Integrated QoS Architecture for IP Switching

Simon Pietro Romano(1)*, Christophe Deleuze(2),
José Rezende(2) and Serge Fdida(2)

(1)Gruppo di Ricerca Informatica Distribuita
Universitá di Napoli Federico II - Naples - Italy
sprom@grid.grid.unina.it

(2)LIP6 - C.N.R.S.
Université Pierre et Marie Curie - Paris - France
{deleuze,rezende,fdida}@rp.lip6.fr

Abstract

In this paper we investigate issues related to the integration of QoS capabilities in the framework of IP switching technology. First, we propose an architecture which includes the RSVP protocol with minimal impact on the original structure and protocols of IP switching. This leads us to further investigate the queueing and scheduling mechanisms adopted both at the routing and the switching level in order to guarantee effective support of Guaranteed (GS) and Best-Effort (BE) services. Based on the observation that GS service typically results in network underutilization, we then introduce a new service class (BR) which provides flows with long-term rate guarantees without impacting GS flows and while exploiting their unused bandwidth. Finally, we propose a possible implementation of this new service and we validate it with the help of simulation.

1 Introduction

New applications such as telemedicine, computer support for collaborative work (CSCW) and distributed simulation systems are very demanding in terms of the QoS received from the communication service. In the Internet community the IETF Integrated Services (IntServ) working group has specified a control QoS framework in order to provide these applications with the appropriate support.

IP has been playing for several years the most important role in global internetworking. Its surprisingly rapid growth makes it the dominating protocol in the scene of networking research. However, IP choice does not only rely on the fact that the vast majority of applications has been conceived for it. The connectionless nature of IP has proved to be a key feature to support integrated services networks. On the other hand, the ubiquitous deployment of ATM-based networks rises many important issues concerning the interoperation of IP and ATM architectures. The different paradigms on which they are based, i.e. connectionless and connection-oriented, make their coexistence a difficult task. Another major concern is QoS integration in an IP/ATM environment.

Different solutions to face the problem of IP over ATM are presented in the literature. One that we deemed worth further investigating is IP Switching introduced by Ipsilon [1]. This solution relies on fully eliminating the burden imposed by ATM signaling while preserving the connectionless nature of IP. Here, a portion of IP flows is switched by the ATM fabric (*IP switch*) at the data link level instead of being forwarded at level 3 by IP routers. Switching IP flows completely in hardware eliminates the current performance bottlenecks imposed by the hop-by-hop forwarding engine.

Nevertheless, the quality of service received by each IP flow that traverses an IP switch depends on a flow classification decision which is done according to local policies. Such a decision in the current implementation is typically based on the type of application that generates the flow [1]. Therefore, the QoS obtained by each IP flow depends on both the Internet traffic distribution and the capabilities of the underlying ATM hardware,

*This work was done while this author was visiting LIP6.

i.e. the scheduling capabilities offered by switch fabrics. To support individual quality of service for IP flows a signaling mechanism must be supplied to allow an application to generate reservation requests. Furthermore, the scheduling mechanisms of the ATM switch must be configured to reserve these resources. RSVP is the signaling protocol that has recently received most of the attention [2]. It is a receiver initiated protocol and it uses a soft state to reserve resources in the network elements. The major issue here is the selection of those IP flows that are to be switched based on the resource reservation requested, i.e. *QoS flow classification* operation. Also, the QoS control of the entire traffic (forwarded and switched) that traverses an IP switch has to be performed in an integrated manner.

In this paper we investigate how IP Switching can provide individual quality of service for multiple IP flows and which support that solution requires from the underlying ATM hardware. Furthermore, we analyze how RSVP can be integrated into this architecture and how to provide a framework to control the quality of service of multiple traffic classes. The paper is organized in five sections. An overview of the Ipsilon solution is presented in section 2. Further, this section also discusses how RSVP can be integrated into IP Switching. Section 3 introduces an Integrated QoS Architecture for IP Switching. Here, we include a proposal for a new service class which allows a better utilization of network resources and a discussion of queueing, scheduling and admission control mechanisms. In section 4 we discuss the simulation results obtained. Conclusions are provided in section 5.

