Securing RSVP For Multimedia Applications

Vanish Talwar
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801, USA
vtalwar@uiuc.edu

Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801, USA
klara@cs.uiuc.edu

ABSTRACT
Distributed multimedia applications require end-to-end quality of service(QoS) in order to be accepted and used. One approach to achieve end-to-end QoS is to provide end-to-end resource reservations. Resource ReSerVation Protocol (RSVP) [5] [1] is a unicast and multicast signalling protocol for setting up network bandwidth reservation. In this paper, we propose a solution for securing RSVP messages in a flexible, efficient and scalable manner. Our solution extends the RSVP protocol with a scalable QoS protection, using a hybrid hierarchical security approach. The RSVP messages go through two different protocol treatments - one within subnetworks and the other across subnetworks. We use delayed integrity checking within the subnetwork by sending feedback messages from the egress node. A stronger integrity and encryption check is made on messages sent across subnetworks. Our solution is thus an intermediate approach between the extremes of hop by hop authentication [2] and the SDS/CD protocol [8] and overcomes the drawbacks of the two protocols.

1. INTRODUCTION
Distributed multimedia applications require end-to-end quality of service(QoS) in order to be accepted and used. One approach to achieve end-to-end QoS is to provide end-to-end resource reservations. Resource ReSerVation Protocol (RSVP) [5] [1] is a unicast and multicast signalling protocol for setting up network bandwidth reservation. RSVP sets up a distributed state in routers and hosts and is used by a host to request specific QoS from the network for particular application data streams or flows. It is also used by routers to deliver QoS requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. The RSVP protocol raises several security issues [5] [2] [8] :
(1) Message Integrity and node(router) authentication : RSVP reservation request messages could be corrupted or spoofed up leading to theft of service or denial of service by unauthorized persons. **(2) User Authentication :** Policy control will depend largely on the positive authentication of the user responsible for each reservation request. **(3) Non-Repudiation :** A user should not be able to falsely deny later that he sent the message. **(4) Confidentiality :** The RSVP request messages must not be visible to outside parties. This implies need for encryption of RSVP messages. **(5) Replay Attacks :** The RSVP messages could be replayed by intruders causing wasteful reservations and/or theft of service. **(6) Traffic Analysis :** The outside parties should not be able to infer the RSVP requests by analyzing the traffic. This depends on the strength of the encryption mechanism used. **(7) Cut and Paste Attacks:** The outside parties may copy the RSVP requests from one RSVP flow and paste it in another leading to Denial of Service or theft of service by unauthorized persons. **(8) Denial of Service Attacks :** This can be achieved by [8] (a) Corrupting packets (b) Degrading network utilization (c) Spoofed up messages by unauthorized persons leading to theft of service (d) Dropping of packets by unauthorized persons.

Current solutions for securing RSVP are not complete in themselves and have their own pros and cons. [2] has the drawback of high overhead, expensive key management and malicious router attack. [8] has the drawback of bad scalability. We propose a solution, RSVP with Scalable QoS protection (RSVP-SQoS) protocol, using a hybrid approach applied to a hierarchical network. Our solution is flexible and also scales well for larger networks. The protocol detects intrusions as quickly as possible with a low overhead. The analysis results show that the protocol detects an intrusion within time atmost proportional to twice the size of the largest subnetwork in the path. The number of extra messages generated for every Path or Resv message is atmost equal to the number of subnetworks in the path. We describe our RSVP-SQoS protocol in the next section.

2. RSVP WITH SCALABLE QOS PROTECTION (RSVP-SQOS)
The current solutions for securing RSVP suffer from a high overhead [2] and scalability problems [8]. To make the design scalable, we divide the network into domains or subnetworks and modify the algorithm in [8]. Each subnetwork has an ingress node and an egress node. All traffic incoming to a subnetwork enters through the ingress node and all outgoing traffic goes through the egress node. In this hierarchical network we design a hybrid protocol, called RSVP with Scalable QoS Protection (RSVP-SQoS) protocol, consisting

of two major protocol processing approaches - one within a subnetwork and the other across subnetworks. The security assumptions for the two approaches are different. Within a subnetwork, we assume a weaker security assumption than that across subnetworks. So within a subnetwork, we use delayed integrity checking. When the RSVP messages go across subnetworks, a stronger integrity check is done with encryption, if necessary. Authentication is done using digital signatures. The delayed integrity check within a subnetwork is done by making each egress node send a feedback message for integrity checking of RSVP QoS parameters within that subnetwork [1]. While this integrity checking is going on, the egress node could either wait (pessimistic approach) or could forward the packet ahead (optimistic approach). The advantages and disadvantages of the two schemes are given in detail later. Whenever an intrusion is detected, TearDown messages are sent tearing the connection down and preventing further intrusions. The detailed protocol is given in the next subsection.

