# The Renegotiable Variable Bit Rate Service: characterisation and prototyping*

Silvia Giordano† Jean-Yves Le Boudec

**Abstract:** *The traffic generated by multimedia applications presents a high degree of burstiness that can hardly be described by a static set of traffic parameters. We present a dynamic QoS negotiation scheme applied to a prototype application that provides temporized data transfer. The dynamic and efficient usage of the resources can be reached with the introduction of the renegotiable variable bit rate (RVBR) service, which is based on the renegotiation of the traffic specification. We describe and discuss the RVBR service and how it applies to resource reservation for Internet traffic with RSVP. We propose an architecture design that we evaluate by accomplishing a prototype implementation, whose performance is measured with temporized file transfer using real MPEG2 video traces. The results we obtained indicate that renegotiation is an efficient mechanism for accommodating traffic fluctuations over the burst timescale and that the RVBR service can be easily implemented in real applications, using available technology.*

**Keywords:** Shaping system, renegotiation, VBR parameters, resources optimisation, RSVP, prototype implementation.

## 1 Introduction

We consider the Renegotiable Variable Bit Rate (RVBR) service, defined as a variable bit rate service whose parameters are changed at periodic renegotiation moments. An example of this service is the Integrated Service of the IETF with the Resource reSerVation Protocol (RSVP), where the negotiated contract may be modified periodically [1]. A flow using the RVBR service is constrained by two leaky buckets: one defines the peak rate, the other defines the sustainable rate and the burst tolerance. We consider a basic scenario where fresh input traffic is shaped in order to satisfy the leaky bucket constraints. We assume that shaping is done by an optimal shaper, with a limited buffer size $X$ [2]. The input traffic may be generated by one source, or it may be an aggregate of sources, in which case the shaper models a service multiplexer. Using VBR in a shaper may be advantageous in all cases where the input traffic is bursty and the network is able to achieve a statistical multiplexing gain on many such input flows [3].

In our model scenario, the RVBR parameters are renegotiated periodically; at every renegotiation, there is a tradeoff to be made between the various parameters, which define the two leaky buckets in the next interval. In [4] we analyse this tradeoff and propose an algorithm (*localOptimum1*) to select, for the next interval, the parameters that minimise a given linear cost function.

Our main goal in this paper is to validate our service by means of simulations and to prove its applicability to real scenarios through trials with a prototype implementation.

RVBR service is based on the output characterisation of shaper systems in terms of the network calculus theory [5], [6], the related characterisation of the VBR service [7], [8] and the study of the optimisation problem for the VBR service [9], [8].

Renegotiation was first specified in ATM networks for CBR class service [10] and only very recently for VBR class service [11]. In the reservation protocol for Integrated Services Internet networks, namely RSVP, a source is requested to refresh the reservation at given times. However, this is not intended as a mechanism for modifying the reservation parameters only, but rather as the general approach for managing the reservation state in routers and hosts [12].

Renegotiable VBR services are also studied in [13],[14],[15]; there the focus is on describing a given traffic with as few leaky buckets as possible, and thus applies to the optimisation of a network offering the RVBR service. In contrast, the approach studied here focuses on the customer side of the RVBR service, and provides an analysis of the various tradeoffs that can be made. It also differs in the systematic use of network calculus that results in simple, efficient algorithms that can easily be implemented in real applications, using available technology. In Section 2, we summarize the mathematical model behind the RVBR service based on the class

*the prototyping work was performed in collaboration with Finsiel S.P.A., Italy

†Contact author: Email. silvia.giordanoepfl.ch Tel. +41 21 693 67 48 Fax. +41 21 693 6610

of time varying leaky-bucket shapers. Then, in Section 3 we describe the RVBR service. We also illustrate the algorithm (*localOptimum1*) and how it applies to the RSVP scenario. In Section 4 in order to validate the RVBR service, we simulate this algorithm in a typical real case: transmission of MPEG2-encoded video using the IntServ Controlled Load service [16] with the RSVP [12], [17] reservation protocol. The results of our simulation suggest that renegotiation allows the better use of network resources and that in protocols such as RSVP, where there is no additional cost for signaling (or so we mainly assume), it is better to renegotiate.