2 IP Switching and RSVP

IP Switching as proposed by Ipsilon consists in the specification of two protocols: GSMP (General Switch Management Protocol) and IFMP (Ipsilon Flow Management Protocol). The IP switch device is built of two components, an ATM switch fabric and the IP Switch Controller. GSMP [3] allows the IP Switch Controller to have full control of an ATM switch. Using GSMP the controller can manipulate the ATM switching table and at the same time the ATM switch can inform the controller about its capabilities. The IFMP [4] runs between the IP Switch Controller and its peers across each external link. It makes possible the distribution of cached information to be used in flow forwarding. This protocol allows a node to attach a label, notably the VPI/VCI pair, to a specified IP flow. After exchanging labels associated with a determined IP flow with the downstream and upstream neighbours, a node can switch packets belonging to this flow at level 2 rather than forwarding them at level 3 (figure 1). The operation that allows flow labelling and labels exchanging between IFMP peers is called *flow redirection*.

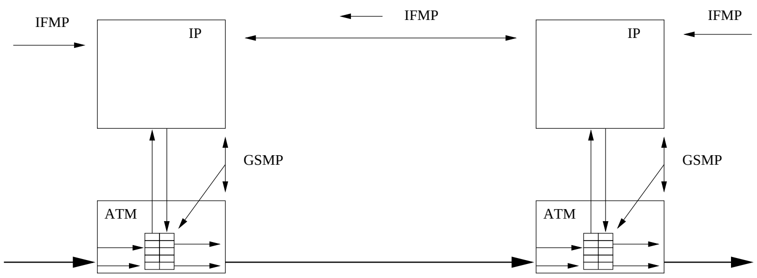


Figure 1: IP Switching

The decision to attach a label to an IP flow is taken according to local policies by the flow classification module. Usually this classification is based on the flow characteristics such as application, type of service field, protocol, etc. A flow identifier is associated to each label. A flow identifier contains the values of the set of fields from the packet header that a packet must match to belong to this flow. To maintain a *soft state* labels are aged. After a lifetime expires without refreshing the corresponding label-flow association is deleted.

To allow applications to request active end-to-end QoS control along the path of a packet flow a signaling protocol must be integrated into the architecture. This signaling protocol serves as a front end between the application requesting QoS control and the communication service. The RSVP protocol contains the main features required by applications to indicate their needs in terms of QoS. It is receiver initiated and likewise IFMP it maintains soft state information about the network resources dedicated to a specified session.

Data treated by RSVP are of three natures according to the entity that supplies (sender and receiver) or modifies (intermediate network elements) them. The information supplied by each sender, and conveyed in the

Sender TSpec object, concerns the type of traffic that it is going to generate. The *FlowSpec* object provided by each receiver is built of two parts, *Receiver TSpec* and *RSpec*. The former contains the traffic description the resource reservation should apply to while the latter carries the service class to be used and the corresponding quality of service parameters. The information modified by intermediate network elements is carried in the ADSPEC object which contains available services, delay and bandwidth estimates, and additional service specific parameters. This information is used by the receivers to choose a service and determine the reservation parameters. The RSVP specification does not define the internal format of RSVP protocol fields or objects described above. Rather, they are specified by each QoS control service according to its needs.

We argue that IFMP and RSVP protocols can be somehow integrated to provide QoS control to individual IP flows. The main motivation for this is that in this way the decision to switch a flow is fully determined by the QoS guarantees required by the applications rather than by local policies adopted by each IP switch. We claim that these two protocols have duplicated functionalities. By piggybacking IFMP redirection messages onto RESV messages we can have a more consistent control of the end-to-end QoS. In fact, the separation between resource reservation phase and flow redirection phase could create incongruous states. Also, the soft state nature of these protocols can be a major source of problems when trying to associate a lifetime to reserved resources. Further, we can reduce the overhead imposed by the two protocols when run separately.

To achieve IFMP-RSVP integration we propose to add a new reservation style, called IFMP, to RSVP. This extension would allow the encapsulation of IFMP redirection protocol messages as new objects within the flow descriptor list of RESV messages. What must be clear, however, is that we do not add to RSVP any further changes other than the introduction of IFMP reservation style. Upon receiving RESV messages with this new style the RSVP module passes the information contained in the corresponding RESV object to the IFMP module. After processing this information the IFMP protocol returns to the RSVP module the new IFMP redirection message to be sent to the upstream neighbour. Again, this message is encapsulated in a RESV message. Upon arrival of a RESV message at the sender a virtual circuit is already instantiated inside the network and can be used by the IP flow requiring QoS guarantees. Thanks to this encapsulation RESV messages automatically trigger the binding between flow and label.

