Scalable Reservation-Based QoS Architecture (SRBQ)

Rui Prior
LIACC – University of Porto, Portugal

Susana Sargento
University of Aveiro, Portugal

INTRODUCTION

Having its roots in the military ARPANET, conceived as a data transport network with a focus on resilience, the Internet supports only a best-effort service model, where all packets are treated the same way, therefore providing a single level of service. Now that the Internet is becoming a ubiquitous global communication infrastructure, new applications are emerging with more demanding and diversified requirements than data transport. Internet telephony, for example, has much stricter delay requirements than remote terminal, the most demanding of the original applications. The deployment of other service models providing better quality of service (QoS) is of great importance for the transport of these new applications.

While many approaches to providing QoS on the Internet have been proposed, none has gained enough acceptance to be deployed across the Internet at large. One important reason for this is the difficulty in conciliating strict QoS guarantees with the required scalability and with good resource utilization.

The realization that the scalability limitations of the existing per-flow reservation QoS architectures are not intrinsic to the per-flow approach itself, but rather to the employed algorithms, led to the development of the scalable reservation-based QoS (SRBQ) architecture (Prior et al., 2003a).

BACKGROUND

Two main architectures have been standardized by the IETF for the introduction of QoS and traffic differentiation on the Internet. The Integrated Services (IntServ) architecture (Braden, Clark, & Shenker, 1994), commonly used with the Resource Reservation Protocol (RSVP) (Braden et al., 1997), provides strict QoS guarantees and efficient resource usage, but suffers from scalability problems concerning the per-flow scheduling, classification, and reservation procedures. The differentiated services (DiffServ) architecture (Blake et al., 1998), conceived as a solution to the limitations of IntServ, is free from these scalability concerns, since flows are aggregated in classes according to specific characteristics, but without admission control mechanisms, all the flows from a given class may receive degraded quality of service due to excessive traffic at the link.

Aiming at the introduction of QoS support without the aforementioned problems, several other architectures have been proposed in the literature. All of these architectures, however, suffer from one or more of the following problems: lack of strict QoS guarantees, underutilization of network resources, or scalability limitations stemming from the complexity of the algorithms and procedures used. For example, the SCORE architecture (Stoica, 2000) keeps the stateless character of the network by carrying state information in data packet headers, but imposes a scheduling discipline on all routers with a computational complexity that is still high. In the Egress Admission Control architecture (Cetinkaya & Knightly, 2000), only the egress routers perform admission control, based on passive monitoring, but it cannot provide strict QoS guarantees. With probing schemes (Almesberger, Ferrari, & Le Boudec, 1998; Bianchi, Capone, & Petrioli, 2000; Elek, Karlsson, & Ronngren, 2000; Breslau, et al., 2000; Sargento, Valadas, & Knightly, 2001; Key & Massoulié, 2003) no network control is required, but no firm QoS guarantees are possible, among other problems. In Bernet et al. (2000) a framework is proposed

for the operation of IntServ over DiffServ networks, avoiding signaling processing inside the domain, but it may have sub-optimal resource allocation and imprecise admission control. The RSVP Reservation Aggregation architecture (Baker et al., 2001) achieves scalability by aggregating reservations in core domains, but it implies a tradeoff between resource utilization and signaling scalability, leading to underutilization in high-speed networks. Westberg et al. (2002) proposed the resource management in DiffServ (RMD), based on similar principles.

Centralized approaches have also been proposed (Nichols, Jacobson, & Zhang, 1999; Schelen, & Pink, 1998; Terzis et al., 1999; Chimento et al., 2002; Sargento et al., 2004) which offload the (core) routers from the need to maintain state and perform signaling by moving resource management and admission control to bandwidth brokers (BB), with per-flow or aggregate reservations. While the routers are simplified, BBs are subject to scalability limitations, particularly severe in per-flow approaches, and become single points of failure. Aggregate BBs, on the other hand, share some of the problems of distributed aggregation-based architectures, namely, lower resource utilization.

The Next Steps in Signaling (NSIS) Working Group has proposed a two-layer extensible signaling architecture that addresses many limitations of RSVP, having QoS signaling as one of the first applications (Fu et al., 2005). NSIS concerns signaling only, and was designed to support any QoS model; therefore, its characteristics in terms of QoS, resource utilization, and scalability are largely dependent on the underlying QoS model.

