Performance Evaluation of the RSVP Reservation Aggregation Model

Rui Prior[1], Susana Sargento[1,2], Pedro Brandão[1], Sérgio Crisóstomo[1]

[1]DCC-FC & LIACC, University of Porto, Portugal
[2]Institute of Telecommunications, University of Aveiro, Portugal
`{rprior, ssargento, pbrandao, slc}@ncc.up.pt`

Abstract. This paper contains an evaluation of the RSVP Reservation Aggregation architecture, proposed by the IETF as a scalable alternative to the standard RSVP/IntServ for usage in high-speed core networks. We point out its main strengths, weaknesses and limitations, and describe our implementation of the architecture in the ns-2 simulator, including the definition of policies which are considered out of the scope of RFC3175, of which the most important is the aggregate bandwidth management policy.
The simulation results confirm that the architecture is able to meet the QoS requirements of a controlled load service class with much lighter classification, forwarding and signalling procedures than RSVP/IntServ. They also demonstrate that the scalability comes at the price of a lower utilization of network resources. We further provide some guidelines for setting the tunable parameters, bulk size and hysteresis time, based on the analysis of the simulation results.
Keywords: QoS, scalability, RSVP, aggregation.

1 Introduction

The IETF has proposed two main architectures aiming at the introduction of quality of service (QoS) support in the Internet. The Integrated Services (IntServ) architecture [1] uses per-flow reservations, through the Resource ReSer-Vation Protocol (RSVP) [2], and provides strict QoS guarantees and efficient resource usage. It has, however, several scalability problems, concerning the per-flow scheduling, classification and reservation procedures. The Differentiated Services (DiffServ) architecture [3] does not suffer from scalability problems: there are no per-flow resource reservations, flows are aggregated in classes according to specific characteristics, and services have a different treatment according to their class. However, without admission control mechanisms to limit the number of flows in the network, all flows belonging to a class may see their service degraded by influence of other flows.

With the objective of benefiting from the virtues of both IntServ and DiffServ and mitigating their problems, several architectures have been proposed in the literature. One of the most promising is the RSVP Reservation Aggregation (RSVPRAgg), defined in [4], based on the aggregation of end-to-end per-flow reservations, using an extension of the RSVP protocol that allows end-to-end RSVP signalling messages to be hidden inside an aggregation region. In the simplest case, aggregate reservations are performed between all ingress and egress routers of a network domain. These reservations are updated in bulks much larger than the individual flow's bandwidth. Whenever a flow requests admission in an aggregate region, the edge routers of the region check if there is enough bandwidth to accept the flow on the aggregate. If resources are available, the flow will be accepted without any need for signalling the core routers. Otherwise, the core routers will be signalled in an attempt to increase the aggregate's bandwidth. If this attempt succeeds, the flow is admitted; otherwise, it is rejected.

The scalability of this model stems from (1) the much lighter packet classification and scheduling procedures, (2) the reduced amount of state stored at the interior nodes and (3) the lower number of signalling messages processed at these nodes. Its main disadvantage is the underutilization of network resources. Since the bandwidth of each aggregate is updated in bulk quantities, each aggregate's bandwidth is almost never fully utilized. The unused bandwidth of all aggregates traversing a link adds up, leading to a significant amount of wasted link capacity.

In this paper, an evaluation of the aggregation model is performed, based on our implementation of the model in the ns-2 network simulator. Section 2 describes the implementation in more detail, namely regarding the bandwidth management policy for aggregates. Some particularities of our implementation and limitations of the model are also discussed in this section. A performance evaluation, based on simulation results, is presented in section 3. We analyze the standard QoS parameters (delay, jitter and packet loss ratio), as well as other parameters relevant to the performance and scalability of the architecture, such as network resource utilization and the number of signalling messages processed at core nodes, and compare them to those obtained with the standard RSVP/IntServ architecture in similar conditions. The results show that while RSVPRAgg is able to meet the QoS requirements of a controlled load class in a scalable way, it suffers from underutilization of network resources. With these simulations we also evaluated the influence of some tunable parameters: the bulk size and the hysteresis time. Based on the results we derive some guidelines for setting these parameters. Finally, section 4 presents the main conclusions and points out some topics for further work.

