Comparative Evaluation of Two Scalable QoS Architectures

Rui Prior, Susana Sargento, Pedro Brandão, and Sérgio Crisóstomo

DCC & LIACC, Faculty of Sciences, University of Porto
Rua do Campo Alegre, 823, 4150-180 Porto, Portugal
`{rprior, ssargento, pbrandao, slc}@ncc.up.pt`

Abstract This paper performs a comparative evaluation of two QoS architectures, RSVP Reservation Aggregation and Scalable Reservation-Based QoS, aimed at providing QoS levels similar to the ones provided by the well-known RSVP/IntServ architecture, but scalable enough for use in high traffic core networks. The comparative analysis, based on each model's characteristics and on simulation results, shows that our Scalable Reservation-Based QoS architecture is able to support the same QoS guarantees provided by RSVP Reservation Aggregation, with significantly increased resource utilisation and a small penalty in signalling processing overhead.
Keywords: QoS, scalability, performance evaluation

1 Introduction

With the goal of benefiting from the virtues of both IntServ [1] and DiffServ [2] architectures and mitigating their problems, several architectures have been proposed in the literature. None of these architectures, however, ensures simultaneously the strict and differentiated QoS support and the maximisation of the usage of network resources without scalability concerns.

One of the most promising architectures [3] is based on aggregation of per-flow reservations, where the RSVP protocol [4] is extended to allow RSVP signalling messages to be hidden inside an aggregate. In the simplest case, all edge routers reserve bandwidth between ingress and egress routers of a network domain; these reservations can be updated in bulks much larger than the individual flow's bandwidth. When 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 there is, the flow is accepted without signalling the core routers. Otherwise, the core routers will be signalled in an attempt to increase the aggregate's bandwidth in bulk quantities. If this attempt succeeds, the flow is admitted; otherwise, it is rejected. This architecture benefits from the fact that signalling messages are only exchanged when the aggregate's bandwidth needs to be updated. Unfortunately, the decrease in signalling rate is accompanied by a decrease in resource utilisation.

In order to address the requirements of end-to-end QoS support without resource utilisation and scalability concerns, we developed a new architecture [5]

based on scalable per-flow signalling and resource reservations at both transit (core) and access networks with aggregate packet classification and scheduling. Several techniques and algorithms have been developed aiming at the minimisation of the computational complexity and, therefore, the improvement of the signalling scalability. More specifically, a label switching mechanism was developed with the goal of avoiding expensive lookups in flow reservation tables. Moreover, a scalable implementation of expiration timers for soft reservations, with a complexity that is low and independent from the number of flows, was also developed. In terms of QoS guarantees, [6] showed that our architecture is able to support strict and soft QoS guarantees to each flow, irrespectively of the behaviour of the other flows in the same and in different classes, with resource utilisation similar to the one obtained with IntServ, but increased scalability.

In this paper we qualitatively and quantitatively compare both architectures in terms of QoS guarantees, resource utilisation, and scalability, in order to evaluate their relative merits and shortcomings, as well as their suitability to replace the reference RSVP/IntServ architecture, which suffers from scalability problems that disallow its usage in high traffic core networks. The results indicate that both the Scalable Reservation-Based QoS (SRBQ) and the RSVP Reservation Aggregation (RSVPRAgg) models are able to provide adequate QoS levels and may, therefore, be used in place of RSVP/IntServ.

This paper is organised as follows. Section 2 consists on a comparative analysis of these two models, and is divided in two subsections, where the models are qualitatively compared based on their main characteristics (2.1) and quantitatively compared based on simulation results (2.2). Section 3 presents the most important conclusions from this work and points out some topics for future work.

2 Comparison between SRBQ and RSVPRAgg

In this section we address both qualitative and quantitative comparisons of SRBQ and RSVPRAgg. The qualitative comparison is based on the nature of the architectures, drawing their advantages and disadvantages. The quantitative comparison is based on simulation results.