Then, in Section 5 we propose an architecture design that we evaluate by accomplishing a prototype implementation, whose performance is measured with real MPEG2 video traces. Our prototype uses RSVP reservation protocol with Controlled-Load service integrated with the RVBR service. To integrate the RVBR, we modified some functions used by the RVBR service, which originally worked with the exact traffic. In this way we use and manage, inside our implementation, a reduced number of traffic information: the exact traffic $R(t)$ for $t$ in $I_i$ is substituted by upper bound functions. The trials were performed between two PCs connected to a shared LAN. We carried this study on a real network with the prototype client and server exchanging RSVP messages containing a *Tspec* renegotiated according to the RVBR service. In the trials, we varied the network parameters and reallocation time. We identified two network configurations: (1) a shared Ethernet with medium load and (2) a switched Ethernet (this was simulated by performing the trials on the Ethernet unloaded).

The measurement performed on our testbed with real MPEG2 video traces showed the benefits of applying the renegotiation. In fact, our results indicate that renegotiation is an efficient mechanism for allowing better use of network resources at the very low price of implementing a service like RVBR. In particular, renegotiation shows to be an efficient mechanism for accommodating traffic fluctuations over the burst time-scale and, to this aim, RVBR service can be easily implemented in real applications, using available technology.

## 2 Time Varying Leaky Bucket Shapers

A shaper is a system that stores incoming bits in a buffer and delivers them as early as possible, while forcing the output to be constrained with a given arrival curve. A shaper is time invariant if the traffic constraint is defined by a fixed arrival curve; it is time varying if the condition on the output is given by a time varying traffic contract. This occurs, for example, with renegotiable variable bit rate (RVBR) services.

The RVBR service is characterised by using a special class of time varying shaper systems, which we call the *time varying leaky-bucket shapers*. This is defined by a fixed number $J$ of leaky bucket specifications with bucket rate $r^j$ and bucket depth $b^j$, where $j = 1, \ldots, J$ and a shaping buffer of fixed capacity $X$. At specified time instants $t_i$, $i = 0, 1, 2, ..$, the parameters of the leaky buckets are modified and $I_i = (t_i, t_{i+1}]$ represents the i-th interval.

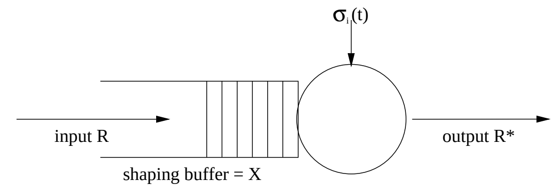



**Figure 1.** Reference Model for a time varying leaky-bucket shaper. The traffic shaping at time $t \in I_i$ is done at source according to the service curve $\sigma_i$ valid in $I_i$.

Inside each interval the parameters of the system do not change. The parameters of the j-th leaky buckets valid in the interval $I_i$ are indicated by $(r_i^j, b_i^j)$. The combination of those parameters takes the form of the shaping function $\sigma_i$ in $I_i$, defined as

$$\sigma_i(u) = \min_{1 \le j \le J} \{\sigma_i^j(u)\} = \min_{1 \le j \le J} \{r_i^j \cdot u + b_i^j\}$$

A time varying leaky-bucket shaper is completely defined by the number $J$ of leaky buckets; the time instants $t_i$ at which the parameters changes; the buckets parameters $(r_i^j, b_i^j)$, for each $j$ and each interval $I_i$; the fixed shaping buffer capacity $X$.

$R(t) : \mathbb{R}^+ \to \mathbb{R}^+$ is the *input traffic function* and represents the amount of traffic that has entered in the system in time interval $[0, t]$. $R$ is the traffic before the shaping. $R^*(t)$ is the *output function* that represents the number of bytes seen on the output flow in time interval $[0, t]$. $R^*$ is the traffic after the shaping.

At the transient times between two adjacent intervals, the time varying leaky-bucket shaper keeps the level of the buckets and restart from that level at the next interval ("no reset" approach). This is validated by the fact that, in the special case where the time varying leaky-bucket is constant, the system is identical to the ordinary, time invariant, leaky bucket shaper [9], [8]. Moreover, it is in line with the Dynamic Generic Cell Rate Algorithm (DGCRA) used to specify conformance at the UNI for the available bit rate (ABR) service of ATM [18], [19]. In [4] the practical implications of the "no-reset" approach are studied in terms of losses.

