MODELLING AND IMPROVING FLOW ESTABLISHMENT IN RSVP

Laurent Mathy, David Hutchison and Steven Simpson
Lancaster University, UK
{laurent, dh, ss}@comp.lancs.ac.uk

Abstract

RSVP has developed as a key component for the evolving Internet, and in particular for the Integrated Services Architecture. Therefore, RSVP performance is crucially important; yet this has been little studied up till now. In this paper, we target one of the most important aspects of RSVP: its ability to establish flows. We first identify the factors influencing the performance of the protocol by modelling the establishment mechanism. Then, we propose a Fast Establishment Mechanism (FEM) aimed at speeding up the set-up procedure in RSVP. We analyse FEM by means of simulation, and show that it offers improvements to the performance of RSVP over a range of likely circumstances.

1. INTRODUCTION

It is now widely recognized that to become a global telecommunication platform with integrated services—a must in the provision of information super-highways—the Internet must evolve to provide proper support for applications, such as distributed multimedia applications, that require a variety of qualities of service. In an ideal world, this evolution should depend on the evolution of the traffic mix in the network (that is the ratio best-effort and guaranteed traffic). Unfortunately, the evolution of the traffic mix is very hard to forecast.

If best-effort traffic clearly dominates, then a well provisioned network, possibly enhanced with some simple form of traffic differentiation [3], can probably satisfy the occasional requests for Quality of Service (QoS) guarantees [6]. In other words, appropriate bandwidth is the key to QoS. On the other hand, if the proportion of guaranteed traffic becomes significant, more advanced resource management mechanisms are likely to be needed to meet the level of service expected by the users. One such mechanism considered here is resource reservation.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35580-1_16

J. D. Touch et al. (eds.), *Protocols for High-Speed Networks VI*

Recent studies [14] have found that in today's Internet, which is dominated by best-effort traffic, congestion occurs mainly at the edge of the network (e.g. in ISP access networks, links from campus networks, etc.). However, it has also been shown that some backbone links (especially some trans-continental links) are saturated for a substantial part of a day. These observations suggest that to support applications — such as interactive multimedia application or real-time applications — with stringent QoS requirements, resource management mechanisms will have to be provided at the edge of the network at least. This argument is reinforced by the fact that, as such applications appear, the traffic mix in the network may shift towards traffic requiring more resource usage control.

Among resource management mechanisms, those offering the finest grain of traffic control operate on a per-flow basis. These mechanisms, however, suffer from state scalability problems as the number of flows with reservations increases. Although this rules out their use within the core of the network, per-flow provisioning can still be used at the edge of the network where the concentration of flows is rather low. In the Internet, the IntServ (Integrated Services) architecture [4] offers a framework for QoS control which relies on RSVP ("Resource ReSerVation Protocol") [5][18] as the signalling protocol. Several proposals, mainly flow aggregation techniques [11][2] and the DiffServ (Differentiated Services) architecture [3], have been put forward to overcome the state scalability problems in the core of the network.

Although RSVP was originally designed for resource reservation, several proposals have now been tabled where RSVP is used to carry other types of control information in the network [13][8][10]. Another example is the possible use of RSVP within the DiffServ architecture [1]. Therefore, we believe that, whether it is for resource reservation or other control/signalling purposes, RSVP will have to operate over routes of various lengths and to satisfy demands exhibiting a broad range of dynamics. Consequently, RSVP's ability to carry control information efficiently across the network in any circumstances will be vital to the effective operation of the Internet.

That is why, in this paper, we study some of RSVP's performance aspects. The lack of experiments in "real conditions" leads us to develop, in section 2, a mathematical model of the flow establishment phase in RSVP. The results yielded by our model clearly show the need to revise the flow establishment procedure of RSVP. The principles of a modified flow establishment mechanism are then presented in section 3. Simulation results comparing the establishment procedure currently used in RSVP with our proposal are given in section 4. Some relevant related work is discussed in section 5, and section 6 concludes our discussion.

It should be noted that the primary context of resource reservation has influenced the naming of the control messages used in RSVP and it is therefore easier to describe the operations of RSVP in this context. The reader should however bear in mind that the results presented in this paper equally apply to RSVP as a "general" signalling protocol. Moreover, in this paper, we are only concerned with performance aspects of RSVP: scalability issues are not addressed.

The work presented in this paper is part of a wider effort at Lancaster University aimed at improving the support for distributed multimedia applications in the Internet, and specifically investigating the viability of resource management mechanisms.

2. MODELLING FLOW ESTABLISHMENT

Although its core ideas appeared a few years ago [18] and both research and commercial implementations are now available, to the best of our knowledge, no large-scale experiment has been done with RSVP yet. This lack of experimentation means that we do not know how RSVP will perform when used in "real conditions", as encountered in the Internet. In this section we develop a mathematical model of the establishment phase of RSVP in order to gain some insight of its performance. We are actually interested in quantifying RSVP's ability to make a successful reservation over a route where resources are plentiful. Although such a question may at first glance seem superfluous, we think it is of paramount importance to address it in order to assess RSVP's viability in the Internet, because of the unreliable character of the delivery of RSVP messages. In other words, we are interested in RSVP's external behaviour at reservation establishment as well as in dealing with network dynamics (local repair [5] may be seen as simply establishing a new reservation on a new portion of route).