3 Integrated QoS Architecture for IP Switching

Our integrated QoS architecture for IP Switching (figure 2) is organized in two parts. The routing part enhances the classical architecture of a QoS-capable IP router to support the capabilities made available by the IP switch as defined by Ipsilon. We find a packet classifier, a packet scheduler, RSVP, IFMP, GSMP and flow classifier modules. The switching portion is made of the ATM switch and is supposed to provide the advanced capabilities of a WFQ scheduler. The resource reservation is made by configuring both the queueing and scheduling hardware into the ATM switch, while flow policing functions are carried out via UPC hardware configuration in accordance to the *flowspec* parameter present in the RESV message.

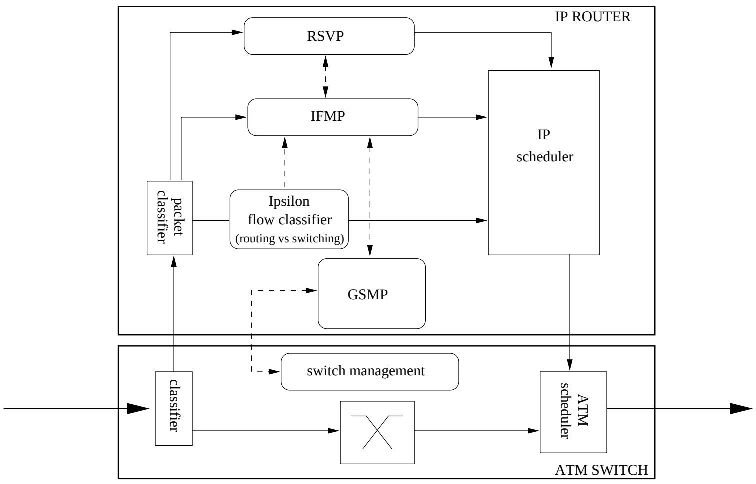


Figure 2: Integrated QoS Architecture for IP Switching.

3.1 Routing level scheduling

Scheduling policy at the routing level takes into account two different kinds of flows: (i) signaling traffic (i.e. RSVP/IFMP traffic); and (ii) elastic best-effort traffic. The signaling traffic is considered vital and is thus given priority over the best-effort traffic. The best-effort traffic is treated as a single flow. Thus, the scheduler comes out to be a prioritized FIFO with two queues; one for signaling traffic and the other for best-effort traffic.

3.2 Switching level scheduling

At the switching level all QoS classes are considered since we assume that, as long as it is possible, all flows requiring QoS are switched rather than routed. We recognize that VC space can be a major source of problems in this case, but we argue that from one side pricing policies will limit the demand in terms of QoS guarantees, and from the other side the continuous evolution of hardware technology will make available more and more powerful switches. Moreover, flow aggregation techniques could overcome this limitation.

3.2.1 Guaranteed Service

The GS class provides the clients data flow with firm bounds on the end-to-end delay experienced by a packet while traversing network queues [5]. It guarantees both bandwidth and delay. The GS emulates the service that would be offered by a dedicated communication channel between the emitter and the receiver. Two parameters apply to this service: *TSpec* and *RSpec*. The former describes the traffic characteristics for which service is being requested. It is composed of a token bucket plus a peak rate (p), a minimum policed unit (m), and a maximum datagram size (M). The *RSpec* specifies the QoS a given flow demands from a network element and it takes the form of a clearing rate R and a slack term S. The clearing rate is computed to give a guaranteed end-to-end delay and the slack term denotes the difference between desired and guaranteed end-to-end delay after the receiver has chosen a value for R.

