Performance Comparison of the RSVP and Boomerang IP Resource Reservation Protocols

Simon Littlejohns

University of Wales, Swansea

Abstract: *This paper covers a simulation comparison of two resource reservation protocols: RSVP, the leading protocol in this field and Boomerang, a relative newcomer. Boomerang's designers claim that it outperforms RSVP under certain conditions. These claims were put to the test.*

1. Introduction

The leading protocol for providing Quality of Service (QoS) guarantees is presently the Resource Reservation Protocol (RSVP) [1], which has been available over selected networks for several years now. It is effective, but its complexity causes it to suffer from a lack of scalability when deployed across large networks handling tens of thousands of flows.

A number of alternatives have been suggested that aim to provide a simpler and therefore scalable solution to the problem. One such attempt is called Boomerang [2]. Its designers have come up with a reservation mechanism that is aimed at a different type of application to RSVP, and so in some respects they are not trying to compete directly. Despite this, RSVP has to be seen as a benchmark for QoS protocols, as it is the only one with any real deployment experience.

Boomerang's designers have pitched it against RSVP using analytical and simulation comparisons [3]. As well as being more scalable, they claim that Boomerang outperforms RSVP under conditions of frequently changing reservation loads. This paper describes work that was undertaken to put this claim to the test using the OPNET [4] network modelling software.

2. Background

RSVP was designed primarily with multicast communication in mind in order to serve many users with few sessions, thus making efficient use of valuable resources. It is therefore receiver-oriented which forces receivers to take care of their own reservations, giving them greater flexibility. In general, a source will send out path-finding messages to all potential receivers in a selected multicast group. Those wishing to receive will send reserve messages back to the source, and if possible, the intermediate routers will reserve the appropriate resources [5] (figure 1).

Boomerang is the latest reservation protocol to appear and its designers have taken a radical new approach in terms of reservation set-up procedure. For some time there has been a question mark over the validity of the multicast leanings of RSVP, as it seems that the majority of desired reservations are not of this nature. Most users would prefer to see a service offering bi-directional unicast reservations for use in Internet Telephony for example. This is therefore, the primary type of reservation offered by Boomerang, although it can also support multicasting.

Under the protocol, an initiating node (IN) will send a reservation request across the network to the far end node (FEN), which will then bounce the message back to the IN, possibly on different paths (figure 2). Incidentally, this is where the protocol acquires its name and the control message is often talked about as a “boomerang”. The IN and FEN terms are introduced because neither node is strictly a sender or receiver as all the reservations are bi-directional.

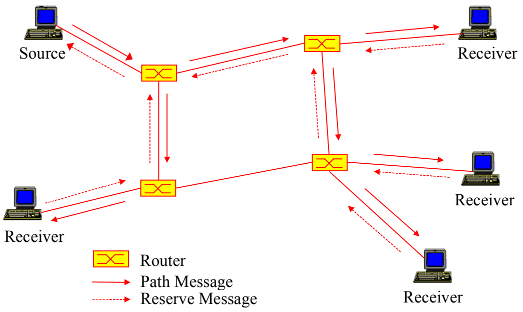

Figure 1. RSVP Signalling.

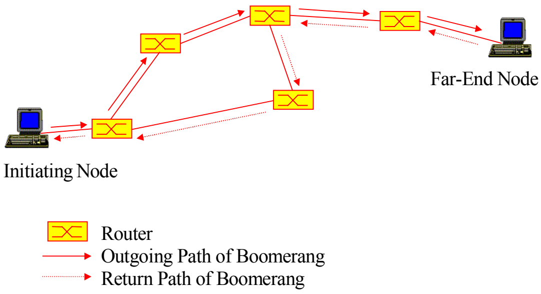

Figure 2. Boomerang Signalling.

3. Simulations

The OPNET models for both RSVP and Boomerang shared a common topology as shown in figure 3. In order to ensure that there was at least on area of congestion, 2 routers were incorporated with a single link between them. This naturally formed a bottleneck.

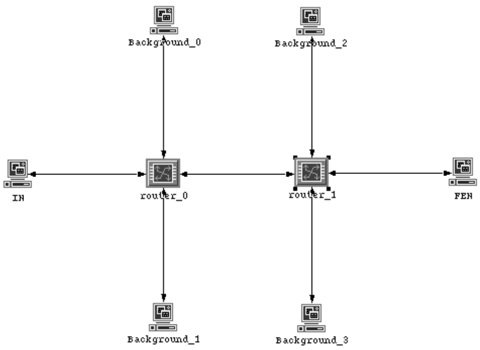

Figure 3. Network Topology.

To expect to see any performance difference between RSVP and Boomerang, there needed be a fairly large number of background flows to ensure congestion with a reasonable number of small gaps between connections. The more frequent the gaps, the shorter the overall simulated time could be, which was important as the real time taken to run the simulation was much longer. For this reason, each background node contained 20 potential sources. As connections needed to be bi-directional, the 80 potential sources gave rise to 40 potential background connections.