In the rest of this section, we label as *sender* an RSVP node that initiate (as opposed to forward) the first Path message on a route where no (path) state has been established for the corresponding flow yet. A sender can either be an end-system (in the case of a reservation establishment) but could also be a router detecting a change of route (in the case of a local repair). We label as *receiver* an RSVP node that initiate (as opposed to forward) the first Resv message in response to the sender's Path message, that is on a "reverse route" where no reservation has been made for the corresponding flow yet. Again, the receiver can either be an end-system or a router. Any other node treating (i.e. creating state and reservation) and forwarding the messages along the route are

called RSVP routers. Although our model will be developed considering only one sender and one receiver, it is nevertheless applicable to the multicast case by applying it to the (sub-)branches of multicast trees.

We know that to establish a reservation for a flow:

1. the sender issues a Path message towards the receiver,
2. upon receipt of that Path message, the receiver issues a Resv message describing the resources required.
3. every node periodically[1] sends *its own* Path and Resv messages, that is there is no way to force a node to send copies of RSVP messages in the network. However, any control message inducing a change in the state of an RSVP node is immediately forwarded by this node.

The periodic messages (a.k.a refresh messages) serve as both error correction mechanism (there is no explicit acknowledgment in RSVP) and state/reservation management mechanism (the absence of too many consecutive refreshes result in state timeouts and removal of the corresponding states/reservations).

The central parameter in our model is p, the *per-hop success probability*, which is the probability that an RSVP message sent by an RSVP node is correctly received by the RSVP process in the next node. We therefore see that p takes into account not only transmission errors but also overflow conditions at the different levels of the protocol architecture (i.e. link, IP and RSVP layers). In a well dimensioned network, routers should be provisioned with enough resources to accommodate most of the control traffic. We therefore expect the value of p to be high (i.e. close to 1). Consequently, in our model, we will ignore state timeouts because such events occur with a probability $(1-p)^K \approx 0$ (with $K = 3$ by default [5]).

It is only when a Resv message reaches the sender that the reservation is fully established (i.e. considered successful). Furthermore, because we ignore state timeouts, if any of the RSVP messages is ever lost along the way, an equivalent message is re-emitted *from the last node where it was last correctly received* at the beginning of the next refresh period. The establishment thus appears to be "incremental": from a refresh period to the next, the number of nodes holding proper state/reservation for the flow cannot decrease. Therefore the refresh messages exchanged between nodes where the corresponding states/reservations have already been established have no influence on the rest of the establishment procedure and can thus be ignored. In other words, we always consider the control message which is "ahead" of the others. Such a message will be called "establishment message".

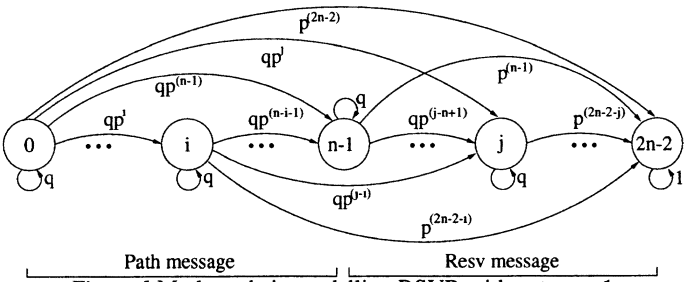


Figure 1 Markov chain modelling RSVP, with $p + q = 1$.

Another way to describe successful reservation establishment, is to note that on a route involving n RSVP nodes (including the sender and the receiver), the Path and Resv messages we consider (the establishment messages) must collectively travel $2n - 2$ hops. This is because the initial Path message is "generated" at the sender while the initial Resv message is "generated" at the receiver: these messages do not need to "travel" to reach these nodes.

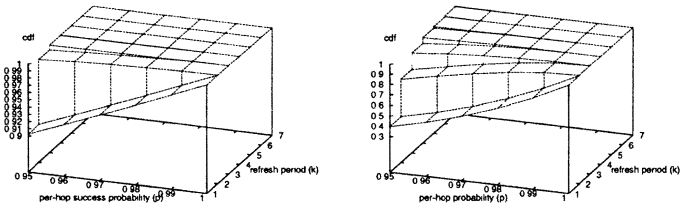
At any time, the "state" of a reservation establishment can be characterised by the number of hops travelled by the establishment messages, and the residual time, in the last RSVP node that correctly received an establishment message, until the next refresh for this message. However, if we consider the state of a reservation at the instants when the residual time until refresh becomes zero (i.e. at the end of the refresh period in the last node that received an establishment message), we have a *discrete time semi-Markov process* with $2n - 2$ states (see figure 1), whose embedded Markov chain (representing the process at the instants of state transitions) has the following transition probabilities:

