University of Würzburg
Institute of Computer Science
Research Report Series

# Performance Evaluation of the Extensions for Control Message Retransmissions in RSVP

Michael Menth, Rüdiger Martin

Report No. 288 November 2001

Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
E-Mail:{menth|martin}@informatik.uni-wuerzburg.de

# Performance Evaluation of the Extensions for Control Message Retransmissions in RSVP

**Michael Menth, Rüdiger Martin**
Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
E-Mail:{menth|martin}@informatik.uni-wuerzburg.de

**Abstract**

Quality of Service for real-time transmission can be achieved by resource reservation in the routers on the data path. In the recent years several protocols and extensions to them have been designed for signaling resource reservation in different ways. This work reviews various protocols that exhibit different signaling concepts. Then we study control message retransmissions for RSVP-like protocols. Numerical results illustrate their behavior in different networking scenarios and quantify the performance gain.

**Keywords:** QoS, RSVP, performance evaluation, signaling, resource reservation

## 1 Introduction

Future communication networks will guarantee seamless quality of service (QoS) data transportation from the sender to the receiver. The network must provide sufficient resources to forward the data in an adequate way to meet the loss and delay requirements of the traffic. To achieve that goal, massive overprovisioning can be applied as well as intelligent traffic engineering techniques. One of them is admission control (AC): When the network's capacity does not suffice to transport all offered traffic, AC shelters the network from overload by admitting only a limited number of reservation requests. Thus, the QoS for the flows in place is maintained at the expense of blocked flows. In order to perform AC in a network entity, the size of a reservation request must be known beforehand and is usually delivered by a resource reservation protocol.

In the recent years, several resource reservation protocols have been designed for IP networks. In this paper we would like to give an overview over the most prominent protocols: RSVP, RSVP refresh overhead reduction extensions, aggregation of RSVP reservations, Boomerang, YESSIR, BGRP, and stateless reservation protocols. These protocols not only different syntax and semantic but reveal also different information passing concepts.

In the past, protocol implementations have been studied using software implementations [1, 2, 3, 4]. In [5] the reliability of RSVP was studied. In our investigation we focus on the new extensions of RSVP and evaluate their behavior regarding the reservation establishment delay (RED) and the reservation teardown delay (RTD). We concentrate on general features of RSVP-like protocols and experiment with different configurations to test them in various networking scenarios.

This paper is structured as follows. In Section 2 we present the above mentioned protocols. Then, we study the response time of RSVP-like protocols with special respect to the control message retransmission feature in the RSVP extensions [6]. Numerical results illustrate their behavior and quantify the performance gain. In Section 4 we summarize this work.

## 2 An Overview of Resource Reservation Protocols

In the context of real-time applications like voice over IP (VoIP) or video conference there are important signaling protocols for the application layer as well as for the network and transport layers.

Applications need to identify and locate their communication peers and to negotiate session parameters. Codecs have to be agreed and translators can be involved in case of incompatible end systems. These and other tasks are performed by standards like H.323, the session Initiation Protocol (SIP), and the Media Gateway Control Protocol (MGCP).

On the transport layer we have the User Datagram Protocol (UDP), the Transmission Control Protocol (TCP), the Real-time Transport Protocol (RTP), and others. They ensure e.g. that IP packets are associated with the correct ports in the end systems, that packet loss can be detected, and provide means for data synchronization to avoid distorted time lines for presentation.

The network layer ensures that the data arrive at the correct destination. It consists of the routing protocols and the IP header.

The link layer offers QoS mechanisms as it has the control over the overall capacity of a link. Resource reservation can be performed per hop on all intermediate links between sender and receiver. The required signaling often takes place inband which means that the signaling information is carried over the same network. The control messages are transported using regular IP packets (network layer) but their information relates to the respective link layer.

This work focuses on signaling for resource reservation. In this section we present various existing protocols and concentrate on their information forwarding paradigm.

### 2.1 Resource Reservation Protocol (RSVP)

RSVP has been conceived by the IETF to signal reservation request within an Integrated Services network [7, 8]. Both unicast and multicast applications are supported and different reservation styles are possible.

**Connection Establishment** To initiate a reservation with RSVP, the sending node issues a so called PATH message that establishes a PATH state in the intermediate hops on the way to the desired destination machine. The flow related information in a router is referred to as the state of this flow. The destination router establishes a RESV state and responds with an RESV message that visits the intermediate routers in the reverse direction using the previous hop information of the PATH state (cf. Figure 1).

This ensures that the path is reserved that the data takes from the sender to the receiver. When an RESV message is received by a router, the required actions are taken to set up the reservations for the respective data flow (cf. Figure 2).

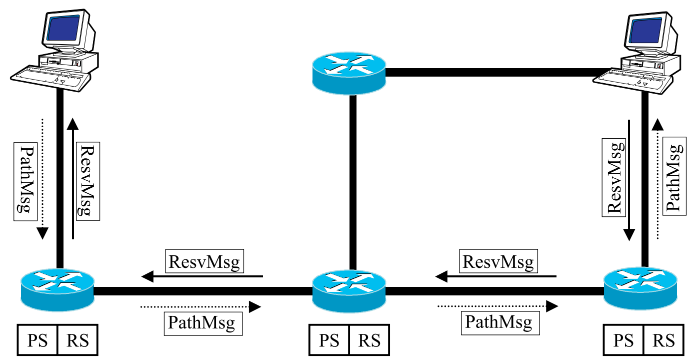


Figure 1: Signaling with RSVP.