2 Implemented solution

The RSVPRAgg model was implemented in the ns-2.26 network simulator as an extension to an existing implementation of the RSVP protocol by Marc Greis. This section describes the aggregate bandwidth management policy we used. It also describes some implementation particularities and some limitations of the aggregation model and its specification.

2.1 Aggregate bandwidth policy

Although in [4] no actual policy for aggregate bandwidth management is defined, since it is considered out of the scope of the document, some guidelines are provided. In particular, it is stated that the aggregates' bandwidth should be modified infrequently, and that some hysteresis should be used in order to avoid oscillations in stable conditions. Figure 1 illustrates the implemented bandwidth policy, described in the next paragraphs. The aggregate bandwidth is plotted along with the sum of the reservations belonging to the aggregate.

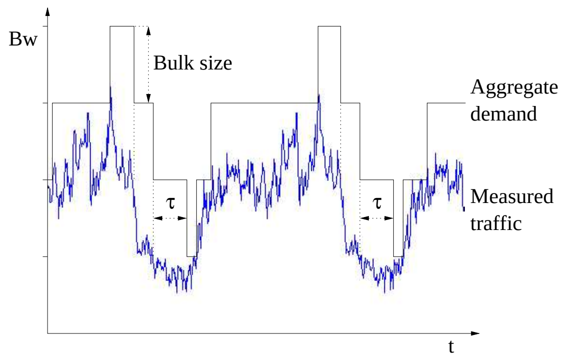


Fig. 1. Aggregate bandwidth management

Bandwidth updates for aggregates are always performed in multiples of a bulk. The bulk size is configurable, and should be set to a value much larger than the individual flows' rates. Bandwidth increase for aggregates is performed on demand, i.e., when a new end-to-end reservation request arrives at the deaggregator, it is assigned to a certain aggregate and there is no bandwidth to accommodate the new flow on that aggregate. Since we are dealing with simulation, the definition of rules to predictively estimate traffic patterns and perform bandwidth management accordingly would not be as meaningful as in the case of real networks with actual customer traffic over large time spans. Though it may lead to an increased reservation setup delay, this reactive policy tends to increase network utilization, since it leaves more bandwidth available for other aggregates which will hold actual traffic. The sole exception to the reactive policy rule happens at the creation time of a new aggregate. Since it is triggered by the reception of a Path message, odds are that a request for a reservation assigned to the new aggregate will soon be received. By predictively allocating some bandwidth (one bulk) to the aggregate, it is possible to reduce the setup time for that end-to-end reservation.

Bandwidth reduction for aggregates is not performed immediately when it ceases to be needed. Instead, it is delayed until the excess bulk has not been in use for a certain, configurable, time period τ. This hysteresis mechanism is intended to avoid unnecessary message processing at the interior nodes by successively increasing and decreasing the bandwidth in a stable operating point around a multiple of the bulk size. If, at a certain instant, the wasted bandwidth of an aggregate exceeds two bulks, though, bandwidth reduction is performed immediately, leaving only one excess bulk and restarting the hysteresis timer.

In order to avoid repetitively trying to increase an aggregate's bandwidth without success, leading to unnecessary message processing at the core (interior) nodes, a configurable hold time was also implemented during which no aggregate bandwidth increase will be tried in response to the arrival of a new end-to-end reservation request assigned to that particular aggregate. During that time period, new end-to-end reservation requests will either be accepted immediately, which may happen if other flows belonging to the same aggregate were terminated leaving some bandwidth available, or they will be rejected.

