Efficient Support of Delay and Rate Guarantees in an Internet

L. Georgiadis*, R. Guérin, V. Peris and R. Rajan
IBM T. J. Watson Research Center
P.O.Box 704, Yorktown Heights, NY 10598
leonid@eng.auth.gr, {guerin,vperis,raju}@watson.ibm.com

Abstract

In this paper, we investigate some issues related to the efficient provision of end-to-end delay guarantees in the context of the Guaranteed (G) Services framework [16]. First, we consider the impact of reshaping traffic within the network on the end-to-end delay, the end-to-end jitter, as well as per-hop buffer requirements. This leads us to examine a class of traffic disciplines that use reshaping at each hop, namely rate-controlled disciplines. In this case, it is known that it is advantageous to use the Earliest Deadline First (EDF) scheduling policy at the link scheduler [8]. For this service discipline, we determine the appropriate values of the parameters that have to be exported, as specified in [16]. Subsequently, with the help of an example, we illustrate how the G service traffic will typically underutilize the network, regardless of the scheduling policy used. We then define a Guaranteed Rate (GR) service, that is synergetic with the G service framework and makes use of this unutilized bandwidth to provide rate guarantees to flows. We outline some of the details of the GR service and explain how it can be supported in conjunction with the G service in an efficient manner.

1 Introduction

The topic of this paper is that of providing service guarantees in the context of an internet. Service guarantees are inherently required by many applications, e.g., voice and video, as well as by an increasingly large number of users as they start relying more and more on an internet infrastructure to conduct daily business. A direct result of such an increased dependency is the need for stronger assurances that the necessary services will be available at "all" times. Here, availability is not only a statement on physical reliability, but also on the quality, e.g., latency, throughput, etc., of the service being provided. This represents a significant departure from the traditional best-effort service model that internets have been relying on until now. The addition of new service definitions to support quality-of-service (QoS) guarantees has been addressed by the IETF Int-Serv working group, and is reflected in a number of service specifications [16, 17, 18, 20].

In the paper, we focus initially on the Guaranteed (G) service specification of [16] that defines the provision of hard end-to-end delay guarantees to individual flows. Our principal motivation is to bring out an important design aspect of providing such service guarantees, namely the impact and potential benefits of reshaping flows inside the network to some pre-determined envelope. We establish that in the context of the G service, irrespective of the scheduling policy used at the Network Elements (NE), the introduction of appropriate shapers does not affect the end-to-end delay bound that the service guarantees. Average delays will obviously be impacted, but this should be more than compensated by the fact that buffer requirements at each NE as well as end-to-end jitter will be significantly reduced by the use of reshaping. In the paper, we motivate and quantify these reductions, and precisely specify the reshaping that is permissible to avoid affecting the end-to-end delay guarantee. Based on the benefits offered by reshaping within the network, we focus next on a class of service disciplines, Rate Controlled Service (RCS) disciplines, that rely heavily on reshaping, i.e., use it at every NE. We briefly describe the operation of the RCS discipline in the context of the G service, and point to the fact that in order to provide similar delay guarantees as the Weighted Fair Queueing (WFQ) discipline, some care must be exercised in the choice of the reshaping parameters.

The rather stringent quality of service requirements of G service flows leads to reservations that are in excess of the mean rates of these flows, resulting in low link utilization in networks that support solely the G service. We illustrate this under-utilization through a simple example with realisitic traffic and reservation parameters. This example motivates the need for a service which would provide less stringent guarantees than the G service, utilizing its unused bandwidth without any adverse impact on G service flows. The desirability of such a service has been expressed by many authors [2, 12, 10]. In this paper, we propose the Guaranteed Rate (GR) service which provides the same rate guarantees as the G service, but with much looser guarantees as far as delay is concerned. When compared to the CL service, the GR service considers the requested service rate as a hard guarantee, so that reshaping of flows, which is precluded by the CL specification, is allowed. Another aspect that we expand on in the paper, is that the RCS dis-

*L. Georgiadis is on leave from IBM and is currently at the Electrical and Computer Engineering Department at the Aristotle University, Thessaloniki, 54006 GREECE

cipline allows synergetic implementations of both the G and GR services.

To summarize, the contributions of this paper are twofold:

- We apply the general results on rate-controlled service disciplines presented in [9] to the provision of the G service in an internet, and systematically investigate the costs and benefits of reshaping traffic at every node. The most significant of such benefits are a reduction in buffering required at switches, and an enhancement of the service through improved jitter control. The latter is notable since even if the G service does not explicitly target jitter control, this parameter is likely to be of importance to many applications, e.g., to size playback buffers. Hence, the ability to control this parameter in the context of the G service is of interest.