The input-output characterisation of the time varying leaky bucket shapers in the interval $I_i$ is given in Theorem 2 of [4]:

$$R^*(t) = \min \left[ \sigma_i^0(t - t_i) + R^*(t_i), \inf_{t_i < s \le t} \{\sigma_i(t-s) + R(s)\} \right] \quad (1)$$

where $\sigma_i^0$, representing the service curve taking into account the initial conditions at time $t_i$, is defined as

$$\sigma_i^0(u) = \min_{1 \le j \le J} \left[ r_i^j \cdot u + b_i^j - q^j(t_i) \right]$$

and $q_j(t_i)$ is the bucket level of the j-th bucket, defined as, at time $t \in I_i$

$$q^j(t) = \max \left[ \begin{array}{l} \sup_{t_i < s \le t} \{R^*(t) - R^*(s) - r_i^j \cdot (t-s)\}, \\ \left[ R^*(t) - R^*(t_i) - r_i^j \cdot (t - t_i) + q^j(t_i) \right] \end{array} \right] \quad (2)$$

Moreover, $w(t_i)$ is the backlog in the shaping buffer at time $t \in I_i$

$$w(t) = \max \left[ \begin{array}{l} \sup_{t_i < s \le t} \{R(t) - R(s) - \sigma_i(t-s)\}, \\ R(t) - R(t_i) - \sigma_i^0(t - t_i) + w(t_i) \end{array} \right] \quad t \in I_i \quad (3)$$

The result in Equation (1) has an intuitive interpretation that has a general validity. The output of a time varying shaper in any interval is either driven by a combination of the shaping function and the past history, or is computed by taking into account the level of the shaping buffer at the beginning of the interval. This definition is evidently recursive because it depends on the output and on the past history, which are themselves computed with the same formulas.

## 3 The RVBR Service and its application to RSVP

The input-output characterisation of the RVBR service comes straightforward as a special case of the time varying leaky bucket shaper. An RVBR source is a time varying leaky-bucket shaper with two renegotiable leaky buckets ($J = 2$); one with rate $r_i$ and depth $b_i$ and the second with rate $p_i$ and depth always equal to zero, plus a buffer of fixed size $X$. Therefore, in the Equations (1), (2) and (3), $\sigma_i$ and $\sigma_i^0$ are given by

$$\sigma_i(u) = \min(p_i \cdot u + b_i^1, r_i \cdot u + b_i^2) \quad (4)$$

$$\sigma_i^0(u) = \min(p_i \cdot u + b_i^1 - q^1(t_i), r_i \cdot u + b_i^2 - q^2(t_i)) \quad (5)$$

In real life, examples of this service are traffic shaping done at source sending over VBR connections as defined in [11] and Internet traffic that takes the form of IntServ specification with RSVP reservation [12], [17].

In RSVP, the sender sends a PATH message with a *Tspec* object that characterises the traffic it is willing to send. If we consider a network that provides a service as specified for the Controlled Load service (CL) [16], the *Tspec* takes the form of a double bucket specification [20] as given by the RVBR service. There is a peak rate $p$ and a leaky bucket specification with rate $r$ and bucket size $b$. Additionally, there is a minimum policed unit $m$ and a maximum packet size $M$. We ignore $m$ and $M$, which are assumed to be fixed. With RSVP as reservation protocol, the reservation has to be periodically refreshed. The suggested period is 30 seconds. Therefore $p$, $r$ and $b$ need to be reissued at each renegotiation time. There is no additional signaling cost in applying a *Tspec* renegotiation at that point, even if there is some computational overhead due to the computation of the new parameters, or to the call admission control, etc. It is important to note here that, contrary to the negotiation of a new connection, with the renegotiation the reservation is never interrupted.

If the requested traffic specification cannot be supported by the network, the old traffic specification is restored and the network may not be able to accommodate the next traffic. Mechanisms to prevent this failure from occurring are still under study. Here we assume that either (1) the *Tspec* is accepted all over the network, as well as at the destination, such that the source can transmit conforming to its desired traffic specification or (2) the source can adapt to transmit with the old *Tspec*, even if at the price of a reduced quality.

