automatic Switched Optical Network (ASON) signaling as defined by the ITU-T, including call and connection separation.
- Crankback during LSP setup.

2.2.14. ITU-T H.323 Interface

ITU-T H.323 ([H.323]) recommends the IntServ resource reservation procedure using RSVP. The information as to whether an endpoint supports RSVP should be conveyed during the H.245 [H.245] capability exchange phase, by setting appropriate qOSMode fields. If both endpoints are RSVP-capable, when opening an H.245 logical channel, a receiver port ID should be conveyed to the sender by the openLogicalChannelAck message. Only after that can a "Path - Resv - ResvConf" process take place. The timer of waiting for ResvConf message will be set by the endpoint. If this timer expires or RSVP reservation fails at any point during an H.323 call, the action is up to the vendor. Once a ResvConf message is sent or received, the endpoints should stop reservation timers and resume with the H.323 call procedures. Only explicit release of reservations are supported in [H.323]. Before sending a closeLogicalChannel message for a stream, a sender should send a PathTear message if an RSVP session has been previous created for that stream. After receiving a closeLogicalChannel, a receiver should send a ResvTear similarly. Only the FF style is supported, even for point-to-multipoint calls.

2.2.15. 3GPP Interface

Third Generation Partnership Project (3GPP) TS 23.207 ([3GPP-TS23207]) specifies the QoS signaling procedure with policy control within the Universal Mobile Telecommunications System (UMTS) end-to-end QoS architecture. When using RSVP, the signaling source and/or destination are the User Equipments (UEs, devices that allow users access to network services) that locate in the Mobile

Originating (MO) side and the Mobile Terminating (MT) side. An RSVP signaling process can either trigger or be triggered by the (COPS) PDP Context establishment/modification process. The operation of refreshing states is not specified in [3GPP-TS23207]. If a bidirectional reservation is needed, the RSVP signaling exchange must be performed twice between the end-points. The authorization token and flow identifier(s) in a policy data object should be included in the RSVP messages sent by the UE, if the token is available in the UE. When both RSVP and Service-based Local Policy are used, the Gateway GPRS Support Node (GGSN, the access point of the network) should use the policy information to decide whether to accept and forward Path/Resv messages.

2.3. Extensions for New Deployment Scenarios

As a well-acknowledged protocol in the Internet, RSVP is expected more and more to provide a more generic service for various signaling applications. However, RSVP messages were designed in a way to support end-to-end QoS signaling optimally. To meet the increasing demand that a signaling protocol also operate in host-to-edge and edge-to-edge ways, and that it serve for some other signaling purposes in addition to end-to-end QoS signaling, RSVP needs to be made more flexible and applicable for more generic signaling.

RSVP proxies [BEGD02] extend RSVP by originating or receiving the RSVP message on behalf of the end node(s), so that applications may still benefit from reservations that are not truly end-to-end. However, there are certainly scenarios where an application would want to explicitly convey its non-QoS purposed (as well as QoS) data from a host into the network, or from an ingress node to an egress node of an administrative domain. It must do so without burdening the network with excess messaging overhead. Typical examples are an end host desiring a firewall service from its provider’s network and MPLS label setup within an MPLS domain.

RSVP requires support from network routers and user space applications. Domains not supporting RSVP are traversed transparently. Unfortunately, like other IP options, RSVP messages implemented by way of IP alert option may be dropped by some routers [FJ02]. Although applications need to be built with RSVP libraries, one article presents a mechanism that would allow any host to benefit from RSVP mechanisms without applications’ awareness [MHS02].

A somewhat similar deployment benefit can be gained from the Localized RSVP (LRSVP) [JR03] [MSK+04]. The documents present the concept of local RSVP-based reservation that alone can be used to trigger reservation within an access network. In those cases, an end-host may request QoS from its own access network without the

cooperation of a correspondent node outside the access network. This would be especially helpful when the correspondent node is unaware of RSVP. A proxy node responds to the messages sent by the end host and enables both upstream and downstream reservations. Furthermore, the scheme allows for faster reservation repairs following a handover by triggering the proxy to assist in an RSVP local repair.

Still, in end-hosts that are low in processing power and functionality, having an RSVP daemon run and take care of the signaling may introduce unnecessary overhead. One article [Kars01] proposes to create a remote API so that the daemon would in fact run on the end-host's default router and the end-host application would send its requests to that daemon.

Another potential problem lies in the limited size of signaled data due to the limitation of message size. An RSVP message must fit entirely into a single non-fragmented IP datagram. Bundle messages [RFC2961] can aggregate multiple RSVP messages within a single PDU, but they still only occupy one IP datagram (i.e., approximately 64K). If it exceeds the MTU, the datagram is fragmented by IP and reassembled at the recipient node.

2.4. Conclusion

A good signaling protocol should be transparent to the applications. RSVP has proven to be a very well designed protocol. However, it has a number of fundamental protocol design issues that require more careful re-evaluation.

The design of RSVP was originally targeted at multicast applications. The result has been that the message processing within nodes is somewhat heavy, mainly due to flow merging. Still, merging rules should not appear in the specification as they are QoS-specific.

RSVP has a comprehensive set of filtering styles, including Wildcard-Filter (WF), Fixed-Filter (FF), and Shared-Explicit (SE), and is not tied to certain QoS objects. (RSVP is not tied to IntServ Guaranteed Service/Controlled Load (GS/CL) specifications.) Objects were designed to be modular, but Xspecs (TSPEC, etc.) are more or less QoS-specific and should be more generalized; there is no clear layering/separation between the signaled data and signaling protocol.

RSVP uses a soft state mechanism to maintain states and allows each node to define its own refresh timer. The protocol is also independent of underlying routing protocols. Still, in mobile networks the movement of the mobile nodes may not properly trigger a reservation refresh for the new path, and therefore a mobile node may be left without a reservation up to the length of the refresh timer.

Furthermore, RSVP does not work properly with changing end-point identifiers; that is, if one of the IP addresses of a mobile node changes, the filters may not be able to identify the flow that had a reservation.

From the security point of view, RSVP does provide the basic building blocks for deploying the protocol in various environments to protect its messages from forgery and modification. Hop-by-hop protection is provided. However, the current RSVP security mechanism does not provide non-repudiation and protection against message deletion; the two-way peer authentication and key management procedures are still missing.