A mapping between SRBQ and RSVPRAgg needs to be performed for the comparison to be fair. The aggregation regions of RSVPRAgg and the non-aggregated RSVP regions correspond to the core and access domains of SRBQ; the aggregators and deaggregators in RSVPRAgg to edge routers in SRBQ. The network topology used for the simulations is the same given this mapping.

2.1 Qualitative comparison

The core routers in the RSVPRAgg architecture only need to store the state of aggregates, whereas in SRBQ per-flow state is stored. This is not a limiting factor of SRBQ, however, considering the routers available nowadays: in a core router handling an average of a hundred thousand flows, if each flow's information occupies 100 bytes, only 10 Mbytes are required. The most complex

task for the core routers is the lookup of the flow information, based on the 5-tuple parameters that specify the flow, when the number of flows is very high. Since SRBQ uses the labels to overcome this problem, the existence of per-flow reservation structures is not a limitation of the architecture.

The scalability of the classification and scheduling procedures at the core nodes is similar in both architectures, since they are performed on a per-aggregate basis according to the DSCP of the flows. At the edge routers classification is much lighter in the SRBQ model, since no flow to aggregate mapping is performed. As scheduling is based on the DSCP on both architectures, the efficiency is comparable. At the access routers, packet classification is per-flow based in both models. Therefore, the processing load of classification is similar, but may be much reduced in SRBQ if labels are used in data packets. Packet scheduling is more efficient in SRBQ, since in RSVPRAgg usually Weighted Fair Queueing (WFQ) or a similar discipline is used outside the aggregation regions.

The approach to reducing the signalling processing load is radically different. In SRBQ end-to-end reservations are used, and scalability is achieved by making use of highly efficient techniques and algorithms (like labels and efficient timers). In RSVPRAgg end-to-end reservations are aggregated, reducing the signalling processing at the core to that needed to maintain and update aggregate reservations in bulk quantities. This approach has two disadvantages: (1) the number of signalling messages processed at the edge nodes of the aggregation region, which may be a high-traffic transit domain, is even higher than in the case of regular RSVP, since both end-to-end and aggregate messages must be processed; worse, packet classification at the edge is per-flow; and (2) the reduction in signalling is highly dependent on the bulk size, but large bulk sizes lead to a very poor utilisation of network resources. Due to the different approaches, although the number of signalling messages processed at core nodes is much higher in SRBQ, the processing load is comparable.

2.2 Quantitative comparison

Both the SRBQ and the RSVPRAgg models were implemented in the ns-2 simulator. An existing implementation of RSVP for this simulator was also used. It is important to keep in mind that ns-2 has some limitations, the most significant of which is the inability to simulate and measure processing delays.

The simulated scenario is depicted in figure 1. It includes 1 transit (TD) and 6 access (AD) domains. Each terminal in the access domains simulates a set of terminals. The bandwidth of the connections in the transit domain, and in the interconnections between the transit and the access domains, is 10 Mbps. The propagation delay is 2 ms in the transit domain connections and 1 ms in the interconnections between the access and the transit domain. The transit domain corresponds to the aggregation region in the RSVPRAgg architecture.

The simulated scenario contains a class for signalling traffic, CL and BE classes. At each referred connection, the bandwidth assigned to the signalling traffic is 1 Mbps. Note that, although this seems very high, the unused signalling bandwidth is used for BE traffic. The bandwidth assigned to the CL

class is 7 Mbps. The remaining bandwidth, as well as unused CL and signalling bandwidth, is used for BE traffic.
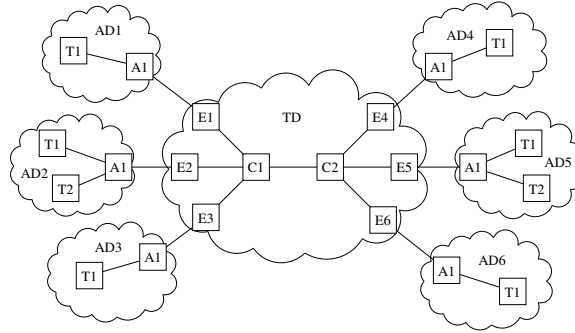