We assume that at any time $t_i = 30 \cdot i$ the application knows (because pre-recorded or predicted) the traffic for the next 30 seconds. We further assume to know the cost to the network of the *Tspec*s (indicated by the cost function $u \cdot r + b$) and the upper bound to the bucket size $b_{max}$ and to the bucket rate $r_{max}$. The backlog $w(t_i)$ and the bucket level $q(t_i)$ can be measured in the system.

In order to use RVBR service for RSVP with CL service scenario, we are faced with the problem of computing the leaky bucket parameters. Therefore, we describe the case of a source that wants to reserve the resources for the next interval. For the RVBR service, this is equivalent to the problem of computing the RVBR parameters for the next interval. Here we present the approach developed in [4] that we also use in the simulations. As we will see in Section 5, in real cases, this approach can require some modifications.

The basic architecture of the sender node is described in Figure 2. From Equations (1) and (3) it comes

$$\begin{array}{ll} R(t) - R(s) \le \sigma_i(t-s) + X & t \in I_i, t_i < s \le t \\ R(t) - R(t_i) \le \sigma_i^0(t - t_i) - w(t_i) + X & t \in I_i \end{array}$$

These equations give a necessary and sufficient condition for a minimum $p_i$. This, in analogy to the work in [8] can

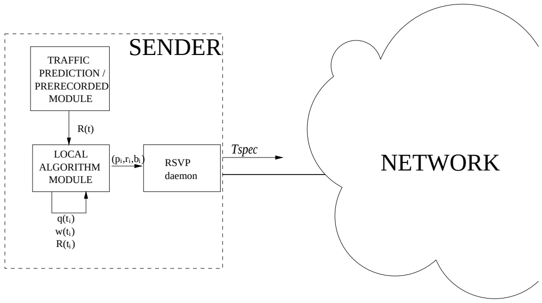



**Figure 2.** A basic architecture to support the usage of the local scheme for RSVP with CL service reservation: each 30 seconds $R(t)$ is predicted and used to compute the optimal $p$, $r$ and $b$ to generate the new *Tspec*.

be seen as the *effective bandwidth* of the arrival stream in $I_i$ taking into account the backlog at time $t_i$. Therefore, given that $p_i$ is computed independently from $r_i$ and $b_i$, the problem of finding a complete optimal parameter set $(p_i, r_i, b_i)$ for the RVBR service is reduced to the problem of finding the optimal parameters $r_i$ and $b_i$.

This optimisation problem, when the cost function is linear: $c(r_i, b_i) = u \cdot r_i + b_i$, for fixed values of $u$, is modeled in [4] in form of an algorithm (*localOptimum1*).

**Algorithm 1** localOptimum1

**if** $b_{max} < \sup_{s \in I}\{\beta_i(s) - r_{max} \cdot s - X\}$ **then** there is no feasible solution;

**else** {

$$p_i = \max \left( \begin{array}{l} \sup_{t,s \in I_i} \frac{R(t) - R(s) - X}{t - s}, \\ \sup_{t \in I_i} \frac{R(t) - R(t_i) - X + w(t_i)}{t - t_i} \end{array} \right)$$

**if** $u \leq 0$ **then** {

$x_0 = min(r_{max}, p_i)$;

}

**else** {

$$x_0 = \sup_{s \in I} \frac{\beta_i(s) - \beta_i(u)}{s - u};$$

$$x_A = \sup_{s \in I, s > 0} \frac{\beta_i(s) - X - b_{max}}{s};$$

$$x_B = \sup_{s \in I, s > 0} \frac{\beta_i(s) - X}{s};$$

**if** $(x_0 \geq \min(x_B, r_{max}, p_i))$ **then** $x_0 = \min(x_B, r_{max}, p_i)$;

**else if** $(x_0 \leq x_A)$ **then** $x_0 = x_A$;

}

$r_i = x_0$;

$b_i = \sup_{s \in I}\{\beta_i(s) - X - s \cdot x_0\}$;

Then, using this algorithm, we compute the *Tspec* that the sender will send at the next renegotiation time[1].

[1] In this context, we do not consider delay issues that are matter of further study.

# 4 RVBR Service validation for RSVP

In this section, we describe how we use the local algorithm to simulate a typical real case: transmission of MPEG2-encoded video using the IntServ Controlled Load service with the RSVP reservation protocol.