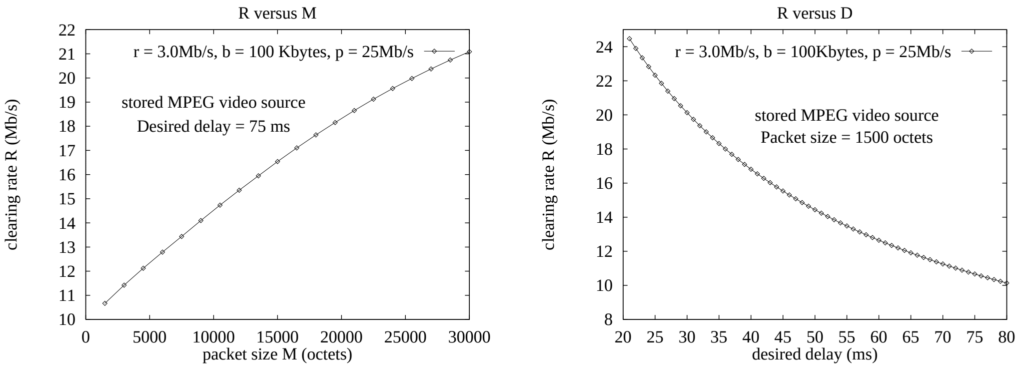
In our QoS architecture, Guaranteed Service flows are always treated directly at the data-link level. In this way, we are able to bypass the whole IP level scheduling mechanism and then to obtain a smaller value of the clearing rate R for the same end-to-end delay bound requirement. Such a policy leads to a better utilization of network resources.

During the reservation phase, each intermediate network element i computes C_i and D_i which represent the deviation from a fluid server which operates at the clearing rate R demanded by a potential receiver for a given flow. These values are accumulated into the C_{tot} and D_{tot} fields within the ADSPEC object of a PATH message. The following equation [6] applies to the end-to-end delay bound computation for a flow characterized by (b, r, p) and a network element with MTU size equal to L:

$$\overline{D}_i = \begin{cases} \dfrac{(b-L)}{R}\dfrac{(p-R)}{(p-r)} + \dfrac{L}{R} + \displaystyle\sum_{j=1}^{i}\left[\dfrac{C_j}{R} + D_j\right] & \text{if } p > R, \\ \dfrac{L}{R} + \displaystyle\sum_{j=1}^{i}\left[\dfrac{C_j}{R} + D_j\right] & \text{if } p \leq R. \end{cases} \tag{1}$$

The actual procedure used to determine the value of the clearing rate to be inserted in the RESV message is the following: (i) substitute $R = p$ in the second of the above formulae and compute the corresponding value of the upper bound on the end-to-end delay ($\overline{D}$); (ii) if $\overline{D} < D_{desired}$, then insert the known values of the flow characterization parameters and desired end-to-end delay bound ($D_{desired}$) into the first formula to determine the corresponding value of the clearing rate R; (iii) otherwise, use the second formula.

As an example, consider an OC-3 (155 Mb/s) output link at an ATM network element. We assume a stored MPEG video flow with an average rate of about 3.0 Mb/s and a maximum burst size of 100 Kbytes. The peak rate is only limited by the speed of the medium to which the source is attached, i.e. a 25 Mb/s UTP. We consider that each flow traverses 5 hops with a total propagation delay of 20 ms. Subtracting the propagation delay from the end-to-end delay requirement we obtain the allowable end-to-end queueing delay. All parameters described above need to be converted to take into account the overhead added by the LLC/SNAP encapsulation and ATM cell header. Substituting the end-to-end queueing delay in (1), we can solve for the rate that needs to be reserved for each flow. Figures 3(a) and 3(b) show the reserved rate as a function of the maximum packet size (M) and the end-to-end delay requirement (D), respectively. In figure 3(a), the associated end-to-end delay requirement, that needs to be translated into a reservation rate R, is 75 ms.



(a) R versus M

(b) R versus D

Figure 3: Stored MPEG video source

Guaranteed Service results in a fairly low utilization of the link bandwidth since the reserved rate R is always greater than the average rate of the flow. A solution must be found to allow other flows to exploit the unused bandwidth. Such flows must present less stringent requirements than GS flows. In the service class described below flows are able to recover this unused bandwidth while having some long-term bandwidth guarantee. The idea behind this service is similar to that presented in [6]. The main difference resides in the way it is implemented and how it interacts with other service classes such as best-effort.

3.2.2 BR Service

Many existing applications do not need firm guarantees in terms of delay bounds but do require a certain level of guarantee for the bandwidth. Actually, most of the applications can be satisfied by a minimum service rate plus a fair share of the residual bandwidth. For the best-effort service the minimum service rate is equal to zero. The introduction of a service class fulfilling these requirements could prove very useful in an integrated services architecture. The traffic generated by web applications is a good example of flows that could benefit from this kind of service.