The connection set up between the IN and FEN was also bi-directional. This was fine for Boomerang, with its bi-directional mechanism, but RSVP required 2 sessions to perform the same task.

Both the software and hardware being used to perform the simulations, although perfectly functional, were far from 'state-of-the-art'. For this reason they differed from the original simulations performed in [3], in terms of scale. In [3] a network with 10 routers, 100Mbps links and comparatively long periods of simulated time is talked about. Fortunately, the network used in figure 3 could be smaller, partly because the actual length of the reservations was not a critical feature. The period of interest in all simulations was the point at which reservation requests were made. The simulations could therefore be greatly shortened by cutting down the time periods in between.

All the links in figure 3 were duplex links with a bandwidth of 10Mbps. All the sources generated MPEG video at an average rate of 1.5Mbps and all hosts would try to reserve 1.5Mbps per source. Each individual background source could be declared active or inactive for the duration of a simulation and would always use the same protocol as the IN and FEN for requesting reservations.

4. Results

Results were measured in terms of blocking probability, i.e. the likelihood of the IN being unsuccessful in its attempts to set up a reservation with the FEN. The value was calculated using equation 1 and the results can be seen in figure 4.

$$\text{Blocking Probability} = \text{Failures} / (\text{Successes} + \text{Failures}) \qquad \text{Equation 1.}$$

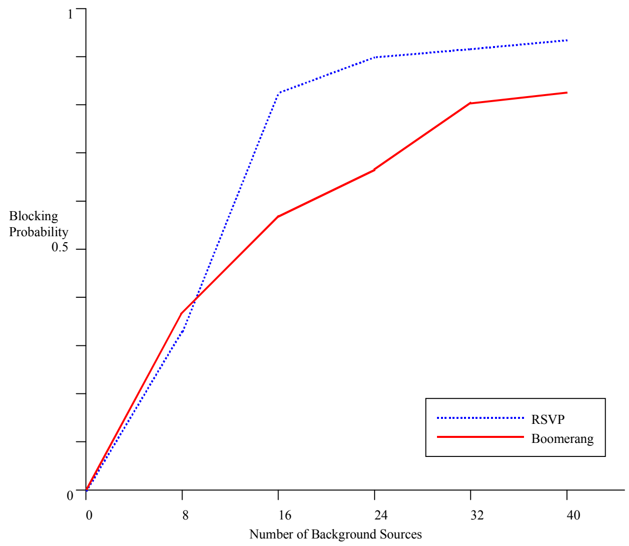

Figure 4. Graph of Blocking Probability against Number of Active Background Sources.

It can be seen from figure 4 that in this experiment, Boomerang experienced a lower probability of having its reservation requests blocked for most levels of background activity.

5. Conclusions

There are differences in performance between RSVP and Boomerang in certain situations. For bi-directional reservations in conditions where resources are in demand but are not unobtainable, then RSVP has a higher chance of having its requests blocked. This is not because other hosts are reserving the available resources however. The simulations showed that even when all sources have the same chance of success or failure, RSVP still fairs worse than Boomerang. The reason stems from the fact that RSVP has a much longer and more complicated reservation set-up procedure. Consider the following example:

Host A manages to get half way through its set-up process, i.e. it has reserved resources in one direction only. Shortly afterwards, host B completes half of its set-up process, but it does this before host A has completely finished. The remaining control messages from host A then try to complete the set-up by reserving resources in the reverse direction. Unfortunately, the last such resources available have just been reserved by host B. To compound the situation, before host A can tear down the reservation that it has made, this reservation succeeds in blocking the second half of host B's reservation requests. Hosts A and B have effectively blocked each other out by their ill-timed requests.

This is just an example and there are other possible variations. The end result is one whereby resources that are available for reservations are simply wasted. The same can happen in Boomerang, but as Boomerang has much shorter reservation set-up times, the chances of this kind of collision are greatly reduced. This is why Boomerang has a lower blocking probability.

An interesting point revealed by this research is that the advantage exhibited by Boomerang can be experienced in much smaller networks carrying far less traffic than those used in [3]. As a consequence of this, Boomerang potentially has a larger advantage over RSVP in these situations than its designers had expected. The large-scale simulations that were carried out in [3] achieved smoother graphs, but are not needed to show the general trend.

It should be noted however, that this kind of blocking was made possible here by the use of bi-directional reservations. Boomerang may still have a performance advantage in other situations due to its simplicity, but it could be significantly less. It should also be noted that RSVP was not primarily designed for this kind of set-up. One of its biggest strengths lies in its multicasting facilities and this is one aspect that has not been tested here or in [3].

References.

[1] B. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", IETF, September 1997.

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

[3] G. Feher, K. Nemeth, I. Cselenyi, "Performance Evaluation of the Boomerang IP Resource Reservation Protocol", InfoComm 2000.

[4] MIL 3, "OPNET Modeler", Software Release 3.0B, 1997.

[5] R. Braden. L. Zhang, "Resource ReSerVation Protocol (RSVP) – Version 1 Message Processing Rules", IETF, September 1997.