$$p_{i,j} = 0 \qquad 0 \leq i < 2(n-1), \quad 0 \leq j < i \tag{1}$$

$$p_{i,j} = p^{j-i}(1-p) \qquad 0 \leq i < 2(n-1), \quad i \leq j < 2(n-1) \tag{2}$$

$$p_{i,2n-2} = p^{2(n-1)-i} \qquad 0 \leq i \leq 2(n-1) \tag{3}$$

Equation (1) is the mathematical expression for the "incremental" establishment simplification. Equation (2) and (3) are based on the fact that a transition from any state of the chain to any other (including itself), is equivalent to a control message travelling a number of hops equal to the distance between the states. Equation (2) simply expresses that if the state reached is not the last one, then the control message must have been lost between two nodes. On the other hand, equation (3) states than when the Resv message reaches the sender, no more control traffic is required. Also, note that the last state of the chain is *absorbing*, stating that the establishment of the reservation is complete. The set



2.(a): along 2 nodes 2.(b): along 10 nodes

Figure 2 Cumulative distribution functions for the success probabilities

of equations (1)–(3) unambiguously describes the *transition probability matrix* $\boldsymbol{P}$ of the Markov chain[2].

We now express RSVP's ability to make a successful reservation. Let S be the number of periods required to establish a flow with reservation ($S \geq 1$: the first Path message sent by the sender determines the start of the first period). In the context of the model, S is the number of transitions required to reach state $2n-2$.

Relations (1) and (3) imply that all the states but the last one are *transient*[3]. Also, because the last state of the chain is absorbing, that state is *recurrent*. In other words, the last state of the chain is eventually entered, and once it has entered it, the random process never leaves it. Consequently, the transient behaviour of $\pi_{2n-2}^{(k)}$—the probability of being in state $2n-2$ at the end of period k—can be interpreted as the probability that a reservation has been established by the end of the k^{th} refresh period:

$$\pi_{2n-2}^{(k)} = P[S \leq k] = F_S(k). \tag{4}$$

$\pi_{2n-2}^{(k)}$ is thus the cumulative distribution function (cdf) of the success probability P_S.

Figure 2 shows the values of $F_s(k)$ for different route length (n) in terms of different per-hop success probabilities (p) and various number of periods (k).

In figure 2, it clearly appears that, even for short routes, RSVP will perform reasonably well only for very high per-hop success probabilities. Indeed, the probability of success within the first period (i.e. establishment without message loss) is in accordance with (4) and (3):

$$P_S^{(1)} = P[S = 1] = \pi_{2n-2}^{(1)} - \pi_{2n-2}^{(0)} = p^{2(n-1)}, \tag{5}$$

where $\pi_{2n-2}^{(0)}$ is 0 because the chain is always started in state 0. $P_S^{(1)}$ is an important quantity because it expresses the chances that a reservation is established without any loss of control messages.

The behaviour of equation (5) when p is in the neighbourhood of 1, is given by:

$$\lim_{p \to 1^-} \frac{\partial}{\partial p} P_S^{(1)} = 2(n-1) \tag{6}$$

and we not only see that any variation of p results in a bigger variation of $P_S^{(1)}$ but also that for routes comprising 6 nodes or more, the variation in $P_S^{(1)}$ will be an order of magnitude bigger than the variation in p.

With this model, we can also derive the average number of refresh periods needed to establish a reservation:

$$E[S] = (2n-2)\frac{1-p}{p} + 1. \tag{7}$$

The previous result allows us to obtain the average contribution of the external behaviour of RSVP to the establishment time T of a reservation, or in other words, the average establishment time when the queueing, transmission, propagation and internal processing delays are neglected. As reservation establishment occurs at any time *within* a period and the first period starts with the very first Path message from the sender, we have:

$$E[T] \approx (E[S]-1)R = R(2n-2)\frac{1-p}{p}, \tag{8}$$

where R is the average refresh period. Both equations (7) and (8) confirm the sensitivity of RSVP to the values of the per-hop success probability.

The model presented in this section may seem a little pessimistic since we use the same value of the per-hop success probability on every link of a route. However, by extending the concept of an "RSVP node" to encompass the idea of a "lossless RSVP cloud", that is a contiguous region of the network where losses of RSVP messages do not occur or can be neglected, the model can be used to describe more realistic situations. For example, the scenario with two nodes can model a route of any length with one bottleneck, that is a route where control messages are only lost at one congested router.

As already pointed out at the beginning of this section, the value of the per-hop success probability p strongly depends on the rate of control traffic generated in the network and the associated resources needed to absorb such traffic. In RSVP, this rate of control traffic depends on both the rate at which

new reservation requests are issued (either by end-systems or following route changes) and the average number of existing flows with reservations (because of the periodic refresh associated with the soft-state). Although techniques have been devised to reduce the latter type of traffic [16], the results exposed in this section strongly suggest the need for a dedicated "signalling channel[4]" in order to keep the per-hop success probability as high as possible.