- We introduce the GR service which like the G service guarantees a certain service rate to applications, but unlike the G service does not provide any hard delay guarantees. This provides for a more efficient use of bandwidth, at the cost of allowing occasional large delays. In addition, we present a possible implementation of the GR service when using RCS disciplines at each node, that is synergetic with support for the G service.

The rest of this paper follows the structure outlined above. Section 2 provides a brief review of the G service, its operation, and the key parameters it relies on. The issue of reshaping is the topic of Section 3, where we explicitly characterize its impact on the end-to-end delay and jitter, and the buffer requirements at each network element. The use of reshaping in the context of the RCS discipline to support the G service is addressed in Section 4, where the potentially low link utilization imposed by the G service is also illustrated by means of an example. The use of this residual bandwidth to provide a GR service is covered in Section 5 for both the RCS and WFQ disciplines. A short conclusion summarizes the findings of the paper and points to some remaining open issues.

2 Guaranteed Services

In this section we briefly outline the workings of the G service specification [16]. In order to better motivate the specifications in [16], as well as provide a better understanding of the G service specification, we use RSVP as an example. RSVP is a resource reservation setup protocol designed for an internet that supports Integrated services [3]. A primary design goal for RSVP was to support multicast with receivers being able to add themselves to a multicast session at will. Thus RSVP is a receiver-initiated protocol, with the resource reservation being made by the receivers. For simplicity, in this paper, we consider an RSVP session that has a single sender[1] and multiple receivers in its distribution list. We only describe the reservation aspects of RSVP, and do not venture into describing the "filtering" of packets from different senders, so as not to digress from the main focus of this paper. In the rest of this paper, we use the term *flow* to refer to the stream of data traffic that is transported from a sender to a receiver.

Path messages are sent from the sender to all the receivers in the distribution list along the default routing path of the internet. These messages contain information about the flow, i.e., a *flowspec* that describes the flow's characteristics in terms of token bucket parameters [19]. In addition, they contain an *adspec*, that can be modified by the different NEs traversed by the *Path* message. A receiver who decides to join an RSVP session needs to send a *Resv* message that specifies the amount of resources that it wishes to reserve for itself. The receiver uses the information in the *flowspec* and the *adspec* that it has previously received to decide the level of resources that it needs to reserve. The *Resv* message sent by the receiver retraces the path of the *Path* message (details are in [3]) and establishes the required reservation.

The role of the G service specification is to allow the receiver to make an intelligent choice about the level of resources it needs to reserve in order to obtain an upper bound on the end-to-end packet delay. The amount of resources that need to be reserved are a function of:

- **User Characteristics** : This has to do with the *flowspec* of the flow and the end-to-end delay and/or throughput requirement of the receiver.

- **Network Characteristics** : These include factors like the number of hops on the path, the scheduling policy employed at each hop, the end-to-end latency that is present, etc.

2.1 User and Network Characteristics

The User Characteristics consist of the flowspec which can be provided in terms of token bucket parameters [19], and an *RSpec* which indicates the level of resources that have to be reserved for this flow. For now, it suffices to say that the *RSpec* is a rate, R; the full meaning of this rate will become clear when we describe the Network Characteristics. The token bucket characteristics for the *flowspec* are a triplet (b, r, p), where b is the token bucket size, r is the token accumulation rate, and p is the maximum peak rate of the flow. If the peak rate parameter, p, is unknown, it defaults to infinity. A few more parameters are signalled as part of the G service specification; for the purpose of this document the only other parameter that we use is the maximum packet size which we denote by L.

The Network Characteristics are signalled to the receiver in the *adspec* element which, in the case of RSVP, is carried in the *Path* messages that traverse the NEs along the path of the flow. At any given time there can be numerous *Path* messages passing through an NE, most of which do not result in any resources being reserved for the flow. Thus, it is necessary that the network characteristics signalled in the *adspec* element be independent of the other *Path* messages that may be passing through the NE. Also, it is possible that a large number of *Resv* messages from different flows reach a particular NE in a rather short span of time. So, basing the Network Characteristics on the current load at the NE can be equally meaningless. The solution adopted by the Integrated Services Working Group of the IETF is to define a characterization of the NE that is independent of the other flows that are passing through it, and is described next.

Network element, i, exports parameters C_i and D_i, that qualify the level of service that it can provide to flows that traverse it. These exported parameters are carried by the *adspec* element and are interpreted in the context of the reserved rate, R, that a potential receiver might reserve for a flow. Typically, C_i and D_i capture the deviation of the NE i, from a fluid server that is operating at the rate R. An example of C_i and D_i terms for a specific service discipline

[1] RSVP supports multiple senders and multiple receivers, as long as they are supported by the underlying internet.

can be found in Section 4.1. More generally, the C_i and D_i terms are best described in reference to a hypothetical system, $\mathcal{S}$, that first delays the arriving flow by $C_i/R + D_i$ units of time, and then serves the resulting flow at the reserved rate of R. The delay encountered in NE i, by any bit of the flow, is required to be no more than the total delay it would experience in the hypothetical system $\mathcal{S}$. The advantage of this type of specification is that it allows a simple computation of the end-to-end delay bound, given the *flowspec*, *adspec* and the reserved rate R. In addition, the buffer requirements at any NE can also be computed in a similar manner. These computations can be easily described in a graphical context, by using a "service curve" [6] to characterize the service provided to a flow by an NE, which is the topic of the next section.