In our simulations, we use a 4000 frame-long sequence that conforms to the ITU-R 601 format ($720 * 576$ at 25 fps). The sequence is composed of several video scenes that differ in terms of spatial and temporal complexities. It was encoded in an open-loop variable bit rate (OL-VBR) mode, as interlaced video, with a structure of 11 images between each pair of I-pictures and 2 B-pictures between every reference picture. For this purpose, the widely accepted TM5 video encoder [21] was used.

The traffic generated by the video is transported by a trunk regulated by a RVBR service $(p, r, b)$ with shaping buffer $X$. In this context we do not consider any scheduling issues, which is the subject of ongoing work. Therefore we assume that the video, with a total size of 550 Mbits, is transmitted in 163 seconds (25 frames pro second). The cost function is linear with $u$. For the sake of brevity, we illustrate here only one scenario. Other scenarios are given in [22] and [4]. Here we have that $X = 40$ Mbits, $r_{max} = 5$ Mbps, $b_{max} = 9$ Mbps and $u = 1$. The initial conditions are: $q(0) = 0$ and $w(0) = 0$. The file is pre-recorded and, given that we do not enter in scheduling matters, we know $R(t)$ for all $t$. At time $t_i$ we know $R^*(t)$ for $t \leq t_i$, we measure $w(t_i)$, $q(t_i)$ and compute $\beta_i(t)$. We obtain the optimal shaper parameters by applying the algorithm *localOptimum1* at Section 3 that we use to generate the *Tspec* that the sender will send at the next renegotiation time.

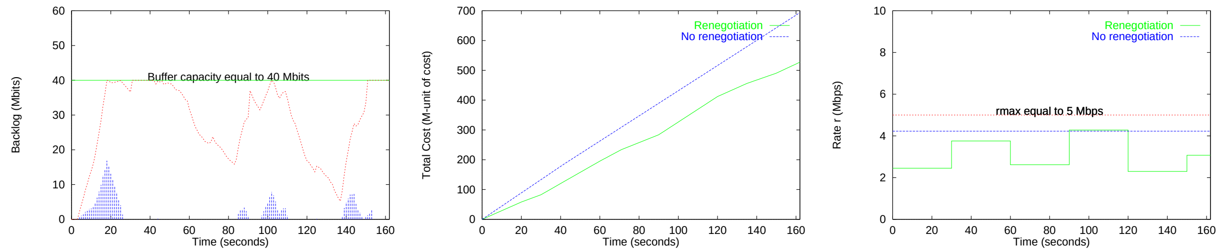
In Figure 3(a) we plot the backlog for the scenario in both cases where we apply the renegotiation and where we do not renegotiate [2]. We observe that in the beginning the curves representing the two approaches do not differ much. This is because the traffic is very high in the first 30 seconds and both traffic specifications conform to this traffic.

After this period, the traffic rate decreases. The case without renegotiation has to keep the traffic specification negotiated at time $t = 0$, even if it is no longer adequate for the current demand. The resources allocated in the network are so large that it is possible to empty the buffer and thereafter the buffer is rarely used.

The curve for the case where we used the RVBR service shows that the buffer is much better utilised, because the traffic specification decreases in the next intervals.

Therefore, with the RVBR service the resources in the network are utilised much better. In fact when the

[2] Even in this case we compute the optimal traffic specification as introduced in [8].




(a) Shaping buffer used with renegotiation (white area) and without renegotiation (black area)

(b) Cost of allocating a renegotiated traffic specification and a traffic specification without renegotiation

(c) Evolution of the rate $r$

**Figure 3.** Comparison between the renegotiation case and the case without renegotiation of the shaping buffer, the cost of the traffic specification and the evolution of the rate $r$. The cost of the traffic specification is given in "millions of unit of cost" (M-unit of cost).

buffer is almost always filled, the output is conforms to the traffic specification and this means that all the resources in the network are optimally used. The usage of the buffer with renegotiation is 58%, while without renegotiation it is 13%.

In the graphs in Figure 3(b) we compare the two approaches in terms of the cost of the traffic specification to the network. The cost of the traffic specification is given in terms of the linear cost function used by the RVBR service in order to compute the optimal traffic parameters. Moreover, there is also a substantial advantage from the cost point of view in reallocating, because the cost of the traffic specifications is in general smaller.