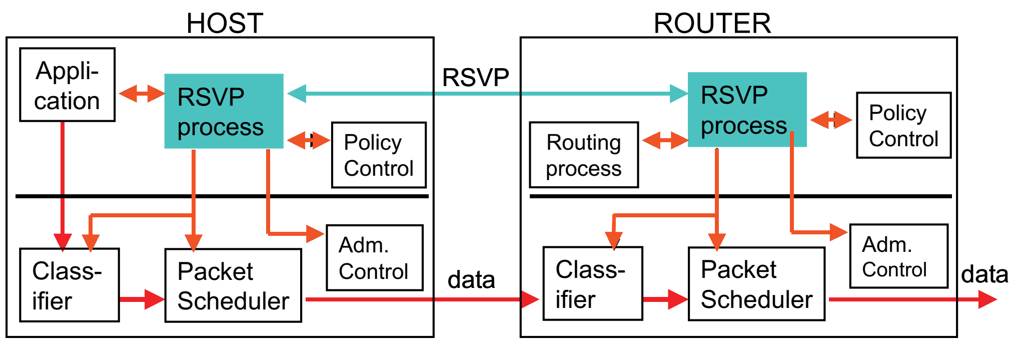


Figure 2: Actions taken by an RSVP process.

Based on the flow and reservation specifiers the admission decision is made. When the request succeeds, the classifier and the scheduler are configured to forward the data flow messages. The signaling requires one pass from the sender to the receiver to collect advertising information that is delivered to the receiver to enable appropriate reservation request. The actual reservation is made on the way back to the sender. Explicit PATHERR and TEARDOWN messages are able to tear down the connection and remove the states in the routers. RSVP uses a two pass signaling approach, also known as one pass with advertising (OPWA).

**Soft States** All the RSVP control messages are sent directly in IP datagrams using protocol number 46 or in UDP packets that are not protected by the reservation for the data flow. In addition, the end systems may go down without notifying the network about it. Thus, the communication is inherently unreliable. RSVP uses a soft state approach to cope with that: The states time out and disappear after a cleanup time $L$ unless they are refreshed by another PATH or RESV message. To keep the connection alive, every participating node sends periodically PATH and RESV messages to its next hop with refresh period $R$. $L$ is usually set to $3 \cdot R$. If the source stops without tearing down the connection or in case of routing changes, the PATH and RESV states will eventually time out in all the obsolete routers.

## 2.2 RSVP Refresh Overhead Reduction Extensions

The RSVP refresh overhead reduction extensions [6, 9] try to reduce the overhead that is due to the refresh messages. This is done by several amendments.

**BUNDLE Messages** RSVP nodes send one PATH and RESV message per refresh interval and connection. Since several connections are carried on the same link, their refresh messages can be handled within one single BUNDLE message where just the different message bodies are assembled. This yields just the reduction of the mere control packet frequency but not of the control message frequency. The control message rate is negligible and reduced only a little by that. There are also as many operations required by the router as in normal RSVP.

**Complexity Reduction by MESSAGE_IDs** PATH and RESV messages are sent periodically per RSVP connection and do not change. Nevertheless, the receiver has to identify the corresponding flows and refresh their states. To alleviate this, every control message that changes a state is equipped with a unique MESSAGE_ID. Consecutive control messages that just refresh this state are equipped with the same MESSAGE_ID. The desired state is then identified by a hash value using the MESSAGE_ID and can be refreshed without processing the whole control message.

**Control Message Retransmissions** It is possible to set an ACK_DESIRED flag within a MESSAGE_ID object to indicate the receiver to send a MESSAGE_ID_ACK to acknowledge the receipt of a control message. If the sender has not yet received the MESSAGE_ID_ACK object after $R_f$ time, it retransmits the respective control message. To adjust this mechanism to potential network overload situations, an exponential backoff algorithm is introduced to avoid unnecessary control messages.

**Summary Refresh Extensions** Last but not least, the MESSAGE_IDs do not require to be sent with their related control message. A MESSAGE_ID LIST object may contain only MESSAGE_ID objects instead of the whole control message that would only be used in the failure case. This reduces also the required bandwidth for signaling, however, this has not an impact on the network utilization since the fraction of signaling traffic is small anyway. To handle cases where the receiver encounters an inconsistent state view, the receiver may order new PATH or RESV messages by issuing a MESSAGE_ID_NACK that refers to the corrupted RSVP connection.

## 2.3 Aggregation of RSVP Reservations

The above mentioned modifications to RSVP tend to reduce the protocol overhead per RSVP control message and allow better performing implementations of the RSVP state machine. However, they are not able to solve the fundamental scaling problem: The processing costs in a router grow linearly with the number of supported reservations which is feasible in the access network with only a few QoS flows but not in a core network. Therefore, [10] suggests an aggregator at a border router of a network that summarizes many individual RSVP reservations that share the same path through the network into one aggregated reservation. At the egress point, the reservations are deaggregated. This reduces the number of reservations drastically within the network and relieves the core routers. The same objective can be achieved by using aggregation by Multiprotocol Label Switching (MPLS) [11, 12, 13].

### 2.4 Boomerang

The Boomerang protocol [14] aims at reducing the overhead that is induced by RSVP. There are no PATH messages and the sender generates a reservation message that goes hop by hop to the receiver. Along that path, the reservations are performed in the routers that understand Boomerang. As soon as the message arrives at the receiver, the reservation is already in place. The receiver just needs to bounce the message back to the sender to notify it, the receiver does not even need to process the message. As an option, the return channel of a bidirectional session may be reserved on the way back. Note that a different path may be taken for that purpose (cf. Figure 3).