2.2 Particularities and limitations

One limitation of the aggregation model is related to the Guaranteed Service (GS). Due to the hard bounds on delay provided by this class, GS flows sharing the same aggregator and deaggregator nodes cannot be assigned to the same aggregate. Instead, they must be partitioned into a set of aggregates, each corresponding to a different delay bound [5]. Dynamically partitioning the flows into aggregates would be too complex and generate too much signalling to be scalable. One must, therefore, resort to static partitioning, leading to sub-optimal results. This sub-optimal partitioning inevitably leads to even more underutilization of network resources in a model which already suffers highly from this problem. In any case, this partitioning leads to a larger number of aggregates and, therefore, to more signalling and underutilization.

One limitation of the model specification, stated in section 1.4.8 of [4], is the present lack of multicast support. Several factors contribute to this, namely (1) the difficulty in constructing a multicast tree that assures that aggregate Path messages follow the same path as data packets and (2) the amount of heterogeneity that may exist in an aggregate multicast reservation. Even if (1) is solved, solving (2) would probably lead to a set of procedures which provide no substantial reduction in the amount of state stored and messages processed by interior nodes. The proposed solution is a hybrid one, where the reservations are setup using end-to-end signalling, making use of aggregation only for packet classification and scheduling. Partly due to this limitation, our implementation was simplified in a way such that multicast is not presently supported.

The procedure for requesting a new aggregate reservation or modifying an existing one consists on sending an aggregate Resv message with the requested flowspec towards the aggregator. Notice that it is essential for the deaggregator to be signalled if the aggregate reservation modification is successful up to the aggregator. One method proposed in [4] to do this, which our implementation uses, is the confirmation of changes to reservations by means of ResvConfirm messages: if there is enough available bandwidth along the path to accommodate the requested aggregate bandwidth up to the aggregator, a ResvConfirm message will be sent to the deaggregator; if not, the deaggregator will receive an aggregate

ResvError message. Since the rejection of the (modification of the) aggregate may occur in any node from the deaggregator up to the aggregator, when the former receives the aggregate ResvError message, the bandwidth reserved for the aggregate may be the larger requested one up to some interior node, though not up to the aggregator. In this case, the deaggregator is responsible for the removal of the excess bandwidth, which will not be used, by sending a new aggregate Resv message with the last confirmed flowspec.

The flowspec value effectively used for admission control in the aggregate must be the GLB (Greatest Lower Bound) of the last requested and confirmed flowspecs, since it is the value guaranteed to be available all the way up to the aggregator. The way the aggregation model is specified, it may lead to inconsistencies if a single ResvConf message is lost. Figure 2 illustrates this problem. Av is the minimum flowspec installed in all the links from the deaggregator up to the aggregator (the available flowspec); R and C are, respectively, the last requested and confirmed flowspecs (at the deaggregator); Eff is the flowspec used for admission control to the aggregate (also at the deaggregator). Suppose a reservation with a bandwidth value of X was successfully installed and confirmed. Then, at some instant, the deaggregator decides to release some unused bandwidth, setting up a reservation with a bandwidth value of $Y<X$. If the confirmation for this reservation was lost in transit, the last confirmed bandwidth value remains X. Now suppose a new modification is attempted, increasing the bandwidth to a value of $Z>Y$ (and, in the illustrated case, also $Z>X$). At that time, the GLB of the two flowspecs becomes X, when effectively only Y bandwidth is reserved. Worse, if this modification fails, triggering a ResvErr message, the deaggregator will try to restore the flowspec X in order to avoid bandwidth wastage. However, since $X>Y$, this request may also fail, totally confusing the deaggregator. This problem may be solved by adding a rule that the effective bandwidth used for admission control can only be increased in response to the arrival of a ResvConf message.

3 Performance analysis

In this section we evaluate the performance of the RSVP Reservation Aggregation architecture based on results from several different sets of simulations. The obtained results are compared against those of simulations performed using the standard RSVP/IntServ architecture with the same topology. We analyze the standard QoS parameters (delay, jitter and packet loss ratio), the network resource utilization at the core link, and the reservation setup time. The number of signalling packets processed at the core is also analyzed in order to ascertain the scalability of the architecture and the improvement over standard RSVP.