Finally, since the publication of the RSVP standard, tens of extensions have emerged that allow for much wider deployment than RSVP was originally designed for -- for instance, the Subnet Bandwidth Manager, the NULL service type, aggregation, operation over tunneling, and MPLS, as well as diagnostic messages.

Domains not supporting RSVP are traversed transparently by default. Unfortunately, like other IP options, RSVP messages implemented by way of IP alert option may be dropped by some routers. Also, the maximal size of RSVP message is limited.

The transport mechanisms, performance, security, and mobility issues are detailed in the following sections.

3. RSVP Transport Mechanism Issues

3.1. Messaging Reliability

RSVP messages are defined as a new IP protocol (that is, a new ptype in the IP header). RSVP Path messages must be delivered end-to-end. For the transit routers to intercept the Path messages, a new IP Router Alert option [RFC2113] was introduced. This design is simple to implement and efficient to run. As shown from the experiments in [PS00], with minor kernel changes IP option processing introduces very little overhead on a Free BSD box.

However, RSVP does not have a good message delivery mechanism. If a message is lost on the wire, the next re-transmit cycle by the network would be one soft-state refresh interval later. By default, a soft-state refresh interval is 30 seconds.

To overcome this problem, [PS97] introduced a staged refresh timer mechanism, which has been defined as a RSVP extension in [RFC2961]. The staged refresh timer mechanism retransmits RSVP messages until the receiving node acknowledges. It can address the reliability problem in RSVP.

However, during the mechanism's implementation, a lot of effort had to be spent on per-session timer maintenance, message retransmission (e.g., avoid message bursts), and message sequencing. In addition, we have to make an effort to try to separate the transport functions from protocol processing. For example, if a protocol extension requires a natural RSVP session time-out (such as RSVP-TE one-to-one fast-reroute [FAST-REROUTE]), we have to turn off the staged refresh timers.

3.2. Message Packing

According to RSVP [RFC2205], each RSVP message can only contain information for one session. In a network that has a reasonably large number of RSVP sessions, this constraint imposes a heavy processing burden on the routers. Many router OSes are based on UNIX. [PS00] showed that the UNIX socket I/O processing is not very sensitive to packet size. In fact, processing small packets takes almost as much CPU overhead as processing large ones. However, processing too many individual messages can easily cause congestion at socket I/O interfaces.

To overcome this problem, RFC2961 introduced the message bundling mechanism. The bundling mechanism packs multiple RSVP messages between two adjacent nodes into a single packet. In one deployed router platform, the bundling mechanism has improved the number of RSVP sessions that a router can handle from 2,000 to over 7,000.

3.3. MTU Problem

RSVP does not support message fragmentation and reassembly at protocol level. If the size of a RSVP message is larger than the link MTU, the message will be fragmented. The routers simply cannot detect and process RSVP message fragments.

There is no solution for the MTU problem. Fortunately, at places where RSVP-TE has been used, either the amount of per-session RSVP data is never too large, or the link MTU is adjustable; PPP and Frame Relay can always increase or decrease the MTU sizes. For example, on some routers, a Frame Relay interface can support a link MTU size up to 9600 bytes. Currently, the RSVP MTU problem is not a realistic concern in MPLS networks.

However, when and if RSVP is used for end-user applications, for which network security is an essential and critical concern, it is possible that the size of RSVP messages can be larger than the link MTU. Note that end-users will most likely have to deal with a small 1500-byte Ethernet MTU.

3.4. RSVP-TE vs. Signaling Protocol for RT Applications

RSVP-TE works in an environment that is different from what the original RSVP has been designed for: in MPLS networks, the RSVP sessions that are used to support Label-Switched Paths (LSPs) do not change frequently.

In fact, the network operators typically set up the MPLS LSPs so that they cannot switch too quickly. For example, the operators often regulate the Constraint-based Shortest Path First (CSPF) computation interval to prevent or delay a large volume of user traffic from shifting from one session to another during LSP path optimization. (CSPF is a routing algorithm that operates from the network edge to compute the "most" optimal routes for the LSPs.) As a result, RSVP-TE does not have to handle a large amount of "triggered" (new or modified) messages. Most of the messages are refresh messages, which can be handled by the mechanisms introduced in RFC2961. In particular, in the Summary Refresh extension [RFC2961], each RSVP refresh message can be represented as a 4-byte ID. The routers can simply exchange the IDs to refresh RSVP sessions. With the full implementation of RFC2961, MPLS routers do not have any RSVP scaling issue. On one deployed router platform, it can support over 50,000 RSVP sessions in a stable backbone network.

On the other hand, in many of the new applications for which a signaling protocol is required, the user session duration can be relatively short. The dynamics of adding/dropping user sessions could introduce a large number of "triggered" messages in the network. This can clearly introduce a substantial amount of processing overhead to the routers. This is one area where a new signaling protocol may be needed to reduce the processing complexity in the resource reservation process.

3.5. What Would Be a Better Alternative?

A good signaling protocol should be transparent to the applications. On the other hand, the design of a signaling protocol must take the intended and potential applications into consideration.

With the addition of RFC2961, RSVP-TE is sufficient to support its intended application, MPLS, within the backbone. There is no significant transport-layer problem that needs to be solved.

In the last several years, a number of new applications have emerged that are proposed to need IP signaling, beyond the traditional ones associated with quality of service and resource allocation. On-path firewall control/NAT traversal (synergistic with the midcom design of [RFC3303]) is one of these. There are far-out applications such as depositing active network code in network devices. Next-generation signaling protocols dealing with novel applications, with network security requirements, and with the MTU problems described above, will prevent the re-use of the existing RSVP transport mechanism.

If a new transport protocol is needed, the protocol must be able to handle the following:

- reliable messaging;

- message packing;

- the MTU problem;

- small triggered message volume.