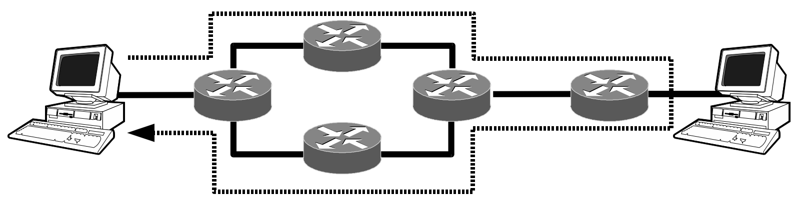

Figure 3: A bidirectional reservation setup by Boomerang.

If a reservation request fails or if a session terminates, the reservation states can be torn down with a reservation request of size zero. Only the sending node generates signaling messages, therefore, the Boomerang approach is simpler than RSVP since the major complexity and processing is located at the sender. The concept is also based on soft states and requires refreshes to keep the reservation alive. If the Boomerang message does not return to the initiating node within a certain time, it is considered to be lost, so that the sender can take appropriate actions. In [3] it is shown that the Boomerang protocol induces clearly less burden on the routers than conventional RSVP implementations.

### 2.5 YESSIR - YEt another Sender Session Internet Reservation

YESSIR [15] is a reservation protocol that is based on RTP [16]. RTP is usually a wrapper for UDP packets to add sequence numbers, time stamps and other identifiers. It comes along with the Real-time Transport Control Protocol (RTCP). Each session consists of one RTP data stream and one corresponding RTCP stream. Senders and receivers send periodically sender and receiver reports (SR, RR). SRs contain throughput and other information about the last report interval that allow e.g. to derive the current round-trip time in the network. RRs indicate packet loss and delay statistics among others. This is extremely useful for adaptive applications. YESSIR works like RSVP also in a unicast and a multicast environment and offers also different reservation styles.

YESSIR reservation messages are piggybacked at the end of RTCP SR or RR messages, possibly enhanced by additional YESSIR-specific data, carried in IP packets with router-alert option and are processed by intercepting routers that support this option. As with Boomerang, reservations are triggered by the sender. If a router along the way is not able to provide the requested resources, the exact reasons for the reservation failure can be remarked. This helps the end systems to either drop the session or to lower the reservation request. The rate for the reservation can be given explicitly, it may be deduced from codec types in the RTP payload or it may also be inferred from the size of the payload and the corresponding time stamps. YESSIR also relies on the soft state approach. As in Boomerang, only the sender issues refreshes and the session can be torn down with an explicit RTCP BYE message. Unlike in RSVP, the intermediate nodes are not able to issue ERROR messages, failure situations have to be recognized by the receiver and reported via RRs to the sender.

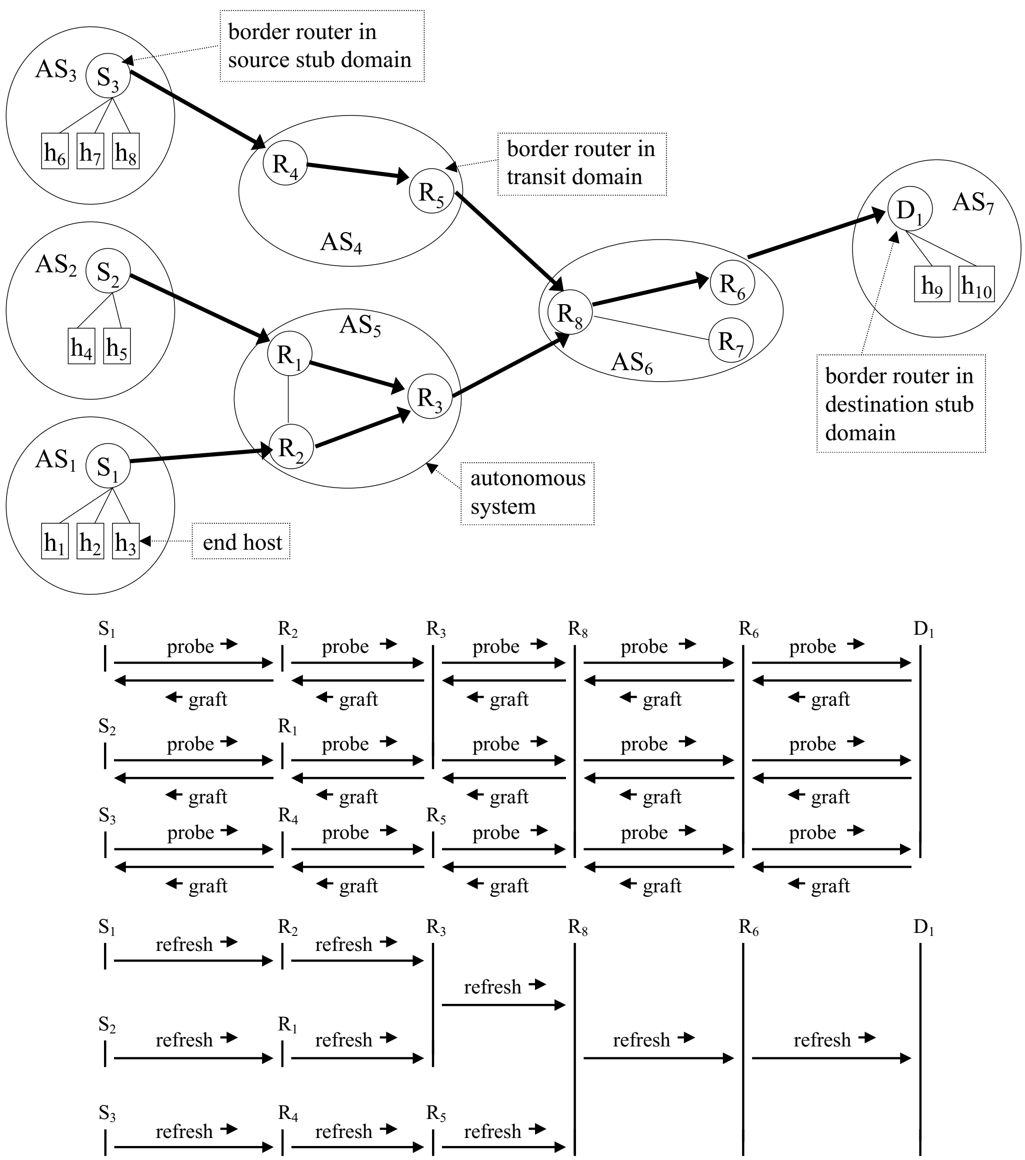


Figure 4: Signaling in BGRP.