Figure 3(c) illustrates the fact that with renegotiation we can optimise the resources requested to the network and therefore at the end the total $r$ and $b$ allocated in this case are in general smaller. We also notice that inside an interval the RVBR service might allocate a *Tspec* that is larger than the one used when not renegotiating. This occurs when the traffic is very bursty and the buffer is full from the previous interval, i.e. at the forth interval $(90 - 120$ seconds).

## 5 RVBR Service implementation in an RSVP-based prototype

For our trials, we used a prototype application that provides RVBR features via RSVP and we implemented it under Microsoft NT 4.0 with RSVP by Intel. The prototype realises a client-server application for data-transfer regulated by pre-defined temporisation. This was an intermediate step towards a VoD application that isRVBR-RSVP based. Currently this application is completed and trials are in progress thus confirming the result of this section. The main role of this prototype is twofold: (1) to analyse the issues of applying the RVBR service in a real application and (2) to validate renegotiation against negotiation in a real scenario. In the prototype, as well as in the VoD application, the access to the traffic information is done through communication between modules. Therefore, the transfer of the complete information on $R(t)$ was not reasonable. We introduce an approximation of $R(t)$, which, requiring less space, can be more easily supported. By substituting the original functions with the following functions in the algorithm *localOptimum1*, we can use the RVBR service. The new functions are $\alpha_i$ and $\alpha_i^0$

$$\alpha_i(u) = \min(p_i^\alpha \cdot u, r_i^\alpha \cdot u + b_i^\alpha)$$

where

$$p_i^\alpha = \sup_{t,s \in I_i} \frac{R(t) - R(s)}{t - s}$$

$$r_i^\alpha = \frac{R(t_{i+1}) - R(t_i)}{t_{i+1} - t_i}$$

$$b_i^\alpha = \sum tR(t) - r_i^\alpha \cdot t$$

and

$$\alpha_i^0(u) = \min(p_i^{\alpha^0} \cdot u, r_i^\alpha \cdot u + b_i^\alpha)$$

where

$$p_i^{\alpha^0} = \sup_{t \in I_i} \frac{R(t) - R(t_i)}{t - t_i}$$

With these two new functions we rewrite $\beta$ and $p_i$ as follows:

$$\beta_i(s) = \max(\alpha_i(s), \alpha_i^0(s) + w(t_i) + q(t_i)$$

$$p_i = \max\left( \sup_{s \in I_i} \frac{\alpha_i(s) - X}{s}, \sup_{s \in I_i} \frac{\alpha_i(s) + w(t_i) - X}{s} \right)$$

As already mentioned, these trials aim at verifying the applicability of the RVBR service and its ability for providing a better allocation of the resources in a real network, taking into account the overhead (time consuming) introduced by the support of re-negotiation. For the first goal we compared, by means of simulations, the results obtained when applying the original RVBR algorthm and when the approximated version was used. The results, not shown here, confirm that the approximation performs well, as expected, due to the “leaky-bucket” nature of the approximated functions in line with the type service.

Than we performed trials between two PCs connected to a shared LAN without routers. We varied network configuration and reallocation time. We used a shared Ethernet with medium load and a switched Ethernet (simulated performing the related trials on the unloaded Ethernet).

The following figures show a comparison of allocated resources varying the reallocation time. We used the same MPEG2 traffic described in the previous section. The graphic in Figure 4 is related to the bucket rate $r_i$ and peak rate $p_i$ allocation with reallocation time respectively set at 30 seconds, 60 seconds, and without reallocation in the case of a shared Ethernet. The bucket size was not significant because it is limited by the the buffer capacity of each NIC (256KB). In this case (shared Ethernet) we limited the bucket capacity to half of the NIC capacity to take into account the problems derived usually from a legacy Ethernet.