In the scalable reservation-based QoS (SRBQ) architecture, scalability is achieved by lowering the computational complexity of all the tasks associated with a per-flow, reservation-based architecture for QoS provisioning, namely those related to packet scheduling, reservation signaling, and admission control. This approach is able to conciliate scalability with a good utilization of network resources (Prior et al., 2003b, 2004a, 2004b, 2004c).

THE SRBQ ARCHITECTURE

SRBQ combines the strict end-to-end QoS guarantees of a signaling-based approach with per-flow reservations subject to admission control, both in terms of bounded delay and minimal loss, with the efficiency and scalability provided by flow aggregation and by several mechanisms and algorithms. The next sub-sections describe some aspects of the architecture. A detailed description is provided in Prior et al. (2003a).

General System Architecture

The underlying architecture of the proposed model is strongly based on DiffServ (with which it may coexist) with the addition of signaling-based reservations subject to admission control. The network is partitioned into domains, consisting of core and edge nodes. In addition, access domains also have access nodes. Individual flows are aggregated according to service classes, mapped to DiffServ (DS) compatible per-hop behaviors (PHBs), and aggregate classification is performed based on the DS field of the packet header.

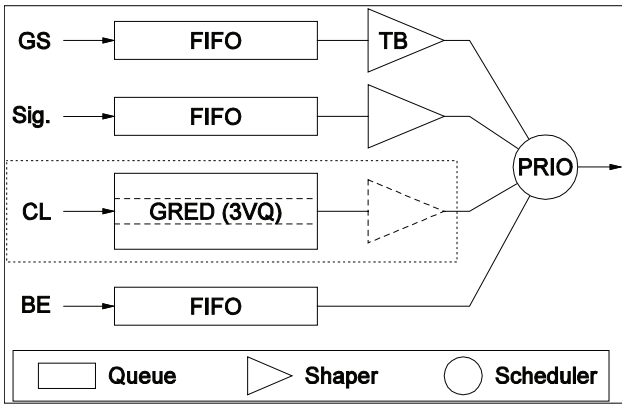
Besides best effort (BE), SRBQ provides two additional service classes: (1) the guaranteed service (GS) class that is characterized by hard QoS assurance in terms of both delivery guarantee and maximum delay, based on the same principles as the EF (expedited forwarding) PHB in DiffServ; and (2) one or more controlled load (CL) classes that emulate the behavior of lightly-loaded best effort networks, based on the AF (assured forwarding) PHB. The simplest queuing model for the routers is depicted in Figure 1(a). There are up to four different controlled load service classes using DiffServ code points (DSCP) from other AF classes, provided these are not used by DiffServ. In this case, the CL queuing block is replaced by the one shown in Figure 1(b). Reservations for traffic flows using the GS class are characterized by a token bucket. Reservations for traffic flows using CL classes are characterized by three average rate watermarks: packets exceeding the first two watermarks will receive a degraded service in terms of drop probability; packets exceeding the third watermark will be dropped.

Admission control is performed at every node along the flow path. A GS flow *i*, characterized by the token-bucket (r_i, b_i), is admitted in a link with *j* GS flows already accepted if:

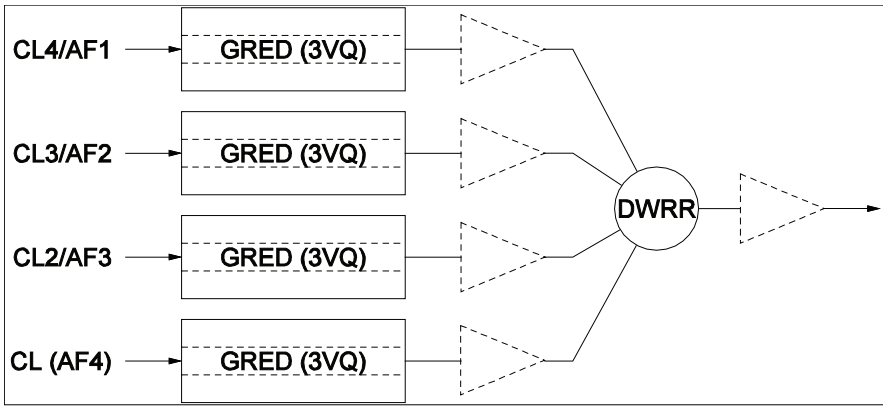
$$r_i + \sum_j r_j \leq R_{GS\max}$$

and

Figure 1. Queuing model



a) Single CL class



b) Multiple CL classes

$$b_i + \sum_j b_j \le B_{GS\max},$$

where $R_{GS\max}$ and $B_{GS\max}$ are the bandwidth and buffer space assigned to the GS class. A CL flow i is admitted if