4. RSVP Protocol Performance Issues

4.1. Processing Overhead

By "processing overhead" we mean the amount of processing required to handle messages belonging to a reservation session. This is the processing required in addition to the processing needed for routing an (ordinary) IP packet. The processing overhead of RSVP originates from two major issues:

1) Complexity of the protocol elements. First, RSVP itself is per-flow based; thus the number of states is proportional to RSVP session number. Path and Resv states have to be maintained in each RSVP router for each session (and Path state also has to record the reverse route for the correspondent Resv message). Second, being receiver-initiated, RSVP optimizes various merging operations for multicast reservations while the Resv message is processed. To handle multicast, other mechanisms such as reservation styles, scope object, and blockade state, are also required to be presented in the basic protocol. This not only adds sources of failures and errors, but also complicates the state machine [Fu02]. Third, the same RSVP signaling messages are used not only for maintaining the state, but also for dealing with recovery of signaling message loss and discovery of route change. Thus, although protocol elements that represent the actual data (e.g., QoS parameters) specification are separated from signaling elements, the processing overhead needed for all RSVP messages is

not marginal. Finally, the possible variations of the order and existence of objects increases the complexity of message parsing and internal message and state representation.

2) Implementation-specific Overhead. There are two ways to send and receive RSVP messages: either as "raw" IP datagrams with protocol number 46, or as encapsulated UDP datagrams, which increase the efficiency of RSVP processing. Typical RSVP implementations are user-space daemons interacting with the kernel; thus, state management, message sending, and reception would affect the efficiency of the protocol processing. For example, in the recent version of the implementation described in [KSS01], the relative execution costs for the message sending/reception system calls "sendto", "select", and "recvmsg" were 14-16%, 6-7%, 9-10%, individually, of the total execution cost. [KSS01] also found that state (memory) management can use up to 17-18% of the total execution cost, but it is possible to decrease that cost to 6-7%, if appropriate action is taken to replace the standard memory management with dedicated memory management for state information. RSVP/routing, RSVP/policy control, and RSVP/traffic control interfaces can also pose different overhead depending on implementation. For example, the RSVP/routing overhead has been measured to be approximately 11-12% of the total execution cost [KSS01].

4.2. Bandwidth Consumption

By "bandwidth consumption" we mean the amount of bandwidth used during the lifetime of a session: to set up a reservation session, to keep the session alive, and finally to close it.

RSVP messages are sent either to trigger a new reservation or to refresh an existing reservation. In standard RSVP, Path/Resv messages are used for triggering and refreshing/recovering reservations, identically, which results in an increased size of refresh messages. The hop-by-hop refreshment may reduce the bandwidth consumption for RSVP, but could result in more sources of error/failure events. [RFC2961] presents a way to bundle standard RSVP messages and reduces the refreshment redundancy by Srefresh message.

Thus, the following formula represents the bandwidth consumption in bytes for an RSVP session lasting n seconds:

$$F(n) = (bP + bR) + ((n/Ri) * (bP + bR)) + bPt$$

bP: IP payload size of Path message
bR: IP payload size of Resv message
bPt: IP payload size of Path Tear message
Ri: refresh interval

For example, for a simple Controlled Load reservation without security and identification requirements (where bP is 172 bytes, bR is 92, bPt is 44 bytes, and Ri is 30 seconds), the bandwidth consumption would be as follows:

$$F(n) = (172 + 92) + ((n/30) * (172 + 92)) + 44$$

$$= 308 + (264n/30) \text{ bytes}$$

5. RSVP Security and Mobility

5.1. Security

To allow a process on a system to securely identify the owner and the application of the communicating process (e.g., user id) and to convey this information in RSVP messages (PATH or RESV) in a secure manner, [RFC3182] specifies the encoding of identities as RSVP POLICY_DATA Object. However, to provide ironclad security protection, cryptographic authentication combined with authorization has to be provided. Such a functionality is typically offered by authentication and key exchange protocols. Solely including a user identifier is insufficient.

To provide hop-by-hop integrity and authentication of RSVP messages, an RSVP message may contain an INTEGRITY object ([RFC2747]) using a keyed message digest. Since intermediate routers need to modify and process the content of the signaling message, a hop-by-hop security architecture based on a chain-of-trust is used. However, with the different usage of RSVP as described throughout this document and with new requirements, a re-evaluation of the original assumptions might be necessary.

RFC2747 provides protection against forgery and message modification. However, this does not provide non-repudiation or protect against message deletion. In the current RSVP security scheme, the two-way peer authentication and key management procedures are still missing.

The security issues have been well analyzed in [Tsch03].

5.2. Mobility Support

Two issues raise concern when a mobile node (MN) uses RSVP: the flow identifier and reservation refresh. When an MN changes locations, it may need to change one of its assigned IP addresses. An MN may have an IP address by which it is reachable by nodes outside the access network, and an IP address used to support local mobility management. Depending on the mobility management mechanism, a handover may force a change in any of these addresses. As a consequence, the filters associated with a reservation may not identify the flow anymore, and the resource reservation is ineffective until a refresh with a new set of filters is initialized.

The second issue relates to following the movement of a mobile node. RFC2205 defines that Path messages can perform a local repair of reservation paths. When the route between the communicating end hosts changes, a Path message will set the state of the reservation on the new route, and a subsequent Resv message will make the resource reservation. Therefore, by sending a Resv message a host cannot alone update the reservation, and thus it cannot perform a local repair before a Path message has passed. Also, in order to provide fast adaptation to routing changes without the overhead of short refresh periods, the local routing protocol module can notify the RSVP process of route changes for particular destinations. The RSVP process should use this information to trigger a quick refresh of state for these destinations, using the new route (Section 3.6, [RFC2205]). However, not all local mobility protocols affect routing directly in routers (not even Mobile IP), and thus mobility may not be noticed at RSVP routers. Therefore, it may take a relatively long time before a reservation is refreshed following a handover.

There have been several designs for extensions to RSVP to allow for more seamless mobility. One solution is presented in [MSK+04], in which one section discusses the coupling of RSVP and the mobility management mechanisms and proposes small extensions to RSVP to handle the handover event better, among other things. The extension allows the mobile host to request a Path for the downstream reservation when a handover has happened.

Another example is Mobile RSVP (MRSVP) [TBA01], which is an extension to standard RSVP. It is based on advance reservations, where neighboring access points keep resources reserved for mobile nodes moving to their coverage area. When a mobile node requests resources, the neighboring access points are checked, too, and a passive reservation is done around the mobile nodes' current location.