Figure 1. Simulation topology

Each terminal of the access domains on the left side generates a set of flows belonging to the CL and BE classes. Each source may generate traffic to all destinations; the destination of each flow is randomly chosen in the set of the terminals in the right side access domains. With 3 source and 3 destination edge routers in the core domain, the number of required end-to-end aggregates in the domain is 9. Traffic belonging to the CL class is a mixture of different types of flows: CBR, exponential on-off and Pareto on-off. These flows are initiated according to a Poisson process with a certain mean time interval between calls (MTBC), and flows' durations are exponentially distributed. Filler traffic in the BE class is composed by on-off Pareto and FTP flows. All simulations were run for 5,400 simulation seconds, discarding the data for the first 1,800 seconds. All values presented are an average of 5 simulation runs with different random seeds.

Table 1 shows the flow types used in the simulations, along with their characteristics. Reservations for RSVPRAgg and RSVP are of token-bucket type; in SRBQ they are based on 3 water-marks. The mean time between calls (MTBC) is adjusted so that all flow types have, on average, the same amount of reserved bandwidth; The amount of offered load is varied from 0.8 to 1.2 (load factor) times the CL bandwidth at the core by adjusting the MTBC in all flows. The MTBC value presented in the table corresponds to a load factor of 1.

Type	Avg. rate (kbps)	Pkt. size (Bytes)	On (ms)	Off (ms)	Pk. rate (kbps)	Token Bucket		Watermarks (kbps)			MTBC (s)	Avg. dur. (s)
						R (kbps)	B (Bytes)	1	2	3		
cbr48cl	48	500				48	1500	48	48.048	56	13.2	120
cbr64cl	64	500				64	1500	64	64.064	72	17.5	120
exp1cl	48	500	200	200	96	64	15000	32	64	96	17.5	120
pareto1cl	48	500	200	200	96	64	15000	32	64	96	17.5	120

Table 1. Flow characteristics

Figure 2 contains the most relevant results. The mean delay is shown for all flow types in SRBQ and RSVPRAgg with a bulk size of 300 kbps. As can be seen, all flows in SRBQ suffer similar average delays, which is an obvious result

since they share all the queues. In RSVPRAgg the average delay inflicted to different flow types is different due to the use of WFQ outside the aggregation region; Pareto flows in this model have significantly higher queueing delays than the other flows.

Regarding packet losses, contrary to all other types, Pareto flows have a very significant packet loss ratio of about 10% in the RSVPRAgg model. It is worth noting that the Pareto distribution is a heavy-tailed one, with infinite variance. This implies that Pareto on-off flows are not well suited for token-bucket characterisation, since unless we use disproportionately large bucket sizes, packet losses will always be high. The three rate water-marks characterisation used in SRBQ for the CL class is much more appropriate for this kind of flow: losses for Pareto flows in SRBQ are always less than 0.003%. Packet losses for exponential on-off flows are very low in RSVPRAgg (about 0.003%) and null in SRBQ. All CBR flows have no packet losses in both models.
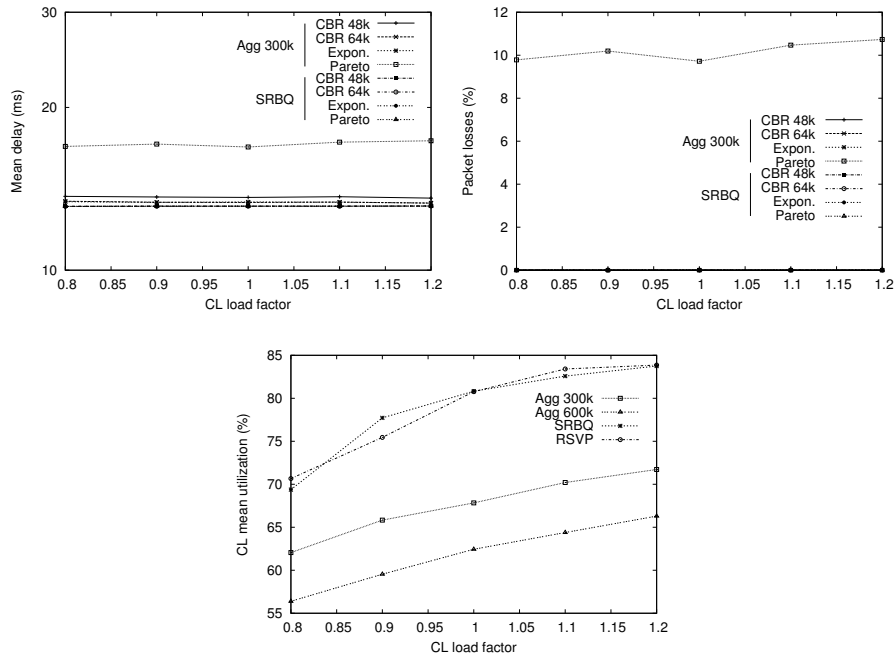