$$r_{i,w} + \sum_j r_{j,w} \le R_{w\max}, \forall w,$$

where $R_{w\max}$ is the configured maximum rate for watermark w. Admission control in the GS class must be parameter-based (PBAC), whereas in the CL class it may be parameter- or measurement-based (MBAC).

As illustrated in Figure 1, the highest priority queue, corresponding to the GS traffic class, is subject to a token-bucket type traffic shaper in order to avoid packet dropping at the next policing node. Since the shaper curve is the summed token bucket, an upper bound to the arrival curve of the aggregate, this shaper does not degrade the QoS guarantees of the GS class (Le Boudec & Thiran, 2001). The signaling/routing traffic, though not subject to admission control, must be shaped in order to prevent starvation of the CL class. This class may also be shaped, but this is only required if the network administrator wants to ensure that the best effort class does not starve. Contrary to the GS shaper, these are work-conserving.

All nodes in the architecture perform signaling and support the previously described queuing model. The access nodes perform per-flow policing for the CL class and per-flow ingress shaping for the GS class. Edge nodes perform aggregate policing and DSCP remarking. Core nodes perform no policing.

Label Switching

In other protocols, one of the scalability-limiting tasks for the core routers, especially in terms of worst case, is the lookup of the stored flow information, based on the 5-tuple parameters that specify the flow, usually implemented using hash tables. In order to efficiently access the reservation structures, SRBQ employs a label-switching mechanism which allows direct access to these structures without any need for hash lookups. These labels are 32-bit values, whose meaning is externally opaque, but internally may be an index to a table of reservation structures or the memory address of the reservation structure. Three label fields are stored in this structure: B, T, and F. The T (this) label, which may be implicit, is the label for the node itself, directly mapped to the memory address of the reservation structure; the B label, to be used in messages sent backwards (upstream), corresponds to the T label of the previous hop; the F label, to be used in messages sent forwards (downstream), corresponds to the T label of the next hop. Labels are installed at reservation setup time.

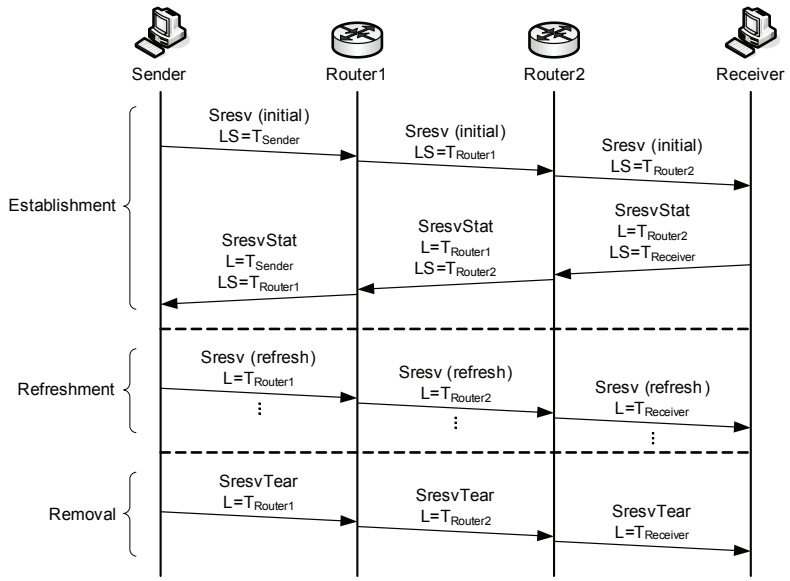
The label switching mechanism is also advantageous in all per-flow processing, like policing performed at the access routers. The labels may also be used to improve route change detection: A mismatch between the next hop assigned by the routing tables with the one stored in the reservation structure of the flow means that the route has changed. In order to profit from these advantages on per-flow processing, all packets would need to carry the label information. Notice that in spite of these advantages, labels are not used for packet classification (except perhaps at the access routers), since it is performed on an aggregate basis using just the DS field of the IP header.