The problem with the various "advance reservation" schemes is that they require topological information of the access network and, usually, advance knowledge of the handover event. Furthermore, the way the resources reserved in advance are used in the neighboring service areas is an open issue. A good overview of these different schemes can be found in [MA01].

The interactions of RSVP and Mobile IP have been well documented in [Thom02].

6. Other QoS Signaling Proposals

6.1. Tenet and ST-II

Tenet and ST-II are two original QoS signaling protocols for the Internet.

In the original Tenet architecture [BFM+96], the receiver sends a reservation request toward the source. Each network node along the way makes the reservation. Once the request arrives at the source, the source sends another Relax message back toward to the receiver, and has the option to modify the previous reservation at each node.

ST-II [RFC1819] basically works in the following way: a sender originates a Connect message to a set of receivers. Each intermediate node determines the next hop subnets, and makes reservations on the links going to these next hops. Upon receiving a Connect indication, a receiver must send back either an Accept or a Refuse message to the sender. In the case of an Accept, the receiver may further reduce the resource request by updating the returned flow specifications.

ST-II consists of two protocols: ST for the data transport and the Stream Control Message Protocol (SCMP) for all control functions. ST is simple and contains only a single PDU format, which is designed for fast and efficient data forwarding in order to achieve low communication delays. SCMP packets are transferred within ST packets.

ST-II has no built-in soft states; thus, it requires that the network be responsible for correctness. It is sender-initiated, and the overhead for ST-II to handle group membership dynamics is higher than that for RSVP [MESZ94]. ST-II does not provide security, but [RFC1819] describes some objects related to charging.

6.2. YESSIR

YESSIR (YEt another Sender Session Internet Reservations) [PS98] is a resource reservation protocol that seeks to simplify the process of establishing reserved flows while preserving many unique features introduced in RSVP. Simplicity is measured in terms of control message processing, data packet processing, and user-level flexibility. Features such as robustness, advertising network service availability, and resource sharing among multiple senders are also supported in the proposal.

The proposed mechanism generates reservation requests by senders to reduce the processing overhead. It is built as an extension to the Real-Time Transport Control Protocol (RTCP), taking advantage of Real-Time Protocol (RTP). YESSIR also introduces a concept called partial reservation, in which, for certain types of applications, the reservation requests can be passed to the next hop, even though there are not enough resources on a local node. The local node can rely on optimized retries to complete the reservations.

6.2.1. Reservation Functionality

YESSIR [PS98] was designed for one-way, sender-initiated end-to-end resource reservation. It also uses soft state to maintain states. It supports resource query (similar to RSVP diagnosis message), advertising (similar to RSVP ADSPEC), shared reservation, partial reservations, and flow merging.

To support multicast, YESSIR simplifies the reservation styles to individual and shared reservation styles. Individual reservations are made separately for each sender, whereas shared reservations allocate resources that can be used by all senders in an RTP session. Although RSVP supports shared reservation (SE and WF styles) from the receiver’s direction, YESSIR handles the shared reservation style from the sender’s direction; thus, new receivers can re-use the existing reservation of the previous sender.

It has been shown that the YESSIR one-pass reservation model has better performance and lower processing cost than a regular two-way signaling protocol, such as RSVP [PS98]. The bandwidth consumption of YESSIR is somewhat lower than that of, for example, RSVP, because it does not require additional IP and transport headers. Bandwidth consumption is limited to the extension header size.

YESSIR does not have any particular support for mobility, and the security of YESSIR relies on RTP/RTCP security measures.

6.2.2. Conclusion

YESSIR requires support in applications since it is an integral part of RTCP. Similarly, it requires network routers to inspect RTCP packets to identify reservation requests and refreshes. Routers unaware of YESSIR forward the RTCP packets transparently.

6.3. Boomerang

Boomerang [FNM+99] is a another resource reservation protocol for IP networks. The protocol has only one message type and a single signaling loop for reservation setup and teardown, and it has no requirements on the far end node. Instead, it concentrates the intelligence in the Initiating Node (IN).

In addition, the Boomerang protocol allows for sender- or receiver-oriented reservations and resource query. Flows are identified with the common 5-tuple, and the QoS can be specified by various means; e.g., service class and bit rate. In the initial implementation, Boomerang messages are transported in ICMP ECHO/REPLY messages.

6.3.1. Reservation Functionality

Boomerang can only be used for unicast sessions; no support for multicast exists. The requested QoS can be specified with various methods, and both ends of a communication session can make a reservation for their transmitted flow.

The authors of Boomerang show in [FNS02] that the processing of the protocol is considerably lower than that of the ISI RSVP daemon implementation. However, this is mainly due to the limited functionality provided by the protocol compared to that provided by RSVP.

Boomerang messages are quite short and consume a relatively low amount of link bandwidth. This is due to the limited functionality of the protocol; for example, no security-specific information or policy-based interaction is provided. Being sender oriented, the bandwidth consumption mostly affects the downstream direction, from the sender to the receiver.

As Boomerang is sender oriented, there is no need to store backward information. This reduces the signaling required. The rest of the issues that were identified with RSVP apply with Boomerang. No security mechanism is specified for Boomerang.

The Boomerang protocol has deployment issues similar to those of any host-network-host protocol. It requires an implementation at both communicating nodes and in routers. Boomerang-unaware routers should be able to forward Boomerang messages transparently. Still, firewalls often drop ICMP packets, making the protocol useless.

6.3.2. Conclusions

Boomerang seems to be a very lightweight protocol and efficient in its own scenarios. However, the apparent low processing overhead and bandwidth consumption results from the limited functionality. No support for multicast or any security features are present, which allows for a different functionality than RSVP, which the authors like to compare Boomerang to.

6.4. INSIGNIA

INSIGNIA [LGZC00] is proposed as a very simple signaling mechanism for supporting QoS in mobile ad-hoc networks. It avoids the need for separate signaling by carrying the QoS signaling data along with the normal data in IP packets using IP packet header options. This approach, known as "in-band signaling", is proposed as more suitable in the rapidly changing environment of mobile networks since the signaled QoS information is not tied to a particular path. It also allows the flows to be rapidly established and, thus, is suitable for short-lived and dynamic flows.