### 2.6 Border Gateway Reservation Protocol (BGRP)

BGRP [17] has been conceived for inter-domain use and to work in cooperation with the Border Gateway Protocol (BGP) for routing. BGRP addresses the scalability problem directly since it is designed to aggregate all reservations with the same autonomous system (AS) as destination into a single funnel reservation, no matter of their origin. Note that BGRP is only used for reservations between border routers.

The messages in BGRP are exchanged reliably between the neighboring border routers. PROBE messages consist of a reservation request and information about the destination network. They collect the border routers on the path from the source to the destination. The destination returns a GRAFT message back on the same path to reserve the required resources in the involved border routers. These routers keep only a single reservation per sink tree which is identified by the destination AS and its border router. The required demand in the GRAFT message is simply added to the already established reservations. Therefore, PROBE and GRAFT messages are only used to set up a reservation. ERROR messages are sent to recover from reservation failur. The soft state approach in BGRP is implemented as follows. REFRESH messages are sent from the senders down the sink tree in periodic intervals. When a REFRESH message arrives at an egress router, it recomputes the total reservations size for the corresponding sink tree and forwards the modified REFRESH message downstream. The use of the different signaling messages is depicted in Figure 4. The source routers $S_1$, $S_2$, and $S_3$ set up a funnel reservation to destination router $D_1$ using PROBE messages. GRAFT messages are sent back and after a some time, the sink tree is refreshed.

Here, we have also receiver based reservations but in contrast to RSVP, the information is not stored in the nodes but it is contained in the PROBE and GRAFT packets. The advantage of the sink tree or reservation merging approach is its scalability. In every border router there is only one sink tree reservation for everey destination AS. Hence, the number of BGRP reservations scales linearly with the number of AS.

### 2.7 Stateless Reservation Protocols

The above described reservation protocols exhibit the disadvantage that they have to keep a record for either individual reservations, for tunnel aggregates, or for funnel aggregates. Another approach to guarantee QoS are stateless reservation protocols. We only describe the basic architecture, for further details the reader is referred to [18, 19, 20].

A new reservation is only admitted if its request passes all AC test in the intermediate routers and the destination signals this back to the source. If such a test fails, the message is just discarded or marked to indicate a failure. Packets that are sent by the end systems under reservation have a special tag. Instead of keeping a record for different connections, the router analyzes the packet streams on each outgoing link. It counts the packets with the reservation tag within a given interval and infers the reserved rate $R_{res}$. This requires of course that the holder of a reservation sends also packets when the application is idle. The overall rate of the newly admitted sessions $R_{new}$ is also recorded over this interval and the sum of $R_{res} + R_{new}$ is an upper bound on the reserved rate on an output port. This is only the basic mechanism that does not reveal the manifold implementation problems.

### 2.8 Measurement Based Admission Control

The above presented resource reservation approaches are primarily used to perform AC based on the declared traffic parameters. As an alternative, measurement based AC (MBAC) may also be done. This means that the AC relies on the current traffic intensity of the network. To support this architecture, the signaling protocols may be the same as in the conventional case, however, some special purpose protocols for MBAC are also conceived [21, 22, 23].

## 3 Performance Evaluation of Control Message Retransmissions

In [6] RSVP was enhanced by retransmissions for control messages (CMR, see also Section 2.2). They make the communication for RSVP control traffic more reliable and lead to faster reactions of the RSVP processes in the involved routers. In this section, we study the impact of different parameters and options on this response time. This is important since the reaction time of the remote processes affects the reservation establishment delay (RED) and the reservation teardown delay (RTD). The RED influences very much the response time of the system which is a crucial factor for the user perception. Unused and blocked capacity is not profitable, therefore, resources should be released very quickly after session termination by the application layer. This requires short RTD. First, we give a short description of the options under study and illustrate then their influence on RED and RTD.

### 3.1 Model Description

In our investigation we consider a general signaling protocol with CMR and several options. We borrow most of the nomenclature from RSVP but we do not limit our experiments to configurations in RSVP. We neglect the message processing times in the routers and focus on the effect of the mere transmission times and involved timeout values to compute performance measures RED and RTD. The calculations are lengthy but straightforward, so we omit them in this presentation.

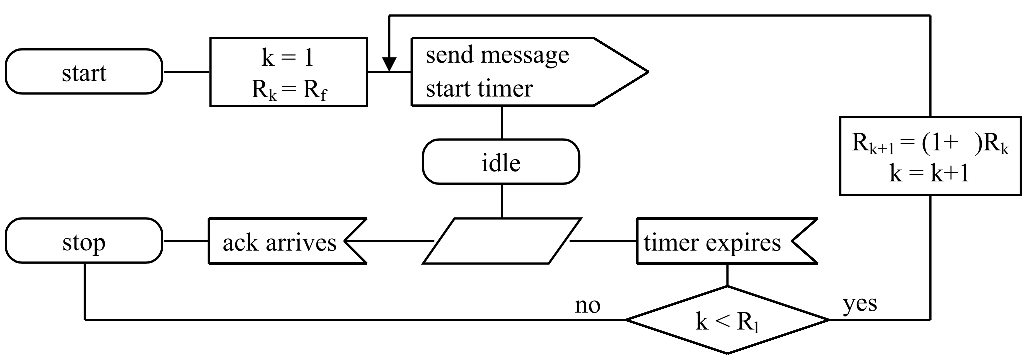


Figure 5: A flowchart of the retransmission algorithm.