Signaling Protocol

The signaling protocol works on a hop-by-hop basis, providing unidirectional, soft state, sender-initiated reservations. Though implemented as an extension to the RSVP protocol, SRBQ is much more scalable, since (1) the access to the flows' information is direct by using the labels, (2) timers for the expiration of soft reservations are implemented in a very efficient way, and (3) it uses simple reservation identification in order to decrease the length of the refresh and explicit tear down messages. As RSVP is meant to perform receiver-initiated reservations, SRBQ extends it by adding three new message types (see Figure 2): SResv (sender reservation), used to establish, refresh, and modify reservations; SResvStat (sender reservation status), used for reservation confirmation and error reporting; and SResvTear (sender reservation tear down), used to explicitly terminate a reservation.

Full SResv messages include flow identification, reservation quantification, a LABEL_SETUP object (used to install the label), an identifier of the service class, and a reservation expiration timeout value. The last two are conveyed by a SRESV_PARMS object. Upon receiving an initial SResv message, the request is subject to admission control; if accepted, the router updates the resource reservation of the flow's class, creates an entry for the flow in the reservation structure list, stores the label at the B field for this reservation, and forwards the SResv message to the next router after changing the LABEL_SETUP to the reservation entry assigned to this flow. If the flow cannot be accepted (anywhere in the path), a SResvStat message is sent towards the sender reporting the error. This message already makes use of the B labels in order to access to the flow reservation structure. When the SResv reaches the destination, all routers along the path have reserved resources for the new flow and all labels required for backward message processing are installed in the reservation state. The receiver acknowledges the successful reservation by sending a SResvStat message towards the sender, making use of the labels already installed in the opposite direction. The LABEL object in this message is used to access the memory structure for this reservation and the LABEL_SETUP object is stored in the F field of each node. Each node switches the LABEL to the value installed at the B field and forwards the message to the next node upstream, until the sender is reached. The SResvStat message will also trigger the commitment of the resource reservation to both the policing and the queuing modules at the routers if the reservation succeeded, or the removal from the admission control module if it failed.

Figure 2. Message flow



Soft State Timers

In SRBQ, reservations are soft state: If no SResvTear message is received and the reservation is not refreshed, the associated timer expires and it is removed. Having a good range of reservation expiration timer values means that short-lived flows will not remain stale for long periods whenever something unusual occurs (such as an application lockup or premature termination, or an undetected route change) but longer-lived flows will not generate too much signaling traffic just to refresh the reservation.

The basic implementation concept for timers is a sorted event queue: The processor waits until the first timer value in the list expires, dequeues it, performs the appropriate processing, then goes on waiting for the next timer value to expire. While dequeuing an event is trivial, inserting an event with a random expiration time is a very expensive operation, highly dependent on the total number of events queued. Contrasting to the complexity of generic timers, fixed delay timers are very simple and efficient to implement (a single FIFO queue). As a compromise between the two types, SRBQ uses an algorithm with trivial timer queuing and low and constant cost timer dequeuing, providing eight possible timer delays in a base-2 logarithmic scale with a range of 1:128. The implementation is based on eight different queues, each of which has an associated fixed delay. Internally, therefore, these queues are served using a FIFO discipline. Enqueuing an event is a simple matter of adding it to the tail of the corresponding queue, which is trivial. Dequeuing an event means choosing one of the eight possible queues (the one whose timers expires first) and taking the first event from that queue.

Applications should use timer values representing a good tradeoff between signaling traffic and fast recovery from faults for the expected flow lifespan. When the lifespan cannot be estimated a priori, the application may use a short timer at first and increase it using the SRESV_PARMS objects in refresh messages.

PERFORMANCE OF SRBQ

The SRBQ architecture has been thoroughly evaluated (Prior et al., 2003b, 2004b), using both synthetic flows and real-word multimedia streams. It has been demonstrated that SRBQ is able to provide strict and soft QoS guarantees in the GS and the CL classes, respectively, and that adequate isolation is achieved between the traffic classes and between flows in the same class (particularly in GS).