INSIGNIA aims to minimize signaling by reducing the number of parameters that are provided to the network. It assumes that real-time flows may tolerate some loss, but are very delay sensitive so that the only QoS information needed is the required minimum and maximum bandwidth.

The INSIGNIA protocol operates at the network layer and assumes that link status sensing and access schemes are provided by lower-layer entities. The usefulness of the scheme depends on the MAC layers, but this is undefined, so INSIGNIA can run over any MAC layer. The protocol requires that each router maintains per-flow state.

The INSIGNIA system implicitly supports mobility. A near-minimal amount of information is exchanged with the network. To achieve this, INSIGNIA makes many assumptions about the nature of traffic that a source will send. This may also simplify admission control and buffer allocation. The system basically assumes that "real-time" will be defined as a maximum delay, and the user can simply request real-time service for a particular quantity of traffic.

After handover, data that was transmitted to the old base station can be forwarded to the new base station, so no data loss should occur. However, there is no way to differentiate between re-routed and new traffic, so priority cannot be given to handover traffic, for example.

INSIGNIA, however, (completely) lacks a security framework and does not investigate how to secure signaled QoS data in an ad-hoc network, where relatively weak trust or even no trust exists between the participating nodes. Therefore, authorization and charging especially might be a challenge. The security protection of in-band signaling is costly since the data delivery itself experiences increased latency if security processing is done hop-by-hop. Because the QoS signaling information is encoded into the flow label and end-to-end addressing is used, it is very difficult to provide security other than IPsec in tunnel mode.

7. Inter-Domain Signaling

This section gives a short overview of protocols designed for inter-domain signaling.

7.1. BGRP

Border Gateway Reservation Protocol (BGRP) [BGRP] is a signaling protocol for inter-domain aggregated resource reservation for unicast traffic. BGRP builds a sink tree for each of the stub domains. Each sink tree aggregates bandwidth reservations from all data sources in the network. BGRP maintains these aggregated reservations using soft state and relies on Differentiated Services for data forwarding.

In terms of message processing load, BGRP scales state storage and bandwidth. Because backbone routers only maintain the sink tree information, the total number of reservations at each router scales linearly with the number of Internet domains.

7.2. SICAP

SICAP (Shared-segment Inter-domain Control Aggregation protocol) [SGV03] is an inter-domain signaling solution that performs shared-segment aggregation [SGV02] on the Autonomous System (AS) level in order to reduce state required at Boundary Routers (BRs). SICAP performs aggregation based on path segments that different reservations share. Thus, reservations may be merged into aggregates that do not necessarily extend all the way to the reservation's destination. The motivation for creating "shorter" aggregates is that, on one hand, their ability to accommodate future requests more easily, and, on the other hand, the minimization of aggregates

created and consequently, the reduction of state required to manage established reservations. However, in contrast to the sink-tree approach (used by BGRP [BGRP]), the shared-segment approach introduces intermediate de-aggregation locations. These are ASes where aggregates may experience "re-aggregation". At these locations, routers that perform aggregation (AS egress routers) have to keep track of the mapping between reservations and aggregates. One possible way to do this is to keep each reservation identifier and the corresponding resources stored at each aggregator. However, this solution incurs a high state penalty. SICAP avoids this state penalty by keeping track of the mapping between aggregates and reservations at the level of destination domains rather than explicitly map individual reservations to aggregates. In other words, SICAP maintains, per aggregate, a list of the destination prefixes advertised by the destination AS an aggregate provides access to.

Pan et al. show that BGRP scales well in terms of control state, message processing, and bandwidth efficiency, when compared to RSVP without aggregation. However, partially given that BGRP was the first approach to explore the issue of inter-domain control aggregation in detail, they did not provide a comparison with other aggregation protocols.

SICAP and BGRP messaging sequences are similar, and consequently, these protocols attain the same signaling load. This load is exactly the same as that attained by proposals that do not perform aggregation, given that SICAP and BGRP exchange messages per individual reservation. In terms of bandwidth, both protocols provision aggregates with the exact bandwidth required by their merged reservations. Therefore, the major difference between SICAP and BGRP is state maintained at BRs, which is significantly reduced by SICAP. We consider this to be of importance not so much for offering a better-performing alternative to BGRP, but for quantifying the performance improvements that might still be available in the research field of control path aggregation. Finally, to deal with the possible problem of the signaling load, SICAP uses an over-reservation mechanism [SGV03b], whose design took into consideration a possible support for BGRP.

7.3. DARIS

Dynamic Aggregation of Reservations for Internet Services (DARIS) [Bless02] [Bless04] defines an inter-domain aggregation scheme for resource reservations. Basically, it aggregates reservations along Autonomous System (AS) paths (or parts thereof). A set of reservations whose data paths share a common sequence of ASes are integrated into a joint reservation aggregate along that shared sub-

path. All entities within the aggregate, except for aggregate starting and end point, can remove state information of the included individual reservations, thereby saving states. They just need to hold the necessary information about the encompassing aggregate. Moreover, these intermediate ASes are no longer involved in signaling that is related to the aggregated reservations. If more aggregate resources are reserved than were actually required, the capacity of the aggregate does not need to be adapted with every new or released reservation (thereby reducing the number of message exchanges).

An aggregate between two ASes is created as soon as a threshold k is exceeded that describes the active number of unidirectional reservations between them. It is, however, possible to apply different aggregation triggers. Furthermore, DARIS allows aggregates to be nested hierarchically. Therefore, the existence of shorter aggregates does not prevent the creation of longer (and thus more efficient) aggregates, and vice versa. An evaluation of recent BGP routing information in [Bless02] showed that 92% of all end-to-end paths contain at least four ASes. Consequently, an aggregate from edge AS to edge AS can span four or more ASes, thus saving states and signaling message processing in at least two ASes.

There is, however, a small chance that a reservation cannot be included in a new aggregate, because it was already aggregated elsewhere. This so-called "aggregation conflict" is caused by the prior removal of state information related to individual reservations within intermediate ASes of the encompassing aggregate. This may also bring difficulties if reservations or aggregates are re-routed between ASes. One must be careful when considering how to define sophisticated adaptation techniques for these special cases, because they seem to become very complex.

The signaling protocol DMSP (Domain Manager Signaling Protocol) supports aggregation by special extensions that reduce the reservation setup time for more than one round-trip time in some cases (e.g., if an aggregate’s capacity must be increased before a new reservation can be included). Details can be found in [Bless02].