**Control Message Retransmissions** We recall briefly the concept of CMR and point out the influencing parameters. The reservation process of a sender issues a control message and sets the retransmission timer. The receiver of this message is required to immediately return an acknowledgement. If the sender does not receive an acknowledgement before the timer expires, the control message is retransmitted. The retransmission timer value $R_k$ depends on the $k^{th}$ retransmission interval and contributes as well to the delay:

$$R_k = (1+\Delta)^{k-1} \cdot R_f, \qquad 1 \leq k < R_l.$$

$R_k$ scales linearly with the rapid retransmission interval $R_f$. An exponential backoff is applied and $\Delta$ (we use $\Delta = 1$) governs the speed at which the sender increases the timer value. This is to avoid unnecessary retransmissions due to links with long transmission delays. The rapid retry limit $R_l$ is an upper bound on the number of control message transmissions without response of an acknowledgement. A flowchart of the algorithm is depicted in Figure 5. This concept reduces only the response time of RSVP but it does not yield a reliable communication. The parameter $R_l = 1$ corresponds to conventional signaling. In case that no acknowledgement returns, the sender tries again after a refresh interval of $R$ time. If a node has not received an update message after $3 \cdot R$, it faces a soft state timeout and sends a teardown control message to indicate the end of the session to its neighboring nodes.

**Endpoint versus Common Control** We have noticed that in Boomerang or YESSIR only the endpoints (sender, receivers) trigger control messages that travel along the path to the receiving node while in RSVP every node controls the connection. This means that they do not only forward the control messages when they arrive, they also create refresh (PATH, RESV) messages when they do not receive them in time. We call the first approach "endpoint control" and the second one "common control". With common control and CMR, the ACKs are created and returned by neighboring hosts and not as under endpoint control by the receiving peer over many intermediate hops. Common control seems to make a reservation more robust against loss of control messages, especially in combination with CMR. Therefore, we study this scenario.

**One-Pass versus Two-Pass** In RSVP OPWA is used for establishing a connection which is in fact a two pass approach: One pass is needed for setting up the PATH states in the router and one pass back is required for setting up the reservations. The same holds for BGRP's PROBE and GRAFT messages. In Boomerang and YESSIR this is different. The reservation is done with the first pass from the sender to the receiver. The successful session setup may be notified to the source or not, therefore, this is a true one-pass approach. With two-pass, the signaling takes twice as long as with one-pass. This is a relatively trivial result. To simplify the analysis, we concentrate only on the one-pass approach. As a consequence, the following results for RED must be doubled in case of true RSVP OPWA.

**Network Parameters** The effects of the retransmission timers depend certainly on the networking scenario. We make the following assumptions. We set the transmission delay per link to 10 milliseconds. The packet loss probability $p_l$ on a single link influences the system as well as the number of hops $n$ in the reservation path. Therefore, we conduct studies varying theses parameters. If the parameters are constant, we assume a path length of $n = 10$ hops and a link packet loss probability of $p_l = 10^{-2}$ which occurs in congested networks. For $n = 10$ hops, this yields an end-to-end packet loss probability of 10%. We observe these values e.g. on transatlantic links with the statistic tool for UDP traffic in realaudio or realvideo. Especially in these situations, reservations are crucial for real-time applications.

### 3.2 Reservation Establishment Delay

We are interested in the influence of the refresh interval $R$ on the RED when the retransmission option is not used and only endpoint control is assumed. This is the case in YESSIR. In Figure 6 the mean of the RED (E[RED]) is shown depending on the packet loss probability $p_l$ of a single link. The establishment delay is almost constant for small loss probabilities up to $p_l = 10^{-4}$ and the effect of $R$ is negligible. E[RED] rises with increasing loss rates and the difference between various values for $R$ becomes visible. For high packet loss probabilities ($p_l = 10^{-2}$) the refresh interval $R$ dominates RED almost linearly since $R$ is several orders of magnitude larger than the transmission delays. They become less important under these circumstances.

We set $p_l = 10^{-2}$ and observe the system for different lengths of the reservation path. Figure 7 shows that the establishment delay behaves linearly to the path length. At first sight this seems to be a consequence of the summation of link transmission times but this is not the case because the size of a round trip time is in the order of hundred milliseconds. This phenomenon is rather due to end-to-end loss probabilities that are raised by the number of hops. This explains the linear influence of the retransmission timer $R$, too.

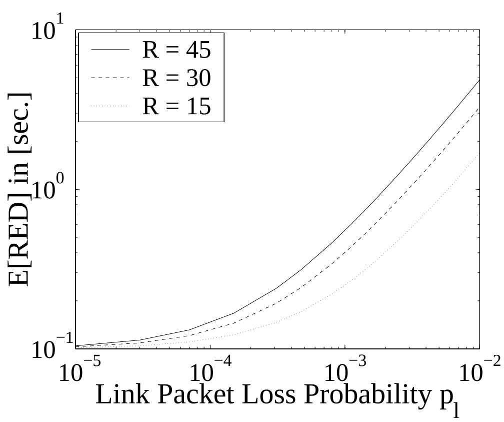


Figure 6: The impact of $R$ without CMR on E[RED] for constant path length.