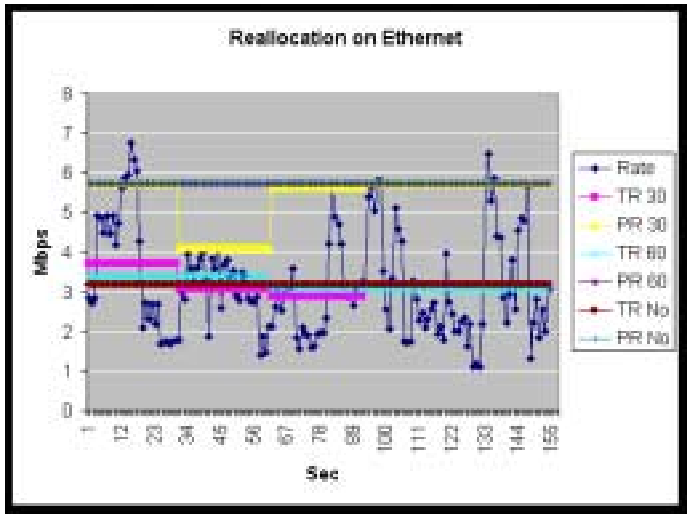



**Figure 4.** Shared Ethernet: The bucket rate is indicated with TR and the peak rate with PR.

The line related to the peak rate is constant in the case of 60 seconds and no reallocation, while it is piecewise in the 30 seconds case and it lies under the others. We can observe that the peak rate is quite constant in the case of 60 seconds. The reallocation at 30 seconds requires a narrow bandwidth with a peak value always under the others.

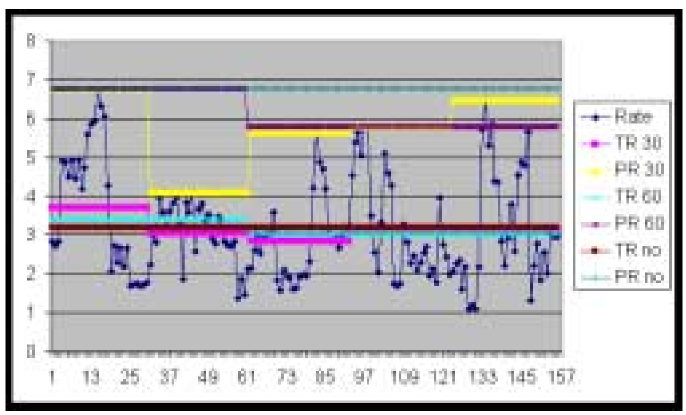



**Figure 5.** Switched Ethernet: The bucket rate is indicated with TR and the peak rate with PR.

The graphic in Figure 5 is related to a Switched Ethernet. In both the 30 and 60 seconds reallocation cases, we can observe a gain derived from the reallocation of resources in terms of bandwidth (derived from bucket rate $r_i$ and peak rate $p_i$ parameters) compared to the negotiation case. An interesting analysis is related to the time needed for reallocation. This is critical because it could affect the correct behaviour of the system. A too slow allocation could loose synchronisation between Control Plan and User Plan. The values measured in our trials are lower than 0.1 second. The average time needed for the reallocation related to all performed experiments is 62 msec.

## 6 Conclusion

We consider the problem of modifying the traffic specification of a connection while keeping the connection active in order to support the traffic QoS requirements while efficiently allocating the network resources. For a two leaky bucket characterised traffic, the RVBR service can be used to solve these optimisation aspects.

We illustrate how the RVBR service can be applied to RSVP *Path* message generation. This is based on the algorithm proposed for the local optimisation problem for the RVBR Service. A numerical example of this is given in Section 4, where we compare the performance of transmitting a MPEG2 video trace both with and without renegotiation. The results of our simulation (see Figure 3) suggest that renegotiation allows for the better use of network resources and that in protocols such as RSVP, where there is no additional cost for signaling (or so we assume), it is better to renegotiate.

Then, we presente the integration of the RVBR service in a prototype application, whose performance is measured with temporized file transfer using real MPEG2 video traces. This application uses RSVP reservation protocol with Controlled-Load service integrated with the RVBR service. We modified some functions used by the RVBR service in order to use and manage a reduced number of traffic information inside our implementation.

We carried this study on a real network with the prototype client and server exchanging RSVP messages containing a Tspec renegotiated according to the RVBR service. The measurement performed on our testbed with real MPEG2 video traces show the benefits of applying the renegotiation. In fact, our results indicate that renegotiation is an efficient mechanism for allowing the better use of network resources at the very low price of implementing a service like RVBR. Some important aspects that were neglected in this first release are currently under study and will be included in future releases. Among them, but not limited to, we consider the introduction of timing constraints to select the right QoS, the synchronization between User Plan and Control Plan and recovery mechanisms in case of fault.