This is not only true to ensure good performance at flow establishment, but also to improve resource utilization in the network. Indeed, for a resource to be released within a short delay, a teardown message must travel the path followed by a flow *without being lost* [5]. It is so because any loss of a teardown message can only be corrected when a lifetime expires, which can take several minutes (see [5]) and thus induce inefficient resource utilization. For a route with n nodes (sender and receiver included), the probability of "immediate" release of the resources of a flow is p^{n-1}. This value shows that, although the release of resources is less sensitive to the value of the per-hop success probability than the establishment (see equation (5)), this sensitivity will nonetheless become a problem over medium-length or long routes.

As a consequence, for RSVP to give acceptable results as the signalling protocol of the Internet, a carefully provisioned signalling channel will be required. Obviously, in the parts of the network where RSVP will be deployed, such a channel will be built by reserving resources for the control traffic; in non-RSVP networks (connecting "RSVP clouds" together), mechanisms such as traffic differentiation [3] or prioritization will be required.

3. IMPROVING RESERVATION SET-UP

RSVP uses periodic messages to manage its states. The lapse of time between consecutive Path or Resv messages defines the refresh period of the protocol (in a refresh period, there is one Path and one Resv message per flow on each link of the path). The default value for the refresh period R is 30 seconds. From the results of our model presented in the previous section, such a lapse of time between similar RSVP messages seems prohibitively long, since it represents the average amount of time in which the loss of a control message can be corrected at reservation establishment. It therefore seems natural to reduce the length of the refresh periods to improve RSVP's performance at establishment time.

Simply reducing the value of the refresh period is not the right approach, however. Indeed, doing so would increase the control traffic associated with every flow, thus increasing the required capacity of the signalling channel of the

network while threatening to pose severe scalability problems. Consequently, reducing the refresh period *at establishment time only*[5] (including local repair conditions) is considered a better solution. In [5], it is suggested that a node could, at establishment, temporarily send control messages more often than dictated by the refresh period. However, the question of how many, as well as how often, such messages should be sent has not been addressed. This is precisely what we propose to do in this section.

In modern high speed networks, message losses are mostly due to buffer overflow. As a consequence, such losses occur in bursts [7]. We therefore see that proper "inter-spacing" is required between consecutive control messages, to prevent them from encountering the same congestion conditions along their route. This observation rules out the use of a fixed, short establishment period for the sending of consecutive RSVP messages during the establishment phase. Furthermore, in order to avoid unnecessary overhead, we must find a way to discover the end of the establishment phase, that is the moment after which the control messages related to a flow simply refresh the path states and reservations associated with that flow.

The only way to discover the end of the establishment phase of a flow is somehow to use the concept of acknowledgment. In order to keep our discussion as clear as possible and focus on principles, we present, in this section, a simple solution that only relies on the use of the Path and Resv messages, and hence does not require the introduction of explicit acknowledgment messages in RSVP.

It is clear that the role of an initial Path message is to "prepare" for a subsequent Resv message. A Resv message can therefore be considered as an acknowledgment for a Path message. This Resv message also indicates a successful reservation *to the sender of the corresponding Path message*. Therefore, any node that has forwarded a Path message, and has received a Resv message from every direct neighbour down the route followed by the corresponding flow, knows that the reservation has been successfully established *downstream*.

We still need to find a way for the receiver of a Path message to discover whether the establishment of a flow is in progress or has been completed. Because upstream nodes will use establishment periods shorter than the refresh period as long as they have not received a proper Resv message, a node can guess the status of a flow from the spacing of the Path messages it receives: if the lapse of time between consecutive Path messages is smaller than the shortest lapse of time allowed in "steady state" (that is $R/2$, see [5]) then the flow is more than likely being established and a Resv message should be forwarded as soon as possible to complete the establishment procedure (we thus see that the Resv message will be re-transmitted by the last RSVP node that

correctly received the previous Resv message). On the other hand, if the time between consecutive Path messages is greater than or equal to the minimum allowed by the "classical" refresh periods then we can suspect that the Path message is simply a refresh and a Resv message should only be sent when the current refresh period expires[6]. Of course, for this technique to be robust in the event of loss of Path messages, the periods used at establishment time must be quite a lot smaller than $R/2$. To be precise, the difference between $R/2$ and any establishment period should be at least an order of magnitude larger than delay variations in the network.

It should be noted that if a node makes a wrong "guess" about consecutive Path messages, the corresponding flow does not suffer any functional damage. In the unlikely event where losses or delay variations cause two consecutively received establishment Path messages to be interpreted as refresh messages, no Resv message is sent. In such a case, FEM simply misses a chance to send a Resv message, which in the worst case, will be re-transmitted as a "classical" refresh. Furthermore, if important delay variations, or routing loops, cause refresh Path messages to be interpreted as establishment messages, a Resv message is sent. This Resv message resets the soft-state timer in the upstream node, but is not propagated any further.