2.2 Bounds on the Delay and Buffer Requirements

First, we use the notion of an *envelope* process to bound the amount of traffic that arrives at the network ingress, in any given time interval [4, 5]. Let $A[t, t+\tau]$ denote the amount of traffic that arrives in the interval $[t, t+\tau]$. We assume that an envelope function, $A(\tau)$, exists such that

$$A[t, t+\tau] \le A(\tau), \qquad t \ge 0, \tau \ge 0.$$

By definition, a (b, r, p) flow, has an envelope $A(\tau)$, that is given by

$$A(\tau) = \min\{L + p\tau,\ b + r\tau\}, \quad \tau \ge 0. \tag{1}$$

An NE is said to guarantee a service curve, $S(\cdot)$ to a flow, if for any $t \ge 0$, there exists an $s \le t$ such that there is no backlog of the flow at time s, and the service received by the flow in the interval $[s, t)$ is no less than $S(t-s)$ [6]. Specifically, the service curve at NE i for a flow that has reserved a rate of R can be obtained from the parameters exported by the NE i, and is given by,

$$\begin{aligned} S(\tau) &= \left[\left(\tau - \frac{C_i}{R} - D_i\right) R\right]^+ \\ &= \left[(\tau - D_i)R - C_i\right]^+, \end{aligned}$$

where $[x]^+ \equiv \max\{x, 0\}$. A consequence of the service curve formulation is that a service curve, $\overline{S}_i(\cdot)$, can be calculated for the tandem of NEs $1, 2, \ldots, i$, along the path of the flow, where [6],

$$\begin{aligned} \overline{S}_i(\tau) &= \min\left\{\sum_{j=1}^{i} S_j(\tau_j) : \tau_j \ge 0 \text{ and } \sum_{j=1}^{i} \tau_j = \tau\right\}, \\ &= \left[\left(\tau - \sum_{j=1}^{i} D_j\right) R - \sum_{j=1}^{i} C_j\right]^+. \end{aligned}$$

The service curve $\overline{S}_i(\cdot)$, can be used to compute an upper bound on the delay incurred by a packet of a (b, r, p) flow from the time it enters the network until it leaves NE i. In addition, $\overline{S}_i(\cdot)$ can also be used to calculate the buffer requirements at NE i. As shown in Figure 1, the delay bound and buffer requirements at NE i, are simply given by the horizontal and vertical distance between the traffic envelope (of the flow) and the service curve, respectively [6].

A closed form expression for an upper bound on the delay incurred by a packet, until and including the delay at NE

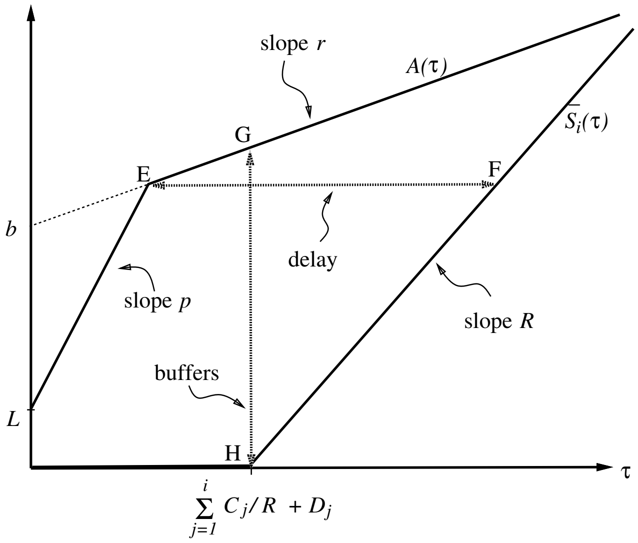

Figure 1: Delay and Buffer calculations for an (r, b, P) flow.

i, can be readily computed, by calculating the length of the segment EF in Figure 1, and is given by

$$\overline{D}_i = \begin{cases} \dfrac{(b-L)}{R}\dfrac{(p-R)}{(p-r)} + \dfrac{L}{R} + \displaystyle\sum_{j=1}^{i}\left[\frac{C_j}{R} + D_j\right] & \text{if } p > R, \\ \dfrac{L}{R} + \displaystyle\sum_{j=1}^{i}\left[\frac{C_j}{R} + D_j\right] & \text{if } p \le R. \end{cases} \tag{2}$$