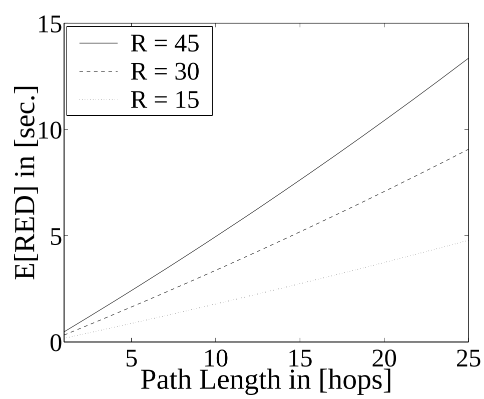


Figure 7: The influence of $R$ without CMR on E[RED] for high loss rates.

The RSVP CMR option has been standardized to reduce the signaling delay in lossy networks. The retransmission interval $R_f$ influences the retransmission times at most linearly and we set it to 0.5 seconds. Figure 8 shows that the retransmission option greatly reduces the response time of the system when we compare this alternative with Figure 6. For $p_l = 10^{-3}$ E[RED] is still negligible but for large loss rates, the effects of the rapid retry limit $R_l$ can be observed. They reduce the influence of the refresh interval $R$ and yield short RED. Even a single retransmission ($R_l = 2$) reduces its mean from about $3$ to $0.5$ seconds.

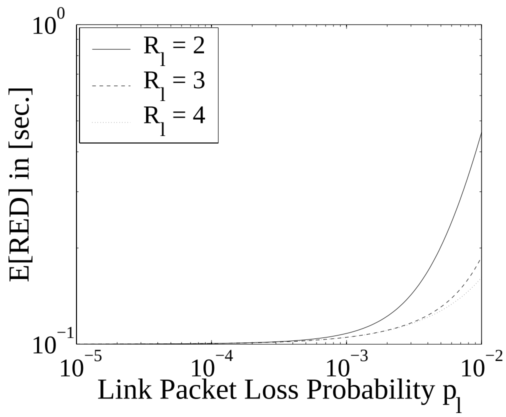


Figure 8: The impact of $R_l$ with CMR on E[RED] for constant path length.

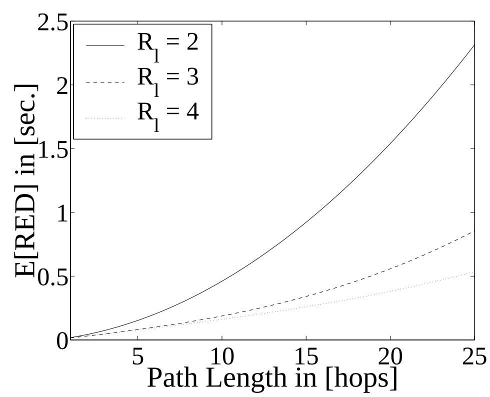


Figure 9: The influence of $R_l$ with CMR on E[RED] for high loss rates.

In Figure 9 ($p_l = 10^{-2}$) we observe that E[RED] scales linearly with the length of the reservation path. If we compare it to Figure 7 we realize that the reduction of E[RED] in absolute time rises with the length of the reservation path. Therefore, CMR is even more important for long paths. In the following experiments, we set $R_l = 3$.

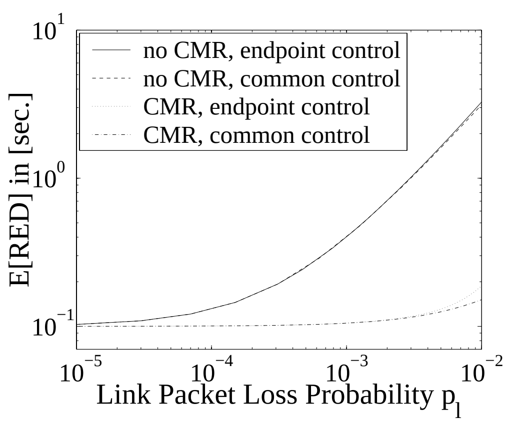


Figure 10: The impact of the control and retransmission option on E[RED] for constant path length.

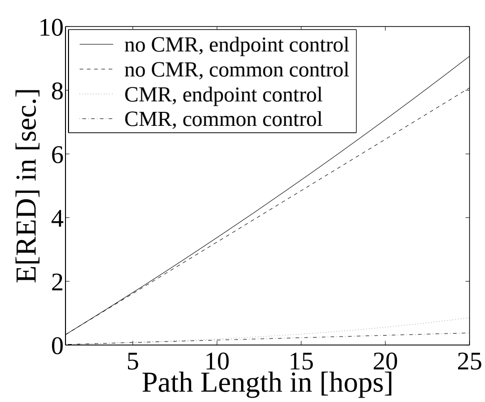


Figure 11: The influence of the control and retransmission option on E[RED] for high loss rates.

So far, we have considered only endpoint control. With common control, the nodes are more active: They generate refresh messages by themselves and serve as peer for CMR. Without CMR this means that after the loss of an initial PATH or RESV message, an intermediate router triggers a refresh message after $R$ time. Then, the remaining distance to the receiver is shorter than with endpoint control where only the endpoints issue control messages. However, this has no impact as Figure 10 shows: Without CMR, common control is hardly better than endpoint control. Figure 10 also illustrates impressively the effect of CMR with rising loss probabilities. With CMR, E[RED] stays small while without CMR it rises notably. In case of common control we have fewer losses between CMR peers (link loss probabilities) than for endpoint control (end-to-end loss probabilities). Therefore, we can see a difference for large loss rates between endpoint and common control with CMR.

We conduct the same experiment for $p_l = 10^{-2}$ and vary over the length of the reservation path. The absolute difference of E[RED] grows with the number of hops but E[RED] seems to scale linearly (cf. Figure 11). In case of CMR, E[RED] stays below one second which means that most of the delay is produced by the link transmission delay ($t_l = 10$ milliseconds) and that the delay due to the retransmission interval $R$ is minimized.