If a reservation fails due to a lack of resources along the path, it may be wise to cancel FEM for the corresponding flow to prevent potentially unnecessary control traffic, although this is not an absolute necessity. The ResvErr message sent [5] may be used to cancel FEM in downstream nodes, while to cancel FEM in upstream nodes, a Resv message containing an empty reservation may be sent.

We have already ruled out the use of fixed periods at establishment. The other important point is that, if the establishment periods are too short, unnecessary RSVP messages will be sent, which increases the overhead of the protocol. Therefore, the initial establishment period (T_0) should not be smaller than the round-trip-time (RTT) for the RSVP messages, which may have to be estimated.

After sending or forwarding the initial Path message, an RSVP node will wait for a lapse of time equal to the initial establishment period (T_0). If by that time a Resv message has not been received, the node suspects a loss of control messages and retransmits the Path message (this procedure is applied by all the nodes supporting our technique, so that the copy of the Path message is generated as close as possible to where the loss of the previous RSVP message occurred). In order to be adaptive to a wide range of congestion conditions, the value of the establishment period must be backed-off: we propose to multiply it by a factor $(1 + \Delta)$ at each retransmission of a Path message. As soon as

a Resv message acknowledges the establishment of the reservation, the nodes start using the refresh period R for their Path messages. A refresh period equal to R is also used if no Resv messages has been received, but the value of the establishment period has become greater than R. We therefore see that, in any case, the nodes "fall back" to the behaviour prescribed by the "classical" RSVP specification. We therefore see that FEM RSVP is backward compatible with "classical" RSVP.

With T_0 set to 3 seconds and Δ set to 0.3, this timer scheme is equivalent to the staged refresh timers described in [15]. It should be noted that for local repairs, a shorter value of T_0 would be acceptable, since we expect the new portion of the route to be fairly short. Furthermore, such a more aggressive behaviour of the protocol is justified by the fact that local repairs apply to existing flows.

The simple solution presented above requires that a receiver be able to send a Resv message immediately on receipt of a Path message. Although it will always be so in the case of local repairs, the reservation requirements might not be readily available if interaction with the end user is needed to determine these requirements. In this latter case, the solution proposed here would result in much unnecessary overhead and would fail to correct swiftly the loss of a Resv message. One way to overcome such a problem would be to define acknowledgment message for Path messages (e.g. PathAck) and Resv messages (e.g. ResvAck). An immediate Resv message would be generated at a receiver whenever possible (and FEM would be applied as presented in this section), otherwise an immediate PathAck would be sent and FEM applied on Path-PathAck pairs. As soon as the reservation requirements would be known, a Resv message would be sent and the FEM mechanisms could be applied, in the "reverse" direction, on Resv-ResvAck message pairs. Using FEM separately on Path and Resv messages would then ensure prompt recovery from losses of control messages.

In the multicast case, two strategies can be adopted for FEM. On one hand, as soon as the first Resv message is received by a node, that node forwards it upstream without delay. This has the advantage of quickly propagating reservations along the multicast tree. However, although any Resv message subsequently received by the node and that increases the reservation demands would be immediately forwarded upstream (according to the message forwarding rules of RSVP [5]), losses of such messages would not be corrected by FEM but by later refreshes. In such a case, FEM speeds up the initial establishment but cannot reduce the latency of increasing reservation demands. On the other hand, a node could hold the reservation requirements received in Resv messages either for a small lapse of time or until it has received Resv/PathAck

messages on all the output ports of the multicast tree, before it forwards its own Resv message upstream. This has the advantage of establishing the final reservation at once, but has the risk of potentially increasing the overall establishment latency. Further work is needed to study and evaluate these possible strategies in the multicast case.

Finally, in order to avoid unnecessary overhead, FEM RSVP nodes should try and discover the capabilities of their neighbours (this could be done by recording the protocol version in the received messages) and refrain from using FEM when the next hop node does not support it. Furthermore, in multicast, the usual state/message merging should be applied.

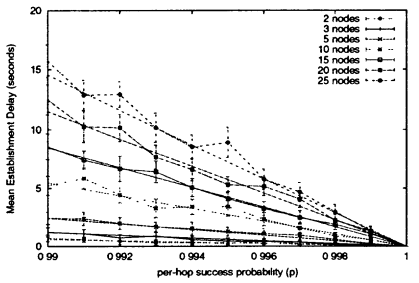
4. SIMULATION RESULTS

We have simulated the external behaviour of both "classical" and FEM RSVP, in order to compare them. Our simulations consisted of repeated reservation establishments between a sender and a receiver, over routes of various lengths and under distinctly different loss conditions.