Although admission control for aggregates must be parameter-based (PBAC), admission control for flows inside aggregates may be either parameter- or measurement-based (MBAC). We performed simulations with PBAC and MBAC. In the RSVP/IntServ simulations we used PBAC.

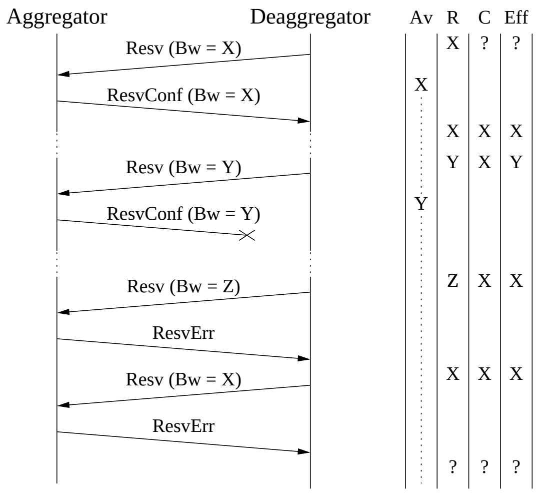


Fig. 2. Loss of ResvConf problem

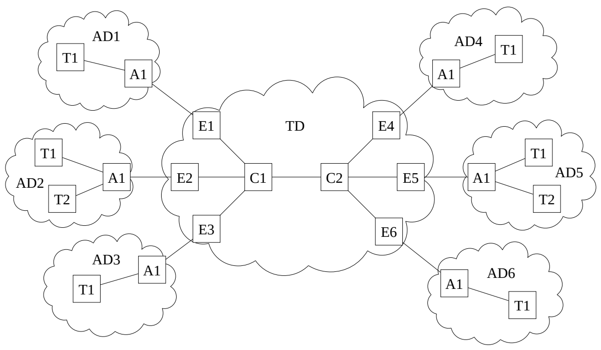


Fig. 3. Simulated topology

Figure 3 shows the topology used in these simulations. It consists of a transit (core) domain, TD, and 6 access domains, AD1–AD6. Each terminal in the access domains simulates a set of terminals. The bandwidth of the links in the transit domain and the interconnections between the transit and the access domains is 10 Mbps. The propagation delay is 2 ms in the transit domain and 1 ms in the interconnections between domains. There are up to 9 different aggregates in the link between C1 and C2, since there are 3 edge routers connected to C1 and other 3 connected to C2. The bandwidth assigned to the Controlled Load (CL) class is 7 Mbps. The bandwidth assigned to signalling traffic is 1 Mbps. Notice that although this seems very high, it is only an upper limit. The unused remaining 2 Mbps, as well as the unused bandwidth from the CL and signalling classes, is used for best-effort (BE) traffic.

The RSVPRAgg implementation has some tunable parameters. Except where otherwise noted, we used a bulk size of 500 kbps and a hysteresis time[1] of 15 s.

Type	Avg. rate (kbps)	Peak rate (kbps)	On time (ms)	Off time (ms)	Resv. rate (kbps)	Resv. burst (bytes)	MTBC (s)	Avg. dur. (s)	MOL (kbps)	ROL (kbps)
CBR	48	-	-	-	48	1500	13.8	120	1670	1670
Exp.	48	96	200	200	64	2500	6.8	120	3388	4518
Video	16	-	-	-	17	4000	17.1	180	674	716
Video	64	-	-	-	68	8000	17.1	180	2695	2863