### 3.3 Reservation Teardown Delay

Resources that are not utilized any more can only be reused after the reservation is torn down. Hence, the network is not profitable for the RTD and it is important to keep the RTD small. We investigate the behavior of RTD with and without CMR.

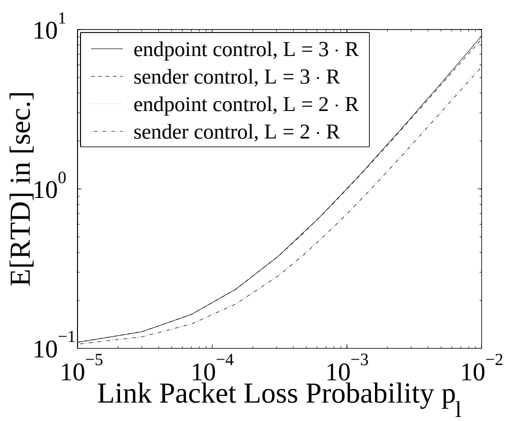


Figure 12: The impact of $L$ and the control option without CMR on E[RED] for constant path length.

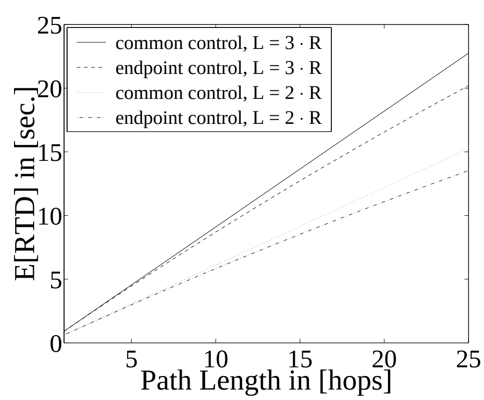


Figure 13: The influence of $L$ and the control option without CMR on E[RED] for high loss rates.

The RTD rises if a TEARDOWN message is lost from its sender to receiver. Due to the soft state concept, the cleanup timer will expire in an intermediate node and tear down the reservation after $L$ time. If only the endpoints control the session, all nodes time out after $L$ since the terminating endpoint refrains from sending refresh messages. If the reservation is under common control (like in RSVP), only the first router is not refreshed because it generates refresh messages autonomously in periodic intervals. When it times out, its teardown message can be lost as well which leads to a maximum RTD of $n \cdot L$ where $n$ is the length of the reservation. However, Figure 12 shows that difference between endpoint and common control is not important for $n = 10$, the lines are hardly to distinguish. The influence of $L$ rises with increasing loss probability and dominates E[CTD] for high link loss probabilities by a linear law.

The same phenomenon can be observed in Figure 13 ($p_l = 10^{-2}$). But the difference between endpoint and common is visible for long reservation paths. Here, common control is clearly worse than endpoint control. But the influence of the value $L$ for the expiration timer is still more important. In the following, we set $L = 3 \cdot R$.

We investigate the influence of the CMR option and its parameters on the RTD. With CMR, up to $R_l$ teardown messages are sent repeatedly until an acknowledgement returns which reduces the RTD. This becomes clear if we compare Figure 14 to Figure 12. Now, endpoint control has a longer RTD than common control which is also due to the fact that end-to-end loss probabilities are higher than link loss probabilities. For the common control option, a rapid retry limit of $R_l = 2$ already suffices to keep E[RTD] small, however endpoint control with $R_l = 4$ exhibits an excellent performance as well.

Figure 15 shows that the combination of endpoint control and $R_l = 2$ performs relatively poorly especially with increasing path length. One retransmission for CMR is not enough to cope with the increased end-to-end packet loss probability. However, the single retransmission reduces E[RTD] to 25% compared to without CMR (Figure 13).

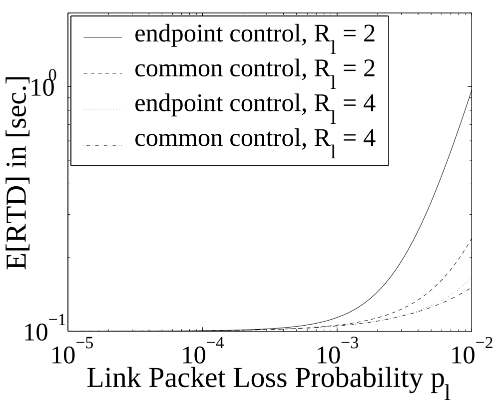


Figure 14: The impact of $R_l$ and the control option with CMR on E[RED] for constant path length.

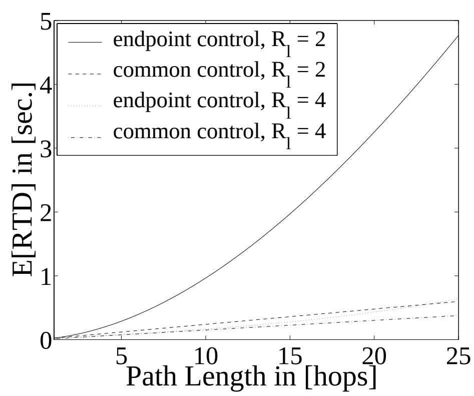


Figure 15: The influence of $R_l$ and the control option with CMR on E[RED] for high loss rates.

### 3.4 Summarizing Remarks

Finally, if packet loss rates are high, CMR is always a powerful method to reduce RED and RTD. In our investigations CMR turned out to be more efficient than common control which is also a method to improve session stability. The combination of both techniques is possible and leads to optimum results, however, they also increase the software complexity.

# 4 Summary

In this paper, we gave an overview of several resource reservation protocols. We presented their basic operations to give an idea about their principle information passing concepts.