The DARIS concept was evaluated by using a simulation with a topology that was derived from real BGP routing table information and comprised more than 5500 ASes. In comparison to a non-aggregated scenario, the number of saved states lay in the range of one to two orders of magnitude, and similar results were obtained with respect to the number of signaling messages. Though [Bless02] describes DARIS in the context of distributed Domain Management entities (similar to a bandwidth broker), it can be applied in a router-based

resource management environment, too. This will achieve a higher degree of distribution, which is beneficial for large ASes, which are highly interconnected.

A general issue with aggregation is that it is not the aggregating and de-aggregating ASes that profit from their initiated aggregates, but all intermediate ASes within an aggregate. Therefore, some incentive for aggregate creation has to be given. This may lead to novel cost models that have to be developed for aggregation concepts in the future.

8. Security Considerations

This document does not present new technology or protocols. Thus, there are no explicit security issues. Still, individual protocols include different levels of security issues and those are highlighted in the relevant sections and references.

9. Summary

Supporting flow-based soft state reservations has been proven useful. Still, there have been different ways to improve the performance, including refresh reduction and aggregation. However, some of the main concerns with these signaling protocols are the complexity of the protocol, which affects implementations and processing overhead, and the security of the signaling. Especially, a proper scheme to handle authentication and authorization of QoS resource requests and a framework for providing signaling message security seem to be missing from most protocols. RSVP has a mechanism to protect signaling messages based on manually distributed keys and concepts for authorization, but they seem to be insufficient for a dynamic and mobile environment. [Tsch03] provides more details on security properties provided by RSVP. Moreover, secure and efficient signaling to and from mobile nodes has been one of the critical challenges not fully met by existing protocols.

Moving QoS signaling protocols into a generic messenger can provide much adoption. It is expected that the development of future protocols should learn from the lessons of existing ones. Nevertheless, the tradeoffs between the expected functionality, protocol complexity/performance would still be taken into account. For example, RSVP uses the two-way signaling mechanism, whereas YESSIR employs only one-pass signaling. Both can be shown to out-perform the other in specific carefully chosen signaling scenarios.

10. Contributors

This document is part of the work done in the NSIS Working Group. The document was initially written by Jukka Manner and Xiaoming Fu. Since the first version, Martin Karsten has provided text about the processing overhead of RSVP, and Hannes Tschofenig has provided text about various security issues in the protocols. Henning Schulzrinne and Ping Pan have provided more information on RSVP transportation after the second revision. Kireeti Kompella and Adrian Farrel provided a review and updates to the discussion on RSVP-TE and GMPLS.

11. Acknowledgements

We would like to acknowledge Bob Braden and Vlora Rexhepi for their useful comments.

12. Appendix A: Comparison of RSVP to the NSIS Requirements

This section provides a comparison of RSVP to the requirements identified as part of the work in NSIS [RFC3726]. The numbering follows the division in the requirements document.

5.1. Architecture and Design Goals

5.1.1. NSIS SHOULD Provide Availability Information on Request

RSVP itself does not support query-type of operations. However, the RSVP diagnosis messages extension [RFC2745] provides a means to query resource availability.

5.1.2. NSIS MUST Be Designed Modularly

RSVP was designed to be modular by way of TLV objects, however it is regarded being lack of sufficient extensibility in various kind of signalling applications.

5.1.3. NSIS MUST Decouple Protocol and Information

RSVP is decoupled from the IntServ QoS specifications. Still, the concept of sessions in RSVP are somewhat coupled to the information it carries.

5.1.4. NSIS MUST Support Independence of Signaling and Network Control Paradigm

The IntServ information carried by RSVP does not tie the QoS provisioning mechanisms.

5.1.5. NSIS SHOULD Be Able To Carry Opaque Objects

RSVP supports this.

5.2. Signaling Flows

5.2.1. The Placement of NSIS Initiator, Forwarder, and Responder Anywhere in the Network MUST Be Allowed

Standard RSVP works only end-to-end, although the RSVP proxy [BEGD02] and the Localized RSVP [MSK+04] have relaxed this assumption. RSVP relies on receiver-initiation way to perform QoS reservations.

5.2.2. NSIS MUST support Path-Coupled and MAY Support Path-Decoupled Signaling

Standard RSVP is path-coupled, but the Subnet Bandwidth Manager (SBM) work makes RSVP somewhat path-decoupled.

5.2.3. Concealment of Topology and Technology Information SHOULD Be Possible

RSVP itself does not provide such capability.

5.2.4. Transparent Signaling through Networks SHOULD Be Possible

RSVP messages are intercepted and evaluated in each RSVP router, and thus they may not cross certain RSVP-routers unnoticed. Still, the message processing rules allow unknown RSVP messages to be forwarded unharmed.

5.3. Messaging

5.3.1. Explicit Erasure of State MUST Be Possible

Supported by the PathTear and ResvTear messages.

5.3.2. Automatic Release of State After Failure MUST Be Possible

On error reservation states are torn down with PathTear messages.

5.3.3. NSIS SHOULD Allow for Sending Notifications Upstream

There are two notifications in RSVP, confirm of a reservation set-up and tear down of reservation states as a result of errors.

5.3.4. Establishment and Refusal To Set Up State MUST Be Notified

PathErr and ResvErr messages provide refusal to set up state in RSVP.

5.3.5. NSIS MUST Allow for Local Information Exchange

RSVP NULL service type [RFC2997] provides such a feature.

5.4. Control Information

5.4.1. Mutability Information on Parameters SHOULD Be Possible

Rspec and Adspec are mutable; Tspec is (generally) end-to-end not mutable.

5.4.2. It SHOULD Be Possible To Add and Remove Local Domain Information

RSVP aggregation [RFC3175] and NULL service type [RFC2997] can provide such a feature.

5.4.3. State MUST Be Addressed Independent of Flow Identification

RSVP states are tied to the flows, thus this requirement is not met.

5.4.4. Modification of Already Established State SHOULD Be Seamless

Modifications of a reservation is possible with RSVP.

5.4.5. Grouping of Signaling for Several Micro-Flows MAY Be Provided

Aggregated RSVP and RFC2961 allow this.