Table 1. Flow characteristics

Each terminal of the access domains on the left side generates a set of flows belonging to the CL class, as well as filler traffic for the BE class. Each source may generate traffic to all destinations (terminals on the access domains of the right side), and the destination of each flow is randomly chosen. Filler BE traffic is composed of Pareto on-off and FTP flows. The traffic in the CL class is composed of a mixture of different types of flows, both synthetic — Constant Bit-Rate (CBR) and Exponential on-off (Exp.) — and real world multimedia streams — packet traces from H.263 videos, available from [6]. We used several different video traces for each bit-rate, starting each flow from a random point in the trace in order to avoid unwanted correlations between flows. The characteristics of the set of flows used are summarized in table 1. These flows are initiated according to a Poisson process with a certain mean time between calls (MTBC), and each flow has a duration which is distributed exponentially (synthetic flows) or according to a Pareto distribution (video traces), with the average value shown in the table (Avg. dur.). BE flows are active for all the duration of the simulations.

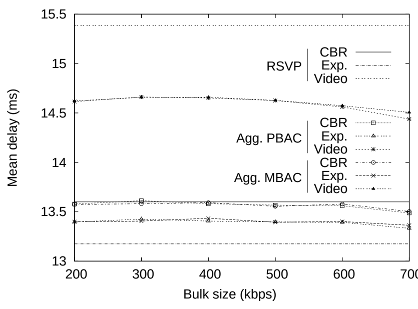
The largest mean offered load (MOL) in the CL class is, in terms of average traffic rates, about 20% higher than the bandwidth allocated to that class, which translates in an excess of about 40% in terms of reserved rates (ROL - Reserved Offered Load).

All simulations presented in this paper are run for 5400 simulation seconds, and data for the first 1800 seconds is discarded. All values presented are an average of, at least, 5 simulation runs with different random seeds. The next sub-sections present the results of these experiments.

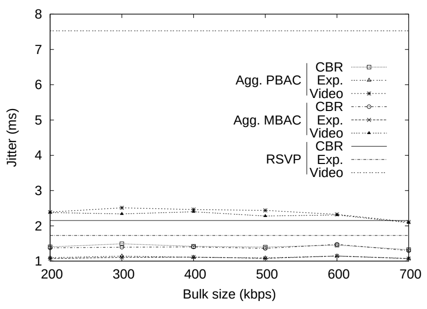
3.1 Variable bulk size

An important parameter in the RSVPRAgg architecture is the bulk size, which has implications on both the network resource usage and the signalling scalability. In the first experiment we vary the bulk size from 200 kbps to 700 kbps and use the maximum offered load (corresponding to the values presented in table 1). The results from this experiment are presented in figure 4. Reference values obtained with standard RSVP/IntServ are also provided.

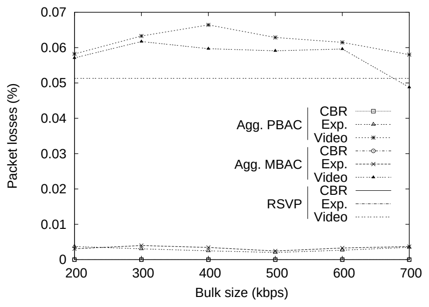
[1] Unused bulk removal delay.