2.1 Protocol Design

The entire network is composed of subnetworks labeled S_i ($1 <= i <= N$) each having ingress node I_i and egress node E_i ($1 <= i <= N$) for a particular direction. The same node may act as an ingress node in one direction and an egress node in other direction.Within each subnetwork S_i we run Phase 1 of RSVP-SQoS for PATH messages and Phase 3 of RSVP-SQoS for Resv messages. Across an egress node E_i and the next ingress node I_j ($1 <= i <= N$, $1 <= j <= N$), we run Phase 2 of RSVP-SQoS. The PATH and Resv messages are illustrated in Figure 1 and Figure 2 respectively.

Phase 1 (Intra Domain PATH protocol)

1. If the subnetwork contains the Sender, the Sender digitally signs the non-mutable parameters of the PATH message, puts a timestamp and sends the message towards the egress node for this subnetwork. If this is an intermediate subnetwork, the ingress node for this subnetwork receives the PATH message from the egress node of the previous subnetwork using Phase 2 of RSVP-SQoS and forwards the message after running Phase 2 towards the egress node of the current subnetwork. Each intermediate router in the subnetwork authenticates the Sender and verifies the timestamp before processing the PATH message. The authentication is done by verifying the sender's signature using the sender's public key.The verification of timestamp is used to detect replay attacks. On success, the router passes the message ahead. If the current subnetwork contains the Receiver node, on receiving the PATH message , the Receiver uses Phase 3 from now onwards. Otherwise, we proceed to the next step.

2. When the egress node receives the PATH message,it creates a feedback PATH_FB message to detect any intrusions that might change AdSpec. This PATH_FB message contains the AdSpec(PATH) it just received digitally signed by the egress node.The egress node sends this PATH_FB message to the ingress node (or sender if this subnetwork contains the sender) along the same path along which the PATH message arrived.

3. Upon receiving this PATH_FB message, each intermediate router verifies if the signed AdSpec(PATH) is equivalent to or lower than the last AdSpec(PATH) it forwarded downstream. If it is not, a conflict has been detected and an alarm is raised to the local decision point. The local decision point decides whether the RSVP message should be dropped or not. Also, it decides whether it is appropriate to flood a TearDown message. The TearDown message could be sent in both directions or only downstream as discussed later.

Phase 2 (Inter Domain PATH/Resv protocol)

1. The egress and ingress nodes of all subnetworks decide on a shared key between every egress node - ingress node pair that are connected to each other. This could be decided based on a key management strategy and could be cached for optimization. The actual key management strategy is not specified in the protocol.

2. A key hash of the entire PATH (or Resv) message is made using this key by the egress node. The particular hashing algorithm to be used is not specified here but examples are MD5, HMAC-5 etc. Since the non-mutable components of the messages are digitally signed, a key-hash of only the mutable components may be required.The keyed hash is attached to the message(PATH or Resv). If required, the message (PATH or Resv) could also be encrypted for confidentiality. The message(PATH or Resv) is then sent to the ingress node.

3. On receiving the message from the egress node of the previous subnetwork,the ingress node creates the keyed hash of the message using the shared key and verifies it with that attached to the message. If they do not match, an intrusion must have occurred and an alarm is raised to the local decision point. The local decision point will decide whether the RSVP message should be dropped or not. Also, it will decide whether it is appropriate to flood a TearDown message. The TearDown message is sent in only one direction.

4. If the keyed hash matches, the ingress node authenticates the Sender and verifies the timestamp before processing the PATH (or Resv) message. The authentication is done by verifying the sender's signature using the sender's public key.The verification of timestamp is used to detect replay attacks. On success, it forwards the message ahead.

5. As in [2], we could maintain a security association for each authenticating key shared between an ingress and egress node. In this case, the key would be the unique number for a given sending router and would be used to determine the security association to use.

[1] The sender advertises QoS parameters in the AdSpec structure, the receiver specifies its QoS parameters in the RSpec structure.