The results we obtained show that the RVBR service can be easily and efficiently adopted by video applications that require strict guaranteed service: when a reasonable renegotiation, (i.e. with renegotiation periods of about 30 seconds, default RSVP) is used, the network resource utilization is better and it works using available technology.

## References


[1] R. Guérin and V. Peris, "Quality-of-service in packet networks - basic mechanisms and directions," *Computer Networks and ISDN, Special issue on multimedia communications over packet-based networks*, 1998.

[2] C. Chang, "On deterministic traffic regulation and service guarantee: A systematic approach by filtering," *IEEE Transactions on Information Theory*, vol. 44, pp. 1096–1107, August 1998.

[3] A. Ziedinsh and J.-Y. Le Boudec, "Adaptive CAC Algoritms," *in proceedings of ITC 15*, 1996.

[4] S. Giordano, J.-Y. Le Boudec, "On a Class of Time Varying Shapers with Application to the Renegotiable Variable Bit Rate Service," Technical Report SSC/1998/035, DI-EPFL, CH-1015 Lausanne, Switzerland, 1998. http://lrcwww.epfl.ch/ ~giordano/tvShaperv33.ps.

[5] J.-Y. Le Boudec, "Network Calculus Made Easy," Technical Report 96/218, DI-EPFL, CH-1015 Lausanne, Switzerland, December 1996.

[6] J.-Y. Le Boudec and P. Thiran, "Network Calculus viewed as a Min-plus System Theory applied to Communication Networks," Technical Report 98/276, DI-EPFL, CH-1015 Lausanne, Switzerland, April 1998.

[7] J.-Y. Le Boudec, "Application of Network Calculus To Guaranteed Service," Technical Report 97/251, DI-EPFL, CH-1015 Lausanne, Switzerland, November 1997.

[8] J.-Y. Le Boudec, "Network Calculus, Deterministic Effective Bandwidth, VBR trunks," *IEEE Globecom 97*, November 1997.

[9] S. Giordano, J-Y. Le Boudec, P. Oechslin, S. Robert, "VBR over VBR: the Homogeneous, Lossfree Case," *INFOCOM97*, 1997.

[10] ITU Telecommunication Standardization Sector - Study group 13, *ITU-T Recommandation Q.2963.1. : Peak cell rate modification by the connection owner*, 1996.

[11] ITU Telecommunication Standardization Sector - Study group 13, *ITU-T Recommandation Q.2963.2. Broadband integrated services digital network (B-ISDN) digital subscriber signalling system No. 2 (DSS 2) connection modification - Modification procedure for sustainable cell rate parameters*, 1998.

[12] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, *RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.* IETF, September 1997.

[13] H. Zhang, E. Knightly, "A New Approach to Support Delay-Sentive VBR Video in Packet-Switching Networks.," *In Proceedings 5th Workshop on Network Operating System Support for Digital Audio and Video (NOSSDAV)*, April 1995.

[14] H. Zhang, E. Knightly, "RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video," *ACM Multimedia Systems Journal*, 1997.

[15] J. Liebeherr, D. Wrege, "An Efficient Solution to Traffic Characterization of VBR Video in Quality-of-Service Network," *to appear in ACM/Springer Multimedia Systems Journal*, 1998.

[16] J. Wroclawski, *RFC2211: Specification of Controlled-Load Network Element Service.* IETF, September 1997.

[17] J. Wroclawski, *RFC2210: The Use of RSVP with IETF Integrated Services.* IETF, September 1997.

[18] The ATM Forum, *ATM User-Network Interface (UNI) Signalling Specification, Version 4.0*, 1996. ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.000.ps.

[19] W. P. 4, "Specification of Integrated Traffic Control Architecture," Deliverable Del06, ACTS project AC094 EXPERT, September 1997.

[20] S. Shenker, J. Wroclawski, *RFC2216: Network Element Service Specification Template.* IETF, September 1997.

[21] C. Fogg, "mpeg2encode/mpeg2decode," *MPEG Software Simulation Group*, 1996.

[22] S. Giordano, J.-Y. Le Boudec, "QoS based Integration of IP and ATM: Resource Renegotiation," *In Proceedings of 13th IEEE Computer Communications Workshop*, 1998. http://lrcwww.epfl.ch/ ˜giordano/publications.html.