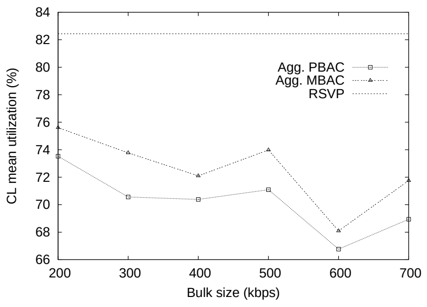
a) Mean delay



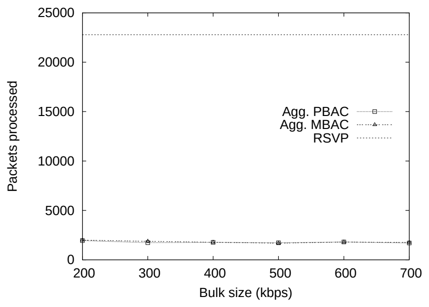
b) Jitter



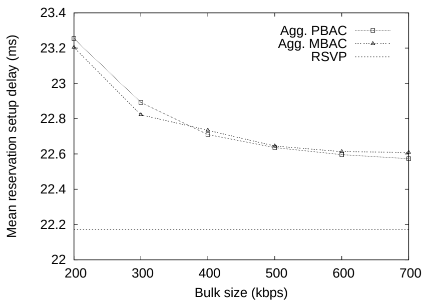
c) Packet losses



d) CL class utilization



e) Processed sig. packets at core node 1



f) Reservation setup delay

Fig. 4. Simulation results with variable bulk size

As we may see from figure 4.a the mean delay does not vary much with the bulk size. It is about the same as in RSVP for CBR flows, slightly higher for Exponential flows, and somewhat lower for the video streams. Jitter (fig. 4.b) is always lower in RSVPRAgg, particularly in the case of video streams. This indicates that indeed the upper tail of the delay distribution is reduced by aggregating flows. Packet losses (fig. 4.c) for video streams are slightly higher in RSVPRAgg than in standard RSVP. Both in RSVPRAgg and in RSVP there are no packet losses in CBR flows. Contrary to RSVP, there is a small amount of loss ($<0.005\%$) in exponential flows in RSVPRAgg. This amount of loss is, however, acceptable in a controlled load class. The admission control method for flows in aggregates does not seem to have a significant impact on the QoS parameters.

Regarding the utilization of the CL class (fig. 4.d), we may see that it is noticeably lower in RSVPRAgg than in standard RSVP. This is due to the fact that sometimes bandwidth is needed in an aggregate when it is not available, though there is spare bandwidth in other aggregates. As expected, utilization is even lower when using PBAC than when using MBAC since less flows are admitted in each aggregate. It is interesting to notice that there are local maxima in network resource utilization for bulk sizes of 500 kbps and 700 kbps, which are submultiples of the bandwidth available for the CL class (7 Mbps). This shows that it is good practice to choose a bulk size that is submultiple of the bandwidth allocated to the service class.

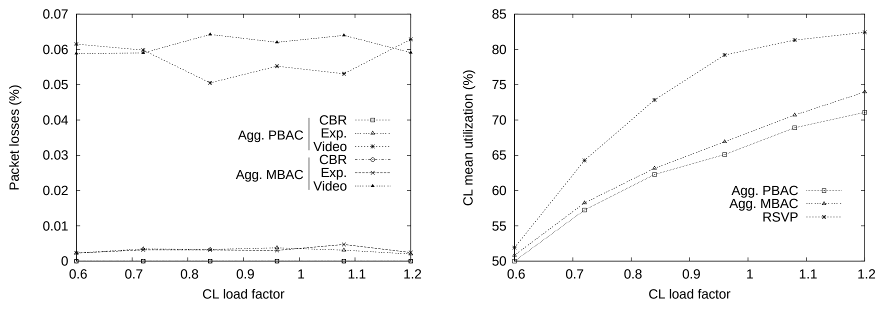
An important parameter in the evaluation of the signalling scalability is the number of signalling packets processed at core nodes. Figure 4.e shows the number of signalling packets processed at node C1. As may be seen, the number of messages processed at the core is reduced more than tenfold from RSVP to RSVPRAgg (from about 23000 to about 1800). This represents, indeed, a very significant increase in signalling scalability. Though not easily seen in the figure, there are local minima in the number of messages processed at the core with bulk sizes of 500 kbps and 700 kbps, which is another reason to choose a submultiple of the assigned bandwidth as the bulk size.