To implement this service class, called BR (Bandwidth Recovery) Service, we use a WFQ scheduling mechanism with separate queues for each GS and BR flow. For each GS flow a weight is assigned that determines the reserved clearing rate for this flow. Assuming M GS flows characterized by the envelope (b, r, p) and a link capacity of C_l, to guarantee their end-to-end delay bounds the following inequality must be satisfied:

$$\sum_{i=1}^{M} R_i \leq C_l - \Delta_{\text{BR}} - \Delta_{\text{SIG}} \tag{2}$$

Δ_{BR} and Δ_{SIG} in the above equation represent the portions of bandwidth reserved to BR and signaling flows, respectively. The weight associated to the signaling flow is based on an estimation of the bandwidth needed by the signaling traffic. The weight associated to each BR flow, and thus the reserved bandwidth, is purposely chosen very small and it does not represent the minimum rate required by the flow. The function of the weight is to allow the flow to gain access to the residual bandwidth left by the GS flows without giving any guarantees in terms of delay bounds. Considering M' BR flows with average rate equal to r' to guarantee their minimum rate on a long-term basis the following inequality applies:

$$\sum_{i=1}^{M} r_i + \sum_{i=1}^{M'} r'_i \leq C_l - \Delta_{\text{SIG}} \tag{3}$$

The equation above ensures that given that GS flows respect their traffic contract or a suitable policing mechanism (e.g. GCRA algorithm at the UPC) is used, BR flows will receive a guaranteed minimum rate over a reasonable period of time. In our architecture, these flows are always switched at the data-link level.

3.2.3 Best-effort Service

BE flows are not in the scope of our study since they do not ask any reservation and thus do not make use of RSVP. Nevertheless they are integrated in our QoS architecture as low priority level flows for the data-link level WFQ scheduling mechanism. All BE flows are merged in a single queue of the scheduler and are only served when no traffic is present at the higher priority level queues. This prevents BE flows from impacting flows requiring QoS. Best-effort traffic is routed or switched according to the policies proposed by Ipsilon [1] et al [7].

Figure 4 depicts the scheduling mechanism adopted at the switching level of our architecture. In this figure we represent all flows, i.e. GS, BR and BE, and their corresponding queues.

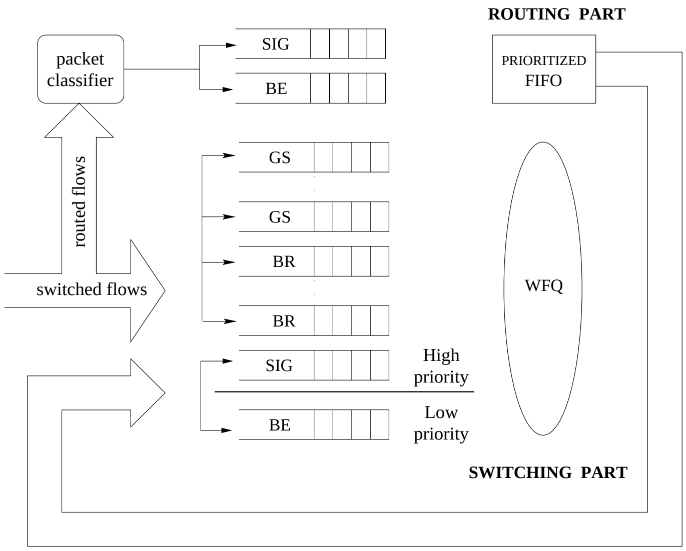


Figure 4: Scheduling mechanism.

4 Simulation results

In our simulations, we use a network (figure 5) with 5 IP switches. Each switch has a WFQ scheduler with per VC-queueing for each GS and BR flow, and one queue for all BE flows. To each IP switch is attached a host which generates and receives flows. A total of 9 GS flows are distributed in such a way that each inter-switch link is shared by 6 GS flows. There are 3 flows of path length five, 2 flows of length four, 2 flows of length three and 2 flows of length two.

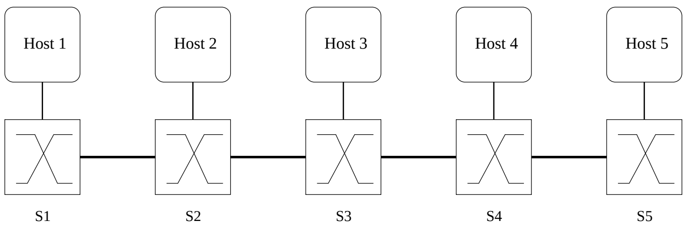


Figure 5: Network simulator.