Figure 2. Delay, packet loss and CL class utilization results

SRBQ and RSVP have similar utilisation figures, showing link saturation around an offered load factor of 1; RSVPRAgg has lower utilisation values which decrease with increasing bulk size, and does not exhibit saturation.

From the previous results we may conclude that both models provide adequate QoS, except for Pareto flows in RSVPRAgg which are not well suited for the token bucket type reservations used in that model.

3   Conclusions and Future Work

In this paper we performed a comparative evaluation of two QoS architectures, RSVPRAgg and SRBQ, aimed at providing QoS levels similar to those provided by the RSVP/IntServ architecture, but scalable enough for use in high traffic core networks. The comparison was based on implementations of both architectures in the ns-2 simulator.

From the results presented in the previous section, we may state that both the RSVPRAgg and SRBQ models provide adequate QoS levels and flow isolation in the CL class. In terms of packet classification and scheduling, both models perform aggregation of end-to-end flows and make use of the DSCP field. The number of signalling messages processed at core nodes is much lower in RSVPRAgg due to the fact that signalling scalability in SRBQ is not obtained by performing reservations at an aggregate level; instead, it makes use of highly efficient techniques and algorithms while keeping the end-to-end character of signalling. The signalling gain in RSVPRAgg is dependent on the bulk size used in aggregate bandwidth management, but the use of large bulks leads to gross under-utilisation of network resources. Based on end-to-end reservations, the SRBQ model does not have this limitation, making good use of network resources under all conditions. In addition to this, edge nodes of aggregation regions in RSVPRAgg must perform per-flow signalling (and in some cases also packet classification and scheduling). In high traffic transit networks, the number of flows even at the edge of transit domains may be huge, imposing a scalability limit to RSVPRAgg which must not be overlooked.

As future work, we plan to evaluate the possibilities for interoperability between SRBQ and other QoS architectures. We also plan to perform simulations with more realistic flows, based on data collected in real networks, and to implement prototypes of both architectures in order to evaluate performance parameters not provided by ns-2, namely regarding processing power required.

References

1. Braden, R., Clarck, D., Shenker, S.: Integrated Services in the Internet Architecture: an Overview. RFC 1633, Internet Engineering Task Force (1994)
2. Blake, S., Blake, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. RFC 2475, Internet Engineering Task Force (1998)
3. Baker, F., Iturralde, C., Faucheur, F.L., Davie, B.: Aggregation of RSVP for IPv4 and IPv6 Reservations. RFC 3175, Internet Engineering Task Force (2001)
4. 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)
5. Prior, R., Sargento, S., Crisóstomo, S., Brandão, P.: End-to-end Quality of Service with Scalable Reservations. In: Proceedings of the 11th International Conference on Telecommunication System, Modeling and Analysis. (2003)
6. Prior, R., Sargento, S., Brandão, P., Crisóstomo, S.: Efficient Reservation-Based QoS Architecture. In: Interactive Multimedia on Next Generation Networks. Volume 2899 of Lecture Notes in Computer Science., Springer-Verlag (2003) 161–181