Figure 4.f shows the reservation setup delay. It is very important to notice that the curves relate only to the delays imposed by signalling message exchange and do not include processing time, since the ns-2 simulator is not suitable for the measurement of processing delays. The reservation setup delay decreases with increasing values of the bulk size. This behavior is expected since with larger bulk sizes more reservations are accepted without the need for increasing the aggregate bandwidth, which requires additional signalling. In the simplest case (appendix 2 in [4]), it basically consists on a round-trip time, the same as standard RSVP. In the more complex cases (appendices 1 and 3 in [4]), one or two round-trip times for the aggregation region are added. With the inclusion of processing times the setup delay would be much lower for RSVPRAgg than for the scalability-impaired RSVP.

The results presented above indicate that the RSVPRAgg architecture is able to meet the QoS requirements of a controlled load class, being able to replace the standard RSVP/IntServ architecture with substantial gains in scalability. The drawback is a lower usage of network resources.

3.2 Variable offered load

In the second experiment we evaluate the behavior of the RSVPRAgg architecture with varying offered load. The flows are the ones shown in table 1, but the mean time between calls (MTBC) is adjusted to vary the offered load from 60% (load factor of 0.6) to 120% (load factor of 1.2) of the bandwidth assigned to the CL class. The MTBC values presented in the table correspond to a load factor of 1.2. Figure 5 shows some results from this experiment.



a) Packet losses b) CL class utilization

Fig. 5. Simulation results with variable offered load

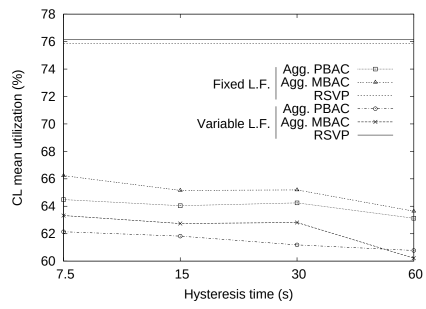
All QoS parameters are essentially constant, not depending on the offered load factor. Admission and traffic control are, therefore, being effective in emulating the behavior of a lightly loaded best-effort network, characteristic of the controlled load class. Regarding CL class utilization, for low values of offered load it is almost the same in RSVPRAgg and in standard RSVP, but it grows much faster in RSVP as the load factor approaches 1. At this point, the utilization curve for RSVP saturates, while those of RSVPRAgg continue to grow, exhibiting no visible saturation. The utilization with MBAC is slightly higher than with PBAC, since more flows are accepted. The largest difference in utilization between RSVP and RSVPRAgg is about 15% of the bandwidth allocated to the CL class (about 1 Mbps difference).

3.3 Variable hysteresis time

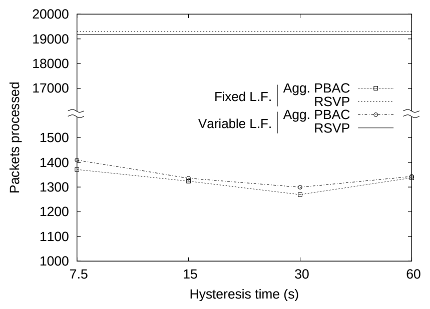
With this experiment we evaluate the influence of the hysteresis time in the utilization of the CL class and the number of signalling packets processed at the core. Hysteresis is needed in order to avoid oscillation in the reserved rate of an aggregate when operating in stable conditions, with the sum of reservations for the aggregate around a multiple of the bulk size. In these simulations, only one terminal in each access domain is transmitting. The offered load is 90% of the bandwidth allocated to the class in terms of traffic and 105% in terms of reserved rates. We performed two different sets of simulations, one using the same average amount of offered load in all transmitting terminals at all times (Fixed LF - Load Factor), and another one affecting the offered load in each terminal by a multiplicative factor of 0.5, 1 or 1.5[2] (Variable LF); these factors are rotated between the transmitting terminals every 400 simulation seconds. This rotation has the effect of forcing bandwidth to be released from some aggregates and requested in different ones.