All traffic sources are modeled by means of a two-state markov chain. In order to simulate real-time traffic, in each burst period (ON) a geometrically distributed random number of packets is generated at some peak rate p; $\overline{b}$ is the average size of this burst. The idle period (OFF) has average length I. The average packet generation rate is given by $r = \rho \times p$, where ρ gives the probability that the markov process is in the ON state. Given the relations: $\alpha = (\text{MeanON})^{-1} = \frac{p}{\overline{b}}$, $\beta = (\text{MeanOFF})^{-1} = \frac{1}{I}$ and $\rho = \frac{\beta}{\alpha+\beta}$, we obtain $r = \frac{1/I}{p/\overline{b}+1/I} \times p = \frac{\overline{b}}{pI+\overline{b}} \times p$.

To obtain flows characterized by a (b,r,p) envelope we subject each source to a series of two leaky buckets (figure 6). Token generation is a deterministic process with arrival time of a new token every $1/r$ units of time in the first bucket and $1/p$ in the second bucket. The token generation rate of the first stage actually determines the mean rate at which packets enter the network, while that of the second stage limits the peak rate to a maximum value of p. Due to losses at the first leaky bucket r_{OUT} is smaller than r.

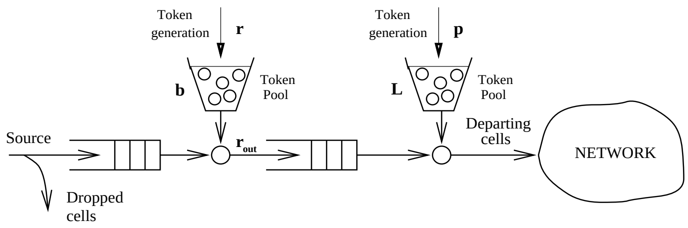


Figure 6: The double leaky bucket.

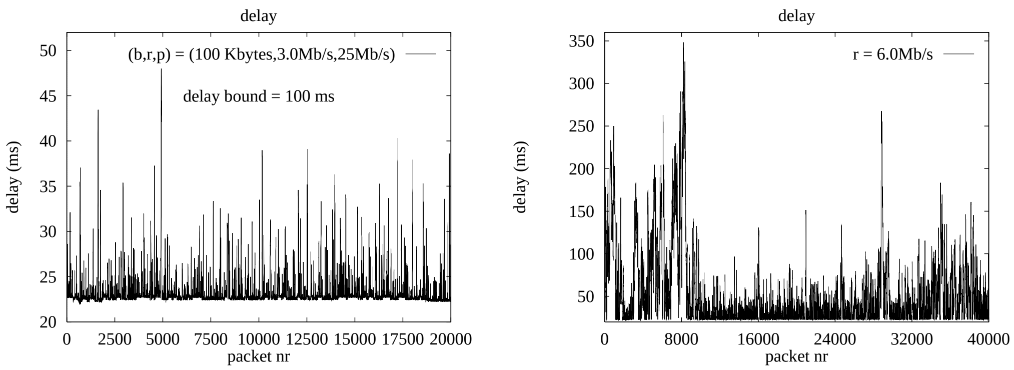
Each GS flow has the following characterization: $b = 100Kbytes$, $r = 3.0Mb/s$ and $p = 25Mb/s$. In this scenario the losses observed at the first leaky bucket are of about 6%, thus the true average rate (r_{OUT}) is around 2.82 Mb/s. The end-to-end delay bound requirement for each GS flow is 100 ms. The clearing rates R and buffer sizes needed to satisfy this requirement are given in table 1. The BR and BE flows cross all switches. All inter-switch and access links have bandwidths of 51 Mb/s and are assumed to be error free.

length	R (Mb/s)	weight	B (# cells)
5	8.4603	0.16589	1804
4	8.4562	0.16581	1804
3	8.4521	0.16573	1804
2	8.4480	0.16565	1804

inter-switch link	total R (Mb/s)	link reservation (%)
S_1-S_2	50.7372	99.49
S_2-S_3	50.7454	99.50
S_3-S_4	50.7454	99.50
S_4-S_5	50.7372	99.49

Table 1: clearing rate (R), weight (W), buffer size (B), link utilization.