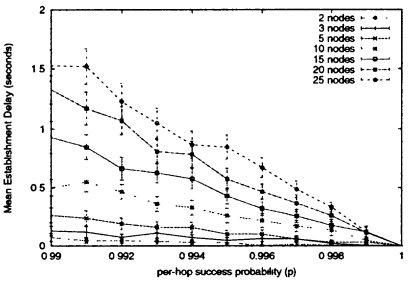
In these simulations, the loss process on each direction of a "link" is represented, independently, by a two-state model. One of the states represents congestion (i.e. loss) periods while the other one represents no-loss periods. The loss process spends an exponentially distributed time in each state, with these exponential distributions set so that the mean congestion period is 200 ms and the loss process spends a long-term proportion of time equal to the per-hop success probability in the no-loss state. Such a model was chosen because of its ability to mimic loss bursts in a simple way.

Configurations comprising respectively 2, 3, 5, 10, 15, 20 and 25 nodes (including the sender and the receiver) were considered with values of the per-hop success probability ranging from 99% to 100% inclusive. Such values for the per-hop success probability were chosen because they are likely to be encountered in a well dimensioned network. For every configuration, 1000 flows were established and no delay was introduced in nodes and links to isolate the time overhead introduced by the external (i.e. observable) operation of the protocol. Finally, the default of 30 s was used as the average value of the refresh periods in RSVP, while for FEM RSVP, T_0 and Δ have the values proposed in section 3.

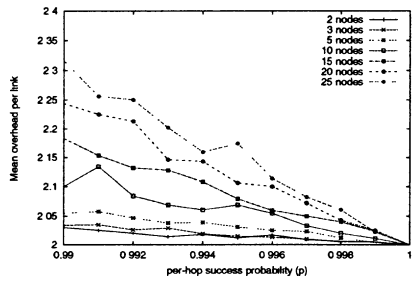
The measured quantities were the mean establishment delay and the mean overhead per link (i.e. the mean number of control messages per link per reservation). For the mean establishment delay, 95% confidence intervals were computed using the method of batch means [12, p. 293] on 40 batches of 25



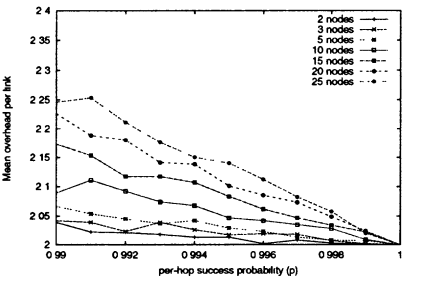
3.(a): MED with RSVP. Simulation (jagged lines) and theoretical (straight lines—eq. (8)) results.



3.(b): MED with FEM RSVP.



3.(c): MOPL with RSVP.



3.(d): MOPL with FEM RSVP.

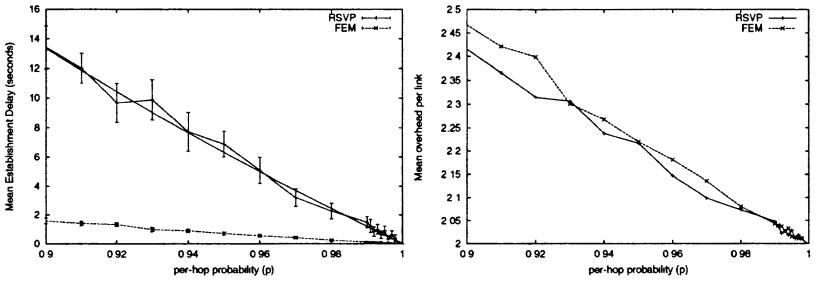
Figure 3 Mean Establishment Delay (MED) and Mean Overhead Per Link (MOPL).

samples each. The results are given in figure 3. In interpreting the results in this section, special attention must be paid to the meaning of the per-hop success probability which essentially represents the chances of survival of a control message from one RSVP process to the next. Therefore, the corresponding per-hop loss probability (i.e. the probability that a control message does not reach the next RSVP process) is expected to be greater than usual packet loss probabilities, because it encompasses possible losses due to overflows of the queue holding messages awaiting to be treated by the RSVP process which usually resides in the slow path of a router.

A part from the obvious gain in performance, figure 3 also confirms the more predictable (or more stable) behaviour of FEM RSVP (the 95% confidence intervals are about an order of magnitude smaller in FEM RSVP than in RSVP). The message overhead (figures 3.(c) and 3.(d)) is fairly similar in both

cases. There is however a slight trend showing a better effectiveness of FEM as reliability decreases. This property could prove very valuable in the case of local repairs, where bursts of repair messages could result in congestion of the signalling channel (including queues to the RSVP processes in routers).

Figure 3.(a) validates the predictions of our mathematical model, despite fundamental differences in the loss processes assumed in section 2 and these simulations. Figure 4, by contrast, specifically compares RSVP and FEM



4.(a): Establishment Delay. Simulation (jagged lines) and theoretical (straight line—eq. (8)) results.