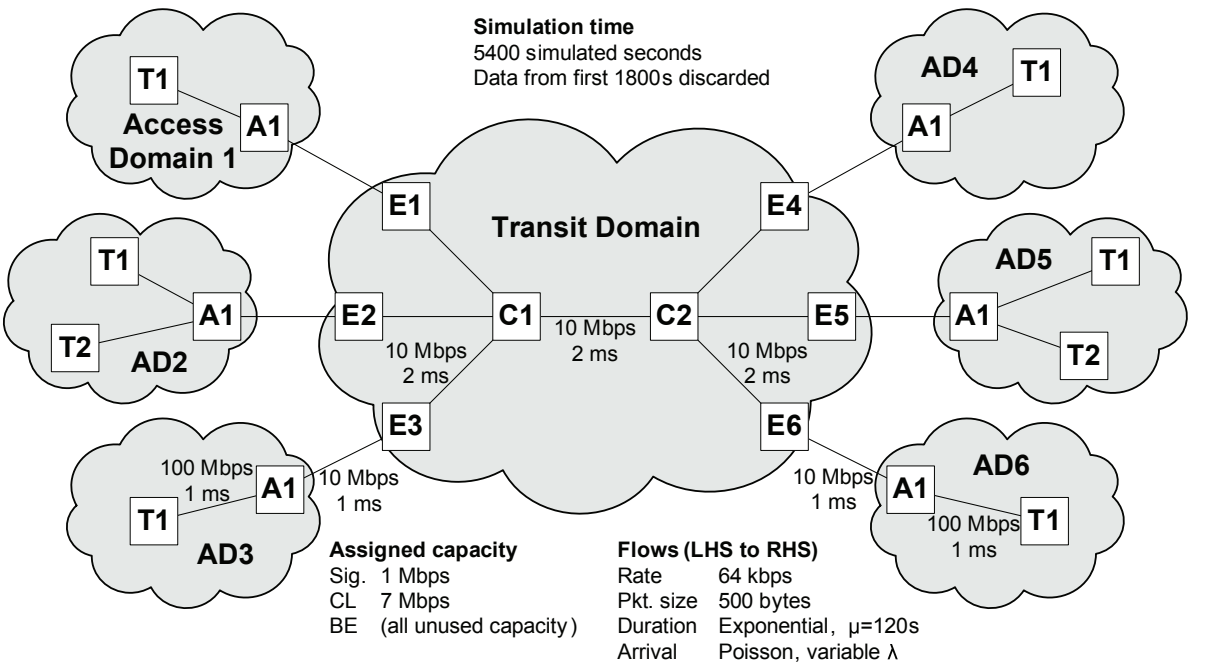
A comparative analysis of SRBQ and RSVP aggregation (RSVPRAgg) was presented in Prior et al. (2004a, 2004c). Both models aim at providing QoS

levels comparable to RSVP/IntServ, but in a scalable manner. Both of them make use of flow aggregation in order to achieve scalability in packet classification and scheduling; the main differences stem from the different approaches to signaling. With RSVPRAgg, reservations at the core are performed in an aggregate basis and their bandwidth is updated in bulk quantities. However, per-flow signaling is performed at edge routers of transit domains (and also classification and scheduling, at the deaggregators). In SRBQ, the end-to-end character of reservation signaling is preserved without scalability issues; the amount of stored state is not a problem (Prior, 2003b), and resource usage is always optimal.

The next paragraphs describe some results from Prior (2004b), obtained by simulation in ns-2 (Version 2.26) using the dumbbell topology of Figure 3. All the information about the simulation setup is summarized in the figure. The mean time between calls is adjusted to vary the offered load between 0.8 and 1.2 times the bandwidth allocated to the CL class at the core link. The results from this set of simulations are presented in Figure 4.