5.5. Performance

5.5.1. Scalability

RSVP scales linearly to the number of reservation states.

5.5.2. NSIS SHOULD Allow for Low Latency in Setup

Setting up an RSVP reservation takes one round-trip time and the processing times are each RSVP router.

5.5.3. NSIS MUST Allow for Low Bandwidth Consumption for the Signaling Protocol

The initial reservations messages can not be compressed, but the refresh interval can be adjusted to consume less bandwidth, at the expense of possible inefficient resource usage.

5.5.4. NSIS SHOULD Allow To Constrain Load on Devices

See discussions on RSVP performance (section 4).

5.5.5. NSIS SHOULD Target the Highest Possible Network Utilization

This depends on the IntServ service types, Controlled Load can provide better overall utilization than Guaranteed Service.

5.6. Flexibility

5.6.1. Flow Aggregation

Aggregated RSVP and RFC2961 allow this.

5.6.2. Flexibility in the Placement of the NSIS Initiator/Responder

RSVP allows receiver as initiator of reservations.

5.6.3. Flexibility in the Initiation of State Change

RSVP receivers can initiate the state change during its refreshment.

5.6.4. SHOULD Support Network-Initiated State Change

As RSVP supports hop-by-hop refreshment, this is made possible.

5.6.5. Uni / Bi-Directional State Setup

RSVP is only uni-directional.

5.7. Security

5.7.1. Authentication of Signaling Requests

Authentication is available in RSVP.

5.7.2. Request Authorization

Authorization with a PDP is possible in RSVP.

5.7.3. Integrity Protection

The INTEGRITY Object is available in RSVP.

5.7.4. Replay Protection

The INTEGRITY Object to replay protect the content of the signaling messages between two RSVP nodes.

5.7.5. Hop-By-Hop Security

The RSVP security model works only hop-by-hop.

5.7.6. Identity Confidentiality and Network Topology Hiding

The INTEGRITY Object can be used for this purpose.

5.7.7. Denial-Of-Service Attacks

Challenging with RSVP.

5.7.8. Confidentiality of Signaling Messages

Not supported by RSVP.

5.7.9. Ownership of State

Challenging with RSVP.

5.8. Mobility

5.8.1. Allow Efficient Service Re-Establishment After Handover

Works for upstream but may not be achieved for the downstream if mobility is not noticed at the cross-over router.

5.9. Interworking with Other Protocols and Techniques

5.9.1. MUST Interwork with IP Tunneling

RFC 2746 discusses these issues.

5.9.2. MUST NOT Constrain either to IPv4 or IPv6

RSVP supports both IP versions.

5.9.3. MUST Be Independent from Charging Model

RSVP does not discuss this.

5.9.4. SHOULD Provide Hooks for AAA Protocols

COPS and RSVP work together.

5.9.5. SHOULD Work with Seamless Handoff Protocols

Not supported by RSVP. Still, [RFC2205] suggests that route changes should be indicated to the local RSVP daemon, which can then initiate state refresh.

5.9.6. MUST Work with Traditional Routing

RSVP expects traditional routing.

5.10. Operational

5.10.1. Ability to Assign Transport Quality to Signaling Messages

This is a network design issue, but is possible with DiffServ.

5.10.2. Graceful Fail Over

RSVP supports this.

5.10.3. Graceful Handling of NSIS Entity Problems

RSVP itself does not supports this.

13. Normative References

[RFC3726] Brunner, M., "Requirements for Signaling Protocols", RFC 3726, April 2004.

14. Informative References

[3GPP-TS23207] 3GPP TS 23.207 V5.6.0, End-to-end Quality of Service (QoS) Concept and Architecture, Release 5, December 2002.

[BEBH96] Braden, R., Estrin, D., Berson, S., Herzog, and D. Zappala, "The Design of the RSVP Protocol", ISI Final Technical Report, July 1996.

[BEGD02] Y. Bernet, N. Elfassy, S. Gai, and D. Dutt, "RSVP Proxy", Work in Progress, March 2002.

[BFM+96] A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma, and H. Zhang, "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences", IEEE/ACM Transactions on Networking, Volume 4, Issue 1, February 1996, pp. 1-10.

[BGRP] P. Pan, E, Hahne, and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations", Journal of Communications and Networks, Vol. 2, No. 2, June 2000, pp. 157-167.

[Bless02] R. Bless, "Dynamic Aggregation of Reservations for Internet Services", Proceedings of the Tenth International Conference on Telecommunication Systems - Modeling and Analysis (ICTSM 10), Vol. 1, pp. 26-38, October 3-6 2002, Monterey, California, available at http://www.tm.uka.de/doc/2003/ictsm-daris-journal-crc-web.pdf.

[Bless04] R. Bless, "Towards Scalable Management of QoS-based End-to- End Services" (PDF), Proceedings of NOMS 2004 (IEEE/IFIP 2004 Network Operations and Management Symposium), April 2004, Seoul, Korea.

[FAST-REROUTE] P. Pan, G. Swallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", Work in Progress, January 2004.

[FNM+99] G. Feher, K. Nemeth, M. Maliosz, I. Cselenyi, J. Bergkvist, D. Ahlard, T. Engborg, "Boomerang A Simple Protocol for Resource Reservation in IP Networks", IEEE RTAS, 1999.

[FNS02] G. Feher, K. Nemeth, and I. Cselenyi, "Performance evaluation framework for IP resource reservation signalling". Performance Evaluation 48 (2002), pp. 131-156.

[FJ02] P. Fransson and A. Jonsson, "The need for an alternative to IPv4-options", in RVK (RadioVetenskap och Kommunikation), Stockholm, Sweden, pp. 162-166, June 2002.

[Fu02] X. Fu, C. Kappler, and H. Tschofenig, "Analysis on RSVP Regarding Multicast". Technical Report No. IFI-TB-2002-001, ISSN 1611-1044, Institute for Informatics, University of Goettingen, Oct 2002.

[H.245] ITU-T Recommendation H.245, Control Protocol for Multimedia Communication, July 2000.

[H.323] ITU-T Recommendation H.323, Packet-based Multimedia Communications Systems, Nov. 2000.