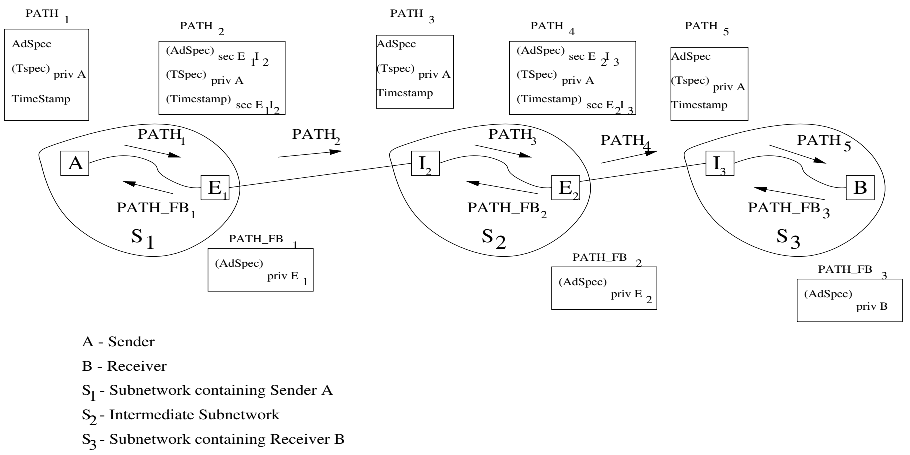

Figure 1: The PATH messages using Phase 1 and Phase 2

Phase 3 (Intra Domain Resv protocol)

1. If the subnetwork contains the Receiver, the Receiver digitally signs the RSpec parameter of the Resv message, puts a timestamp and sends the message towards the egress node for this subnetwork. If this is an intermediate subnetwork, the ingress node for this subnetwork receives the Resv message from the egress node of the previous subnetwork using Phase 2 of RSVP-SQoS and forwards the message after running Phase 2 towards the egress node of the current subnetwork. Each intermediate router in the subnetwork authenticates the Receiver and verifies the timestamp before processing the Resv message.The authentication is done by verifying the receiver's signature using the receiver's public key.The verification of timestamp is used to detect replay attacks. On success, the router passes the message ahead. If a merge from multiple Resv messages is necessary, the router picks the Resv message with the largest RSpec value and forwards it upstream. If the current subnetwork contains the Sender node, on receiving the Resv message , the Sender uses Protocol 1 from now onwards. Otherwise, we proceed to the next step.

2. When the egress node receives the Resv message,it will create a feedback RESV_FB message to detect any intrusions that would have occurred in the subnetwork. This RESV_FB message contains the RSpec(RESV) it just received digitally signed by the egress node.The egress node sends this RESV_FB message to the ingress node (or receiver if this subnetwork contains the receiver) along the same path along which the Resv message arrived.

3. Upon receiving this RESV_FB message, each intermediate router verifies if the signed RSpec(RESV) is lower than the last RSpec(RESV) it forwarded upstream. If it is not, a conflict has been detected and an alarm is raised to the local decision point. The local decision point decides whether the RSVP message should be dropped or not. Also, it decides whether it is appropriate to flood a TearDown message. The TearDown message could be sent in both directions or only downstream as discussed later.

2.2 Protocol Evaluation

The proposed protocol achieves the following: (1) Integrity of constant objects in the RSVP messages through Digital Signatures. The intrusions to the constant objects would cause the integrity check to fail when a router authenticates the sender/receiver. (2) Integrity of mutable objects in the RSVP messages through a combination of delayed integrity checking and subnet to subnet integrity check. Any unauthorized increase to the AdSpec or RSpec parameter within a subnetwork would be detected using feedback PATH_FB and RESV_FB messages. Any intrusion made to messages across subnetworks would be detected by the subnet to subnet integrity check. (3) User Authentication through Digital Signatures. Every PATH (Resv) message is digitally signed by the Sender (Receiver). (4) Non-repudiation through Digital Signatures. Since a digital signature is unique to a user, it cannot deny sending the message. (5) Protection against Replay attacks through timestamps. A message with a duplicate timestamp would be discarded. (6) Subnet to Subnet Confidentiality through encryption but is optional. (7) Protection against Malicious Router Attack through delayed integrity check. Any malicious increase in the AdSpec or RSpec parameter by a malicious router would be detected using delayed integrity check within the subnetwork. To detect malicious end routers (egress or ingress nodes), the feedback messages would go upto the egress node of the previous subnetwork.

The RSVP-SQoS protocol provides, furthermore,

Flexibility : The subnetwork can be flexibly chosen. If we assume the subnet sizes to be one each, then we would be converting the protocol to a hop by hop authentication as in [2]. If we assume a single subnet, then we would be using end to end authentication with delayed integrity checking as in [8]. Thus the protocol can be flexibly adapted to choose subnets of different sizes depending on the security assumption in the subnets.