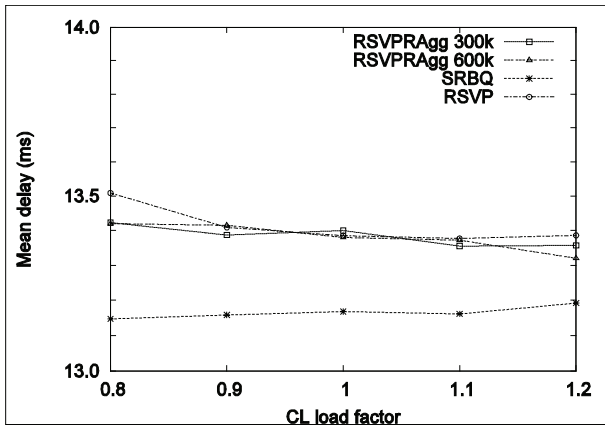
In all models, the mean delay is not much larger than the sum of transmission and propagation delays (12.08 ms), meaning that the time spent in queues is low. Nevertheless, it is lower in SRBQ, as is jitter (not shown), probably due to the use of WFQ in RSVP and in RSVPRAgg outside the aggregation region. In all models presented there are no losses. Regarding the utilization of bandwidth allocated to the CL class, it is much higher in SRBQ (similar to standard RSVP) than in RSVPRAgg. In the latter, the utilization decreases noticeably with the increase in the bulk size: The use of larger bulk sizes in order to increase the scalability would lead to very poor network resource utilization. Corresponding to the lower utilization figures, the blocked bandwidth in RSVPRAgg is higher than in SRBQ. The blocked bandwidth figures are similar in SRBQ and regular RSVP, since end-to-end reservations are accepted up to the bandwidth reserved for the CL class in both models, contrasting to the RSVPRAgg model in which end-to-end reservations are only accepted up to the reserved rate of the corresponding aggregate. The number of signaling messages processed at C1 (see Figure 3) was also evaluated. This number

Figure 3. Topology used in comparison simulations

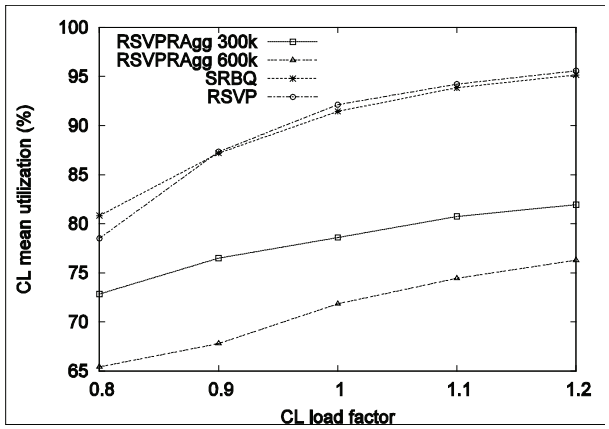


9 possible aggregates in RSVPRAgg from left hand side (LHS) to right hand side (RHS)

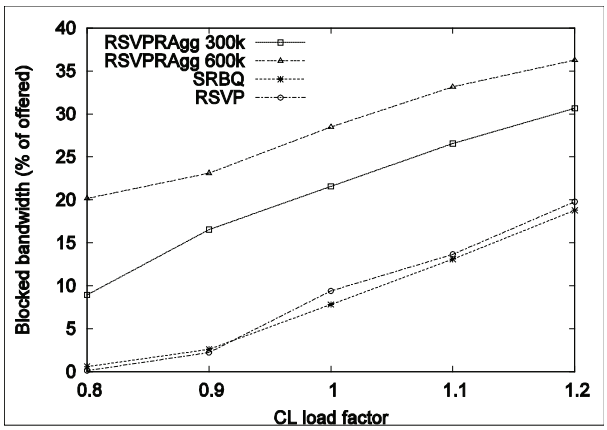
Figure 4. Performance of SRBQ, RSVPRAgg, and RSVP with a single type of flow



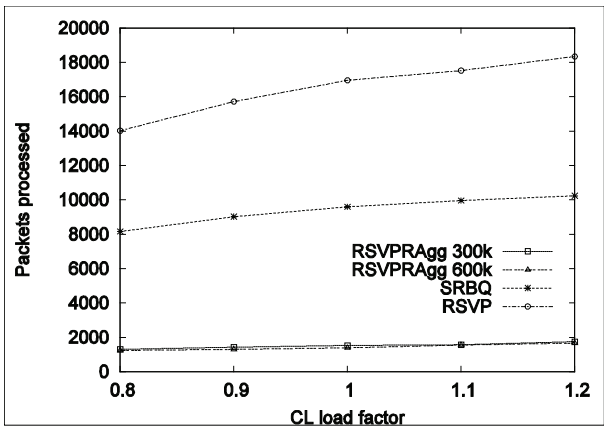
a) Delay



b) CL class utilization



c) Blocking probability



d) Signaling packets processed at the core

is much lower in RSVPRAgg than in SRBQ or RSVP. This is an obvious result, since at interior nodes only aggregate messages are processed in RSVPRAgg. The almost twofold difference between SRBQ and RSVP is due to the fact that in RSVP both Path and Resv refreshes are needed. Though from the number of processed messages at the core nodes alone RSVPRAgg looks more scalable, the SRBQ model makes use of low complexity, highly efficient algorithms which use much less CPU time to process each message.