Figure 6 shows the results with the variation of the hysteresis time (i.e., the delay for the removal of an unused bulk) from 7.5 s to 60 s. As expected, the

[2] The total offered load, therefore, remains the same.



a) CL class utilization



b) Processed sig. packets at core node 1

Fig. 6. Simulation results with variable hysteresis time

utilization decreases when the hysteresis time increases. This is due to the fact that unused bandwidth bulks are being held in aggregates for longer periods of time before being released and made available to other aggregates which may need it. The largest difference in utilization is obtained in the variable (rotating) load factor simulations when using PBAC; in this case, the difference is larger than 3% of the bandwidth allocated to the class. The number of signalling packets processed at the core also depends on the hysteresis time, although the variation is not very large, particularly if compared with the gains of using RSVPRAgg instead of the standard RSVP. It is interesting to notice that there is a minimum in the number of packets processed for a hysteresis time of 30 s. This behavior is due to the prevalence of one of two factors. For low values of hysteresis time, increasing this value means an increased probability that a flow will be admitted into the aggregate without need for increasing its bandwidth, since spare bandwidth is being held for longer periods. For higher values of hysteresis time another factor becomes dominant: the higher number of failed attempts to increase the bandwidth in some aggregates while spare bandwidth is being held in others. Although there is a minimum hold period between attempts to increase an aggregate's bandwidth, it was fixed at 5 s in these simulations. In face of these results, large values of hysteresis time are not recommended.

We performed a similar experiment keeping the hysteresis time constant at 15 s and varying the offered load rotation time between 200 s and 800 s. The results show that this variation does not noticeably affect the CL class utilization.

4 Conclusions

In this paper we performed an evaluation of the RSVP Reservation Aggregation architecture, proposed by the IETF as an alternative to the standard RSVP/IntServ architecture which is scalable enough for usage in high-speed core domains. We gave an overview of the architecture, pointing out its main strengths and weaknesses. We described our implementation of RSVPRAgg in the ns-2 simulator and discussed some particularities of the implementation and limitations of the architecture and its definition. Policies which are considered out of the scope of [4] were defined, namely the aggregate bandwidth management policy. The tunable parameters of our implementation were also presented.

The simulation results indicate that the RSVPRAgg architecture is able to meet the QoS requirements of the controlled load IntServ class. This is achieved with much lighter classification, forwarding and signalling procedures than those of RSVP/IntServ. A comparison of the number of signalling packets processed at the core in RSVPRAgg and standard RSVP/IntServ shows that the former is, indeed, much lighter and scalable, which is an absolute requirement for its deployment in high-speed core networks. The drawback, as demonstrated, is a lower utilization of network resources. Based on the analysis of the simulation results, we also provide some guidelines for setting the tunable parameters, namely the bulk size and the hysteresis time.

Due to the unsuitability of the ns-2 simulator to evaluate processing times, the scalability of the architecture could only be ascertained by the number of signalling packets processed at the core, as well as a qualitative analysis based on the nature of the packet classification and scheduling procedures. Only a prototype implementation would allow for a quantitative analysis, which is a topic for further work. The study of a better solution for multicast than the one proposed in [4], which consists on the use of end-to-end RSVP signalling with aggregate packet classification and scheduling, is another topic for further work.

References

1. Braden, R., Clarck, D., Shenker, S.: Integrated Services in the Internet Architecture: an Overview. RFC 1633, Internet Engineering Task Force (1994)
2. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource Reservation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, Internet Engineering Task Force (1997)
3. Blake, S., Blake, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. RFC 2475, Internet Engineering Task Force (1998)
4. Baker, F., Iturralde, C., Faucheur, F.L., Davie, B.: Aggregation of RSVP for IPv4 and IPv6 Reservations. RFC 3175, Internet Engineering Task Force (2001)
5. Schmitt, J., Karsten, M., Wolf, L., Steinmetz, R.: Aggregation of Guaranteed Service Flows. In: Seventh International Workshop on Quality of Service. (1999) 147–155
6. Arizona State University: MPEG-4 and H.263 Video Traces for Network Performance Evaluation. http://trace.eas.asu.edu/TRACE/trace.html (2004)