Scalability : The idea of using hierarchical networks based on subnetworks directly gives us a scalable architecture. The protocol also scales well for large networks since the delayed integrity checking is only within a subnetwork. Thus intru-

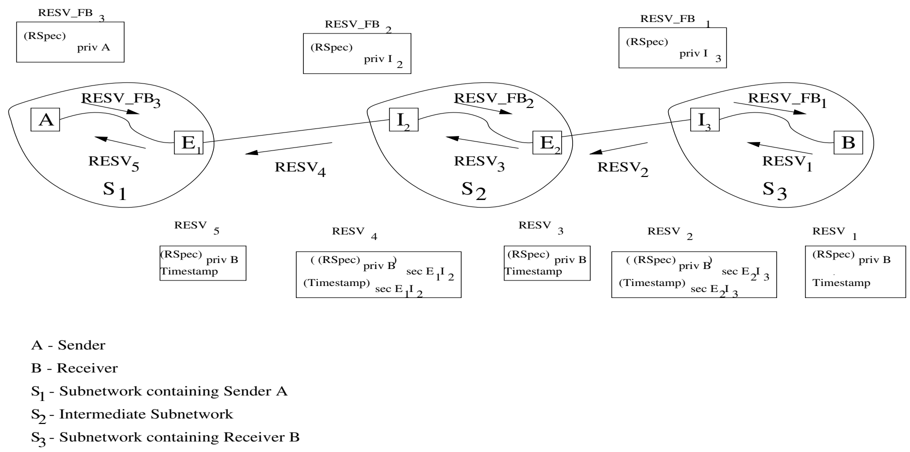

Figure 2: The Resv messages using Phase 3 and Phase 2

sion detection latency is not affected by increase in network size.

Multicast Support : When multicast flows are used, then the router picks up the Resv messages with the largest RSpec value. When a merge takes place at a router, a feedback message is sent from this router to the previous router where the last merge took place or to the ingress node, whichever is closer. The feedback message from the egress node is sent similarly. In case the request cannot be satisfied, the reservation requested by that Receiver is cancelled. However, the feedback message would still be sent.

2.3 Discussion

If we do not want unnecessary resource reservations to be made while the intrusion is being detected within a subnetwork, we can adopt a pessimistic approach and make the egress node to wait for a TearDown message for a known round trip time of the subnetwork. In this case, whenever an intrusion is detected, we would send TearDown messages in both directions. If the egress node does not receive a TearDown within the known round trip time, it assumes that no intrusion took place and forwards the message ahead using Phase 2. If we do not want to wait for intrusion detection, we can adopt an optimistic approach and the egress node can forward the message ahead using Phase 2. In this case, it would be sufficient to send the TearDown message only towards the sender. The reservations being made would be wasteful but would timeout ultimately since the sender (or receiver) would not send further PATH (or Resv) messages. We could also send TearDown messages in both directions which would have the effect of tearing down the reservations as they are made. The latter solution seems better since in this case the Resv (or PATH) messages would not be sent upstream (or downstream) saving on reservations. We also note that the PATH_FB (or RESV_FB) for the subnetwork containing the receiver (or sender) could be piggybacked with the next Resv (or PATH) message. This is an implementation optimization.

3. CONCLUSIONS

In this paper, we identified the security problems of the existing RSVP protocols [5] [2] and proposed a solution that is flexible and scalable for larger networks. Our design views the network as a set of subnetworks. The security assumptions in a security domain are less stricter compared to that across subnetworks. The RSVP messages follow different approaches within a subnetwork and across subnetworks. It achieves integrity checking of the RSVP messages, authenticates the sender of the messages and protects against replay attacks. It aims at providing a scalable and flexible solution minimizing the delay in detecting intrusions with low overhead. The lower overhead in time is achieved by subnet to subnet authentication as against hop by hop authentication [2] and lower overhead in space is achieved by not storing all keys.

4. REFERENCES

[1] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'dell, A. Romanow, A. Weinrib, and L. Zhang. Resource ReSerVation Protocol – Version 1 Applicability Statement Some Guidelines on Deployment. *RFC 2208*, September 1997.

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

[3] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing For Message Authentication. *RFC 2104*, March 1996.

[4] S. Mittra and T. Y. Woo. A Flow-Based Approach to Datagram Security. *Proceedings ACM SIGCOMM*, 1997.

[5] R.Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol – Version 1 Functional Specification. *RFC 2205*, September 1997.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture For Differentiated Services. *RFC 2475*, December 1998.

[7] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. *RFC 2401*, November 1998.

[8] Tsung-Li Wu, S. Wu, and F. Gong. Securing QoS: Threats to RSVP Messages and Their Countermeasures. *Proceedings IWQoS*, pages 62–64, 1999.