FUTURE TRENDS

The future of QoS on the Internet is still unclear. In order for operators to deploy QoS mechanisms, they must regard them as an opportunity for increased revenue, by charging a premium for connections with QoS. Given the ever-increasing number of services provided over the Internet, many with real-time and other QoS requirements, the demand is there, but it is currently cheaper for operators to increase the core capacity than to deploy QoS mechanisms. However, the ratio of core-to-access capacity, currently at a peak,

has historically oscillated as a result of the evolution of core and access technologies, and the need for QoS mechanisms becomes evident at the troughs (Crowcroft et al., 2003). On the other hand, there is a case for end-to-end QoS mechanisms inside administrative domains, where it is much easier to deploy a single mechanism in every node: For example, many broadband ISPs provide content-oriented and other value-added services only to their customers. SRBQ may certainly find application in this field, even if it will not be implemented in the whole Internet.

CONCLUSION

The SRBQ architecture emerged as a scalable alternative to RSVP/IntServ. It is able to provide both IntServ service models—guaranteed service with strict QoS guarantees, and controlled load with soft QoS guarantees—with an underlying DiffServ-like architecture. The use of aggregate packet classification and scheduling mechanisms combined with the use of efficient algorithms (label switching, etc.) minimizes the processing load at each network element, allowing SRBQ to scale to a very large number of simultaneous flows. Compared with the RSVP reservation aggregation architecture proposed by the IETF, SRBQ provides comparable (and usually more favorable) QoS values, while allowing for a substantially higher utilization of the network resources, therefore reducing the amount of blocked reservations.

REFERENCES

Almesberger, W., Ferrari, T., & Le Boudec, J.-Y. (1998). SRP: A scalable resource reservation protocol for the Internet. *Computer Communications*, *21*(14), Special issue on "Multimedia Networking," 1200-1211.

Baker, F., Iturralde, C., Le Faucheur, F., & Davie, B. (2001). *Aggregation of RSVP for IPv4 and IPv6 reservations*. IETF RFC-3175.

Bernet, Y., et al. (2000). *A framework for integrated services operation over Diffserv networks*. IETF RFC-2998.

Bianchi, G., Capone, A., & Petrioli, C. (2000). Throughput analysis of end-to-end measurement-based admission control in IP. *Proceedings of IEEE INFOCOM 2000*.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., & Weiss, W. (1998). *An architecture for differentiated services*. IETF RFC-2475.

Braden, R., Clark, D., & Shenker, S. (1994). *Integrated services in the Internet architecture: An overview*. IETF RFC-1633.

Braden, R., Zhang, L., Berson, S., Herzog, S., & Jamin, S. (1997). *Resource reservation protocol (RSVP)–Version 1 functional specification*. IETF RFC-2205.

Breslau, L., Knightly, E. W., Shenker, S., Stoica, I., & Zhang, H. (2000). Endpoint admission control: Architectural issues and performance. *Proceedings of ACM SIGCOMM 2000*.

Cetinkaya, C., & Knightly, E. (2000). Egress admission control. *Proceedings of IEEE INFOCOM 2000*.

Chimento, P., et al. (2002). *QBone signaling design team.* Final report.

Crowcroft, J., Hand, S., Mortier, R., Roscoe, T., & Warfield, A. (2003). QoS's downfall: At the bottom, or not at all! *Proceedings of the ACM SIGCOMM 2003 Workshops*.

Elek, V., Karlsson, G., & Ronngren, R. (2000). Admission control based on end-to-end measurements. *Proceedings of IEEE INFOCOM 2000*.

Fu, X., Schulzrinne, H., Bader, A., Hogrefe, D., Kappler, C., Karagiannis, G., Tschofenig, H., et al. (2005). NSIS: A new extensible IP signaling protocol suite. *IEEE Communications Magazine*, *43*(10), 133-141.

Key, P., & Massoulié, L. (2003). Probing strategies for distributed admission control in large and small scale systems. *Proceedings of the IEEE INFOCOM 2003*.

Le Boudec, J.-Y., & Thiran, P. (2001). Network calculus: A theory of deterministic queuing systems for the Internet. *Lecture Notes in Computer Science, 2050*. Springer-Verlag.