[JR03] Jukka Manner, Kimmo Raatikainen, "Localized QoS Management for Multimedia Applications in Wireless Access Networks". IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA 2003), August, 2003, pp. 193-200.

[Kars01] M. Karsten, "Experimental Extensions to RSVP -- Remote Client and One-Pass Signalling". IWQoS 2001, Karlsruhe, Germany, June 2001.

[KSS01] M. Karsten, Jens Schmitt, Ralf Steinmetz, "Implementation and Evaluation of the KOM RSVP Engine", IEEE Infocom 2001.

[LGZC00] S. Lee, A. Gahng-Seop, X. Zhang, A. Campbell,"INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks". Journal of Parallel and Distributed Computing (Academic Press), Special issue on Wireless and Mobile Computing and Communications, Vol. 60, Number 4, April, 2000, pp. 374-406.

[MA01] B. Moon, and H. Aghvami, "RSVP Extensions for Real-Time Services in Wireless Mobile Networks". IEEE Communications Magazine, December 2001, pp. 52-59.

[MESZ94] D. Mitzel, D. Estrin, S. Shenker, and L. Zhang, "An Architectural Comparison of ST-II and RSVP", Infocom 1994.

[MHS02] Y Miao, W. Hwang, and C. Shieh, "A transparent deployment method of RSVP-aware applications on UNIX". Computer Networks, 40 (2002), pp. 45-56.

[MSK+04] J. Manner, T. Suihko, M. Kojo, M. Liljeberg, K. Raatikainen, "Localized RSVP", Work in Progress, September 2004.

[OVERLAY] G. Swallow, J. Drake, H. Ishimatsu, and Y. Rekhter, "GMPLS UNI: RSVP Support for the Overlay Model", Work in Progress, February 2004.

[PS97] P. Pan and H. Schulzrinne, "Staged refresh timers for RSVP", Global Internet, Phoenix, Arizona, November 1997.

[PS98] P. Pan, and H. Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet". Proceedings of NOSSDAV, Cambridge, UK, July 1998.

[PS00] P. Pan, and H. Schulzrinne, "PF_IPOPTION: A kernel extension for IP option packet processing", Technical Memorandum 10009669-02TM, Bell Labs, Lucent Technologies, Murray Hill, NJ, June 2000.

[RFC1819] Delgrossi, L. and L. Berger, "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, August 1995.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2207] Berger, L. and T. O'Malley, "RSVP Extensions for IPSEC Data Flows", RFC 2207, September 1997.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.

[RFC2379] Berger, L., "RSVP over ATM Implementation Guidelines", BCP 24, RFC 2379, August 1998.

[RFC2380] Berger, L., "RSVP over ATM Implementation Requirements", RFC 2380, August 1998.

[RFC2745] Terzis, A., Braden, B., Vincent, S., and L. Zhang, "RSVP Diagnostic Messages", RFC 2745, January 2000.

[RFC2746] Terzis, A., Krawczyk, J., Wroclawski, J., and L. Zhang, "RSVP Operation Over IP Tunnels", RFC 2746, January 2000.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC2749] Herzog, S., Boyle, J., Cohen, R., Durham, D., Rajan, R., and A. Sastry, "COPS usage for RSVP", RFC 2749, January 2000.

[RFC2750] Herzog, S., "RSVP Extensions for Policy Control", RFC 2750, January 2000.

[RFC2814] Yavatkar, R., Hoffman, D., Bernet, Y., Baker, F., and M. Speer, "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks", RFC 2814, May 2000.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[RFC2996] Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996, November 2000.

[RFC2997] Bernet, Y., Smith, A., and B. Davie, "Specification of the Null Service Type", RFC 2997, November 2000.

[RFC2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000.

[RFC3175] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.

[RFC3181] Herzog, S., "Signaled Preemption Priority Policy Element", RFC 3181, October 2001

[RFC3182] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S., and R. Hess, "Identity Representation for RSVP", RFC 3182, October 2001.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

[RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[RFC3520] Hamer, L-N., Gage, B., Kosinski, B., and H. Shieh, "Session Authorization Policy Element", RFC 3520, April 2003.

[SGV02] R. Sofia, R. Guerin, and P. Veiga, "An Investigation of Inter-Domain Control Aggregation Procedures", International Conference on Networking Protocols, ICNP 2002, Paris, France, November 2002.

[SGV03] R. Sofia, R. Guerin, and P. Veiga. SICAP, a Shared-segment Inter-domain Control Aggregation Protocol. High Performance Switching and Routing, HPSR 2003, Turin, Italy, June 2003.

[SGV03b] R. Sofia, R. Guerin, and P. Veiga. A Study of Over-reservation for Inter-Domain Control Aggregation Protocols. Technical report (short version under submission), University of Pennsylvania, May 2003, available at http://einstein.seas.upenn.edu/mnlab/publications.html.

[TBA01] A. Talukdar, B. Badrinath, and A. Acharya, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts", Wireless Networks, vol. 7, no. 1, pp. 5-19, 2001.

[Thom02] M. Thomas, "Analysis of Mobile IP and RSVP Interactions", Work in Progress, October 2002.

[Tsch03] H. Tschofenig, "RSVP Security Properties", Work in Progress, February 2004.

[ZDSZ93] L. Zhang, S. Deering, D. Estrin, and D. Zappala, "RSVP: A New Resource Reservation Protocol", IEEE Network, Volume 7, Pages 8-18, September 1993.

[URL1] http://www.atm.tut.fi/list-archive/diffserv/thrd3.html

[URL2] OPENSIG http://comet.columbia.edu/opensig/

[URL3] SIGLITE http://www1.cs.columbia.edu/~pingpan/projects/siglite.html

Authors' Addresses

Jukka Manner
Department of Computer Science
University of Helsinki
P.O. Box 68 (Gustav Hallstrominkatu 2b)
FIN-00014 HELSINKI
Finland

Phone: +358-9-191-51298
Fax: +358-9-191-51120
EMail: jmanner@cs.helsinki.fi

Xiaoming Fu
Institute for Informatics
Georg-August-University of Goettingen
Lotzestrasse 16-18
37083 Goettingen
Germany

Phone: +49-551-39-14411
Fax: +49-551-39-14403
EMail: fu@cs.uni-goettingen.de

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.