To verify the feasibility of guaranteeing an average rate to BR flows, we use 5 BR flows, with average rates of 12Mb/s, 9Mb/s and 6Mb/s for the first three and 1.5 Mb/s for the last two. In this way, we push network utilization up to its maximum while respecting the inequality expressed in equation 3. Thus, the weight associated to each BR flow is $(\alpha_i \times 4.66e-3)$ $, i = 1..5$, where α_i represents the weighted share of the bandwidth left by the GS flows after their clearing rate has been reserved.

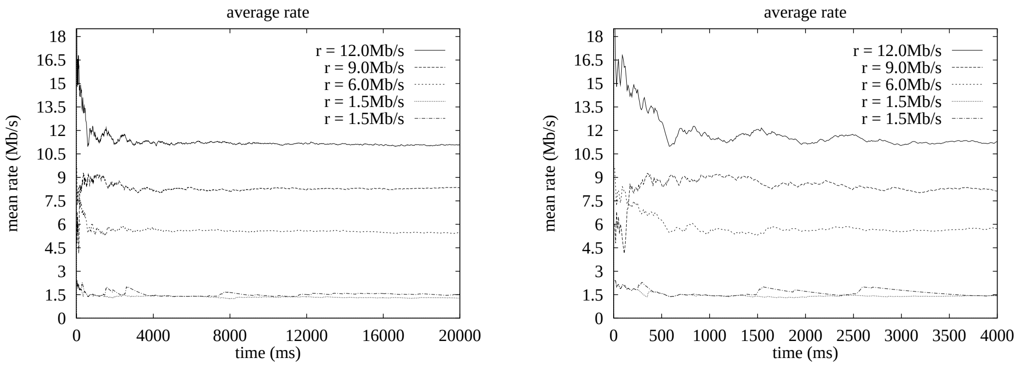


(a) GS flow delay

(b) BR flow delay

Figure 7: delay.

The simulation results are shown in figures 7 and 8. Curve 7(a) shows the delay experienced by GS packets belonging to a single flow. It demonstrates that GS flows are not affected by BR flows since the end-to-end delay bound of 100 ms is always respected. Curve 7(b) depicts the delay behavior of a BR flow. As expected, its packets endure a much greater delay than the one experienced by GS packets. Curves 8(a) and 8(b) show, respectively, the long and short-term average rates obtained by the five BR flows. The former illustrates how the average rate for the BR flows is guaranteed on a long-term basis. The latter shows that the average rates converge quite rapidly (after about 3000 ms) to the desired values.



(a) BR flows long-term mean rate

(b) BR flows short-term mean rate

Figure 8: mean rate.

5 Conclusions

The contribution of this paper is twofold. First, we propose a QoS-capable architecture for IP switching. This architecture mainly aims at providing flexible and effective integration of per-flow QoS with the fast capabilities of IP switching devices. The objective is to build a framework capable to respond to the more and more stringent requirements of the new Internet applications. In a second time, we define a WFQ-based strategy for the support of a new service class (BR). The purpose of this class is that of recovering the unused portion of the bandwidth reserved to GS flows without affecting their delay bounds and while giving long-term rate guarantees to other flows (such as web traffic) which are less sensitive in terms of delay. Based on the encouraging results obtained via simulation, the next step in our research work concerns the implementation of a prototype of our theoretical model.

References

[1] P. Newman, G. Minshall, and T. Lyon, "IP Switching: ATM under IP," *submitted to IEEE/ACM Transactions on Networking.*

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine*, pp. 8–18, Sept. 1993.

[3] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. C. Liaw, T. Lyon, and G. Minshall, "Ipsilon's General Switch Management Protocol Specification Version 1.1," RFC 1987, Aug. 1996.

[4] P. Newman, W. L. Edwards, R. Hinden, E. Hoffman, F. C. Liaw, T. Lyon, and G. Minshall, "Ipsilon's Flow Management Protocol Specification for IPv4 Version 1.0," RFC 1953, May 1996.

[5] S. Shenker, C. Partridge, and R. Guérin, "Specification of Guaranteed Quality of Service," draft-ietf-intserv-guaranteed-svc-08, Internet Draft, Feb. 1997.

[6] L. Georgiadis, R. Guérin, V. Peris, and R. Rajan, "Efficient Support of Delay and Rate Guarantees in an Internet," in *Proceedings of ACM SIGCOMM'96*, (Stanford, CA), Aug. 1996.

[7] S. Lin and N. McKeown, "A Simulation Study of IP Switching," in *Proceedings of ACM SIGCOMM'97*, (Cannes, France), Sept. 1997.