We explained RSVP and various extensions of the protocol that reduce the number of refresh messages, make it more robust against packet losses and more scalable for its use in transit networks. Apart from RSVP, light-weight protocols as Boomerang and YESSIR were presented. BGRP is a different approach which is intended for reservation aggregation between autonomous systems. Its design entails excellent scaling behaviour since funnel reservations are used. We also presented the principle of stateless reservation protocols that only keeps one state per outgoing link and not per reservations.

In RSVP every node supporting a reservation is actively involved in keeping the reservation alive (common control). This is unlike in Boomerang or YESSIR (endpoint control). Recently, an option for RSVP control message retransmission was created to make RSVP more responsive in networking scenarios with high packet loss probabilities.

We investigated these protocol concepts that are basic features for general signaling protocols. They have an influence on the reservation establishment delay and on the session teardown delay. Their impact depends both on the packet loss probability of a single link as well as on the number of hops in the reservation path. For small loss probabilities ($< 10^{-3}$), however, their effect is negligible. In networking scenarios with high loss probabilities the performance gain by control message retransmissions is considerable whereas the alternative endpoint or common control plays only a marginal role.

# References


[1] T. Chiueh, A. Neogi, and P. Stirpe, "Performance Analysis of an RSVP-Capable Router," in *4th Real-Time Technology and Applications Symposium*, 1998.

[2] M. Karsten, J. Schmitt, and R. Steinmetz, "Implementation and Evaluation of the KOM RSVP Engine," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2001)*, pp. 1290–1299, IEEE, Apr. 2001. ISBN 0-7803-7016-3.

[3] G. Fehér, K. Németh, and I. Czslényi, "Performance Evaluation Framework for IP Resource Reservation Signalling," in "$8^{th}$ *IFIP Workshop n Performance Modelling and Evaluation of ATM & IP Networks*", July 2000.

[4] P. Pan and H. Schulzrinne, "Processing Overhead Studies in Resource Reservation Protocols," in $17^{th}$ *International Teletraffic Congress*, December 2001.

[5] V. Eramo, U. Mocci, M. Fratini, and M. Listanti, "Reliability Evaluation of RSVP," in $8^{th}$ *International Telecommunication Network Planning Symposium (Networks 98)*, Okt. 1998.

[6] L. Berger, D.-H. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini, "RSVP Refresh Overhead Reduction Extensions." http://www.ietf.org/rfc/rfc2961.txt, April 2001.

[7] B. Braden *et al.*, “RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.” ftp://ftp.isi.edu/in-notes/rfc2205.txt, Sep. 1997.

[8] J. Wroclawski, “RFC2210: The Use of RSVP with IETF Integrated Services.” ftp://ftp.isi.edu/in-notes/rfc2210.txt, Sep. 1997.

[9] P. Pan and H. Schulzrinne, “Staged Refresh Timers for RSVP,” in $2^{nd}$ *Global Internet Conference*, Nov. 1997.

[10] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie, “Aggregation of RSVP for IPv4 and IPv6 Reservations.” http://www.ietf.org/rfc/rfc3175.txt, September 2001.

[11] E. C. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture.” http://www.ietf.org/rfc/rfc3031.txt, Jan. 2001.

[12] M. Menth, “A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks,” in $17^{th}$ *International Teletraffic Congress*, December 2001.

[13] M. Menth and N. Hauck, “Scalable QoS Transport in IP Using LSP Hierarchies,” Technical Report, No. 287, University of Würzburg, Institute of Computer Science, Nov. 2001.

[14] G. Fehér, K. Németh, M. Maliosz, I. Czslényi, J. Bergkvist, D. Ahlard, and T. Engborg, “Boomerang - A Simple Protocol for Resource Reservation in IP Networks,” in *”IEEE Workshop on QoS Support for Real-Time Internet Applications ”*, (Vancouver, Canada), June 1999.

[15] P. Pan and H. Schulzrinne, “YESSIR: A Simple Reservation Mechanism for the Internet,” in *”Computer Communication Review”*, vol. 29, April 1999.

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC1889: RTP - A Transport Protocol for Real-Time Applications.” ftp://ftp.isi.edu/in-notes/rfc1889.txt, Jan. 1996.

[17] P. Pan and H. Schulzrinne, “BGRP: A Tree-Based Aggregation Protocol for Interdomain Reservations,” *Journal of Communications and Networks*, vol. 2, pp. 157–167, June 2000.

[18] I. Stoica and H. Zhang, “Providing Guaranteed Services Without Per Flow Management,” in *”Computer Communication Review”*, vol. 29, October 1999.

[19] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, “SRP: a Scalable Resource Reservation for the Internet,” in *IFIP* $6^{th}$ *International Workshop on Quality of Service (IWQoS’98)*, May 1998.

[20] A. Eriksson and C. Ghermann, “Robust and Secure Light-weight Resource Reservation ofr Unicast IP Traffic,” in *IFIP* $6^{th}$ *International Workshop on Quality of Service (IWQoS’98)*, May 1998.

[21] L. Breslau, E. W. Knightly, S. Shenker, and H. Zhang, “Endpoint Admission Control: Architectural Issues and Performance,” in *”ACM SIGCOMM”*, August 2000.

[22] L. Breslau, S. Jamin, and S. Shenker, “Comments on the Performance of Measurement-Based Admission Control Algorithms,” in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM’2000)*, pp. 1233–1242, IEEE, March 2000. ISBN 0-7803-5880-5.

[23] I. Más and G. Karlsson, “PBAC: Probe-Based Admission Control,” in *”$2^{nd}$ International Workshop on Quality of future Internet Services (QofIS2001)”*, September 2001.