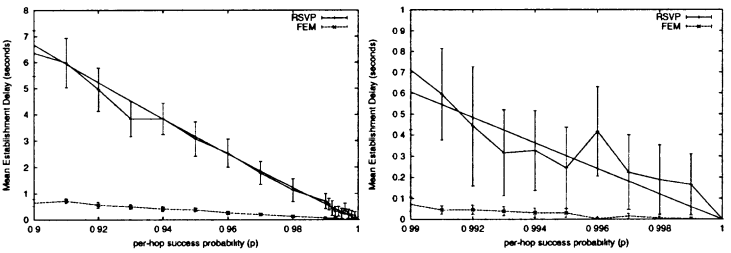
4.(b): Message Overhead.

Figure 4 Comparison of RSVP and FEM over routes of 3 nodes in a wireless scenario.

RSVP over routes of 3 nodes with per-hop success probabilities ranging from 90% to 100%. This scenario is important since it represents a case where the sender and receiver are wireless terminals, with the wired network in-between considered lossless. Again, the stability and gain in performance of FEM, with no significant increase in message overhead, are clearly demonstrated.

Another equally important scenario is the case of a route of any length with a single bottleneck. As mentioned in section 2, such a scenario is modelled as a route comprising 2 nodes connected by a lossy link. The mean establishment delay is given in figure 5, for different ranges of per-hop success probabilities. In this case, although the mean time penalty introduced by the external behaviour of RSVP is quite small and would not be considered unacceptable as long as the message loss probability does not exceed a percent. Once more, this confirms RSVP's sensitivity to the values of the per-hop success probability and their variations, even when only facing a single point of congestion.

Finally, in view of the results presented in this section, it could be argued that if the per-hop success probability was kept very close to 1, the performance of RSVP would be satisfactory, and hence the FEM extension would not



5.(a): "Low" per-hop success probabilities at bottleneck. Simulation and theoretical (eq. (8)) results.

5.(b): "High" per-hop success probabilities at bottleneck. Simulation and theoretical (eq. (8)) results.

Figure 5 Comparison of RSVP and FEM over routes with a single bottleneck.

be necessary (especially over short routes). It should however be noticed that our results consist of mean values, averaged over a large number of flows and that on any particular occasion, the loss of any control message at flow establishment, is penalised by a delay of *at least* $R/2$ seconds (i.e. 15 seconds by default) with RSVP, but only by a delay of at least T_0 seconds (i.e. 3 seconds in the context of our simulations) with FEM RSVP. This fact alone probably justifies the use of FEM RSVP, even when the probability of losing a control message is extremely low. Finally, because losses of several control messages within a reservation establishment result in a set-up latency of several seconds, a signalling channel may still be necessary with FEM to ensure an acceptable establishment latency to each flow.

5. RELATED WORK

To the best of our knowledge, few other authors have reported on work closely related to our own. In [15], it is proposed to define and use explicit hop-by-hop acknowledgment messages for every control message in RSVP. To improve the responsiveness of the protocol, a procedure similar to the one described in section 3 is used for the retransmission of the control messages that have not been acknowledged. Once states or reservations have been acknowledged, it is then proposed to use long refresh periods (of the order of quarter of an hour) in order to reduce the steady state overhead.

Although this approach seems similar to ours, there is a major difference in the use of acknowledgments: the acknowledgments are used hop-by-hop. A

node that has correctly received a message from one of its neighbours acknowledges it. Therefore, the semantic of these acknowledgments is weak, because the receipt of an acknowledgment does not mean that the initial message has reached, or will reach, its final destination. Finally, the long refresh periods will result in performance far worse than the one of "classical" RSVP in the following circumstances: path state instability after route changes or transient failures undetected by the routing protocol. Furthermore, because once a reservation has been established the subsequent refresh messages are not acknowledged, losses of such messages can result in a loss of state (due to soft state time-out) that will be unacceptably long to correct.

In contrast, FEM RSVP is based on end-to-end notification, which covers the conditions cited above. Furthermore, as outlined in section 3, FEM RSVP can avoid the use of explicit acknowledgment messages when reservation requirements are readily available. As this is always the case for local repairs, FEM RSVP helps in reducing the size of the message bursts that occur in those circumstances.

Compared with signalling protocols used in ATM networks [17], "RSVP-like" signalling protocols (including FEM) do not rely on underlying reliable protocols for the transfer of their messages. The design of these reliable protocols (e.g. the use of sequence numbers) is such that they not only correct losses, but can also detect node failures (e.g. when too many consecutive error correction attempts fail). As a consequence, although ATM signalling messages are acknowledged hop-by-hop, their semantics is strong. It is precisely this lack of node failure that led us to the use of end-to-end acknowledgment in FEM (as FEM does not introduce sequence numbers in RSVP).