Nichols, K., Jacobson, V., & Zhang, L. (1999). *A two-bit differentiated services architecture for the Internet.* IETF RFC-2638.

Prior, R., Sargento, S., Brandão, P., & Crisóstomo, S. (2004a). Comparative evaluation of two scalable QoS

architectures. *Networking-2004*: *Lecture Notes in Computer Science, 3042*, 1452-1457. Springer-Verlag.

Prior, R., Sargento, S., Brandão, P., & Crisóstomo, S. (2003b). Efficient reservation-based QoS architecture: Interactive multimedia on next generation networks. *Lecture Notes in Computer Science*, *2899*, 168-181. Springer-Verlag.

Prior, R., Sargento, S., Crisóstomo, S., & Brandão, P. (2003a). End-to-end QoS with scalable reservations. *Proceedings of the 11th International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM11)*.

Prior, R., Sargento, S., Brandão, P., & Crisóstomo, S. (2004b). Evaluation of a scalable reservation-based QoS architecture. *Proceedings of the Ninth IEEE International Symposium on Computers and Communications (ISCC-2004)*.

Prior, R., Sargento, S., Brandão, P., & Crisóstomo, S. (2004c). SRBQ and RSVPRAgg: A comparative study. *Telecommunications and Networking. ICT 2004, 11th International Conference on Telecommunications*. *Lecture Notes in Computer Science, 3124* (pp. 1210-1217). Springer-Verlag.

Sargento, S., et al. (2004). *QoS architecture and protocol design specification.* IST-DAIDALOS project deliverable D321.

Sargento, S., Valadas, R., & Knightly, E. (2001). Resource stealing in endpoint controlled multi-class networks. *Proceedings of IWDC 2001* (invited paper).

Schelen, O., & Pink, S. (1998). Aggregating resource reservations over multiple routing domains. *Proceedings of the 6th International Workshop on Quality of Service (IWQoS'98)*.

Stoica, I. (2000). *Stateless core: A scalable approach for quality of service in the Internet.* PhD thesis, Carnegie Mellon University.

Terzis, A., Wang, L., Ogawa, J., & Zhang, L. (1999). A two-tier resource management model for the Internet. *Proceedings of GLOBECOM'99*.

Westberg, R., Császár, A., Karagiannis, G., Marquetant, A., Partain, D., Pop, O., et al. (2002). Resource management in Diffserv (RMD): A functionality and performance behavior overview. *Protocols for High Speed Networks – 7th IFIP/IEEE International Workshop*, *PfHSN*. *Lecture Notes in Computer Science*, *2334* (pp. 17-34). Springer-Verlag.

TERMS AND DEFINITIONS

Arrival Curve: The arrival curve of a flow is a wide-sense increasing function α defined for $t \geq 0$, such that the amount of data flowing in any time interval of length t is less than or equal to $\alpha(t)$. It is used to place a constraint on the flow's arrival process.

Flow Aggregation: Merging of multiple flows, possibly sharing common characteristics, in order to treat them as a single flow in the use of a given resource.

Label Switching: Technique used in protocols whereby a short identifier (label) is carried in signaling messages and/or data packets in order to allow for easy and efficient classification or access to a given data structure or state information. Along the path, labels may be kept intact, exchanged, or stacked, according to the protocol.

Quality of Service (QoS): Subjectively defined in Recommendation E.800 of the ITU-T as "The collective effect of service performance which determines the degree of satisfaction of a user of the service," QoS refers to the probability of the network meeting a given traffic contract, which may be quantitatively expressed by parameters such as transfer delay and jitter and probability of packet loss, error, or out-of-order delivery.

Scalability: The ease with which a system or component can handle increased dimensions of the problem it is designed to solve.

Soft State: Technique whereby the state information is automatically deleted if not refreshed for a given period. It is used to improve the resilience of protocols by providing automatic recovery from faults and changing conditions.

Traffic Policing: Forcing an input flow to have an output that conforms to a given traffic envelope σ by discarding non-conformant bits (or packets) of the flow or reclassifying them to a different flow or aggregate.

Traffic Shaping: Forcing an input flow to have an output that conforms to a given traffic envelope σ by

delaying the non-conformant bits (or packets) of the flow in a buffer. At the output of the traffic shaper, the flow has σ as an arrival curve.