6. CONCLUSIONS

We have modelled the resource reservation establishment mechanisms in RSVP and have shown that it is very sensitive to the values of the per-hop probability measured between RSVP processes. We have also shown that, to a lesser extent, this sensitivity affects resource release too. Consequently, there is a need for a signalling channel in the Internet, to protect as much as possible the value of the per-hop success probability experienced by RSVP messages from being adversely influenced by data traffic. Furthermore, because even the best provisioned signalling paths are never totally lossless, we have presented the principles of FEM, a Fast Establishment Mechanism that is not only more robust to the conditions in the network than the establishment mechanism currently used in RSVP, but also establishes resources faster in any circumstances.

In the case of local repairs, FEM can even achieve better performance without any increase of message overhead.

FEM introduces a slight increase in protocol state. However, we anticipate that RSVP will only be operated on a "per-flow" basis in areas of the Internet (in particular at the edges) where the concentration of flows is low. Elsewhere, RSVP will be operated in an aggregation context which greatly reduces the state scalability problem. Consequently, the small state increase in FEM RSVP should be of little consequence.

Finally, the underlying principles of FEM are very simple. The resulting modest increase of protocol complexity is negligible compared with the achieved gains in performance.

REFERENCES

[1] Y. Bernet, J. Binder, S. Blake, M. Carlson, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss. A Framework for Differentiated Services. Internet Draft draft-ietf-diffser-framework-00, IETF, May 1998. Work in Progress.

[2] S. Berson and S. Vincent. Aggregation of Internet Integrated Services State. Internet Draft draft-berson-rsvp-aggregation-00, IETF, Aug 1998.

[3] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Internet Draft draft-ietf-diffserv-arch-00, IETF, May 1998. Work in Progress.

[4] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, IETF, Jun 1994.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification. RFC 2205, IETF, Sep 1997.

[6] L. Breslau and S. Shenker. Best-effort versus Reservations: a Simple Comparative Analysis. *ACM Comp. Comm. Reviews*, 28(4):3–16, Sep 1998.

[7] I. Cidon, A. Khamisy, and M. Sidi. Analysis of Packet Loss Processes in High Speed Networks. *IEEE Trans. Info. Theory*, 39(1):98–108, Jan 1993.

[8] B. Davie, Y. Rekhter, E. Rosen, A. Viswanathan, V. Srinivasan, and S. Blake. Use of Label Switching with RSVP. Internet Draft draft-ietf-mpls-rsvp-00, IETF, Mar 1998. Work in Progress.

[9] S. Floyd and V. Jacobson. The Synchronization of Periodic Routing Messages. *IEEE/ACM Trans. Network.*, 2(2):122–136, Apr 1994.

[10] O. Fourmeaux and S. Fdida. Multicast for RSVP Switching. *Telecommunication Systems Journal*, 11(1-2):85–104, Mar 1999.

[11] R. Guérin, S. Blake, and S. Herzog. Aggregating RSVP-based QoS Requests. Internet Draft draft-guerin-aggreg-rsvp-00, IETF, Nov 1997. Work in Progress.

[12] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, second edition, 1994.

[13] T. Li and Y. Rekhter. A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE). RFC 2430, IETF, Oct 1998.

[14] A. Odlyzko. The Internet and Other Networks: Utilization Rates and Their Implications. In *Proc.* 26th *Telecommunications Policy Research Conf.*, Oct 1998. Available from www.research.att.com/~amo/doc/networks.html.

[15] P. Pan, H. Schulzrinne, and R. Guérin. Staged Refresh Timers for RSVP. Internet Draft draft-pan-rsvp-timer-00, IETF, Nov 1997. Work in Progress.

[16] P. Sharma, D. Estrin, S. Floyd, and V. Jacobson. Scalable Timers for Soft State Protocols. USC CS-TR 96-640, University of Southern California, Sep 1996.

[17] B. Stiller. A Survey of UNI Signaling Systems and Protocols for ATM Networks. *ACM Comp. Comm. Reviews*, 25(2):21–33, Apr 1995.

[18] L. Zhang, S. Deering, D. Estrin, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, 7(5):8–18, Sep 1993.

Notes

[1] Each value of the refresh period is randomly chosen in $[R/2, 3R/2]$, with $R = 30$ sec. by default [5], to avoid message synchronisation [9]

[2] It is easy to verify that $\sum_{j=0}^{2(n-1)} p_{i,j} = 1, \quad 0 \leq i \leq 2(n-1)$.

[3] Indeed, for $0 \leq i < 2(n-1)$, we have $\sum_{n=1}^{\infty} p_{i,i}(n) = \sum_{n=1}^{\infty} (1-p)^n = \frac{1-p}{p} < \infty$.

[4] This signalling channels includes the RSVP processes in the routers, and hence the associated queues.

[5] Such shortened refresh periods are called *establishment periods* in the rest of the paper.

[6] The period used by a node to send Resv messages is the refresh period defined in "classical" RSVP. The concept of establishment period timer does not apply to Resv messages.