In a similar manner, for the (b, r, p) flow, the buffer requirement at the NE i (GH in Figure 1), is given by

$$B_i = L + \frac{(p-X)}{(p-r)}(b-L) + \sum_{j=1}^{i}\left[\frac{C_j}{R} + D_j\right] X, \tag{3}$$

where,

$$X = \begin{cases} r & \text{if } \dfrac{(b-L)}{(p-r)} \le \displaystyle\sum_{j=1}^{i}\left[\frac{C_j}{R} + D_j\right] \\ R & \text{if } \dfrac{(b-L)}{(p-r)} > \displaystyle\sum_{j=1}^{i}\left[\frac{C_j}{R} + D_j\right] \ \&\ p > R \\ p & \text{otherwise.} \end{cases} \tag{4}$$

Equation (3) is only an upper bound on the buffer requirement for a given flow, and can be further improved. Notice, that the D term actually includes the latency that is due to the propagation delay on the link (see (16) for an example of the D term). Since this is a fixed latency, it does not contribute to the buffer requirement at the NE, which only depends on the variable portion of the delay bound. Therefore, if the fixed latency component from the first NE is known, it can be subtracted from the $\sum_{j=1}^{i} D_j$ term in (3), to yield a more accurate bound.

3 Reshaping traffic from a flow

In the previous section, we have described the characteristics of the flow at the network ingress. However, it is important to note that the characteristics of a flow can change as it makes its way through the network. In particular, if $A(\tau)$ is the envelope of the flow at the input to NE 1, the output associated with the same flow, need not have $A(\tau)$ as its envelope. This is because the scheduling of packets at the output link of NE 1 may introduce a clumping effect, that can result in an increase in the burstiness of the flow. When the flow traverses many NEs this perturbation can become quite significant. By reshaping the flow at an NE, we can ensure that the flow reaching the link scheduler has certain pre-specified characteristics, *viz.* (b, r, p) values or more generally a fixed envelope $A(\tau)$.

A traffic shaper smoothes a burst of packets that may arrive at its input by spacing them out, so that the flow at its output is bounded by a particular envelope. More formally, a shaper with envelope $A(\tau)$ outputs packets in order, with each packet being released at the smallest time, t, such that

$$A_{\text{out}}[t-\tau, t] \leq A(\tau), \qquad 0 \leq \tau \leq t, \tag{5}$$

where $A_{\text{out}}[t-\tau, t]$ denotes the traffic that is output from the shaper in the interval $[t-\tau, t]$.

In the rest of this section, we examine reshaping in terms of the benefits it affords to both the network and the traffic flows. Note that the requirement that a network element be able to reshape traffic already exists in some contexts, e.g., splitting and merging of flows as specified by the RSVP protocol [3]. As we illustrate in Sections 3.4 below, reshaping traffic within the network enhances the G-service by providing a mechanism for controlling the jitter across the network. Further, reshaping also reduces the buffer requirement at network elements (Section 3.3). These benefits have to be traded off against the costs of reshaping, which are an increase in mean delay and the burden of implementing per connection reshapers in routers. While the need for reshaping devices in a variety of environments, e.g., ATM networks and adapters [1], is likely to result in economies of scale that will mitigate the cost of reshaping, more work is needed to devise components that efficiently combine reshaping and scheduling.

3.1 Splitting of Flows

In the context of RSVP, it is possible for a receiver that is making a reservation to specify a *flowspec* that is different from that specified by another receiver for the same flow. Consider two receivers that are receiving data from the same sender, but one has made a reservation with a smaller *flowspec* than the other[2]. Now assume that the path from the sender to both the receivers share a common set of links after which it splits into two separate paths. The flow that is destined to the receiver that has specified the smaller *flowspec* has to be reshaped at this split point. In addition, whenever there is a shared reservation that is made for multiple flows, the combined traffic has to be reshaped[3] to ensure that the aggregate traffic satisfies the specified *flowspec*. Thus, reshaping within the network has to be supported.

[2] The ordering of the flowspecs is specified in [16]

[3] Alternatively, the traffic can be policed, which can be considered as a specific case of reshaping where packets are not buffered if they do not satisfy the shaper envelope

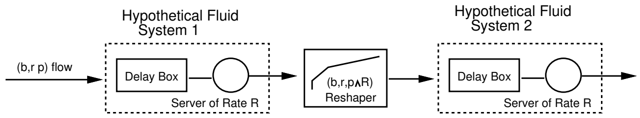

Figure 2: Hypothetical fluid systems and reshaper

3.2 Effect on the End-to-End Delay

Reshaping a G service flow delays individual packets, and can contribute to an increase in the average delay experienced by the packets of the flow. However, it is important to note that if the envelope of reshaping is chosen appropriately then the worst case delay suffered by packets of the flow does not increase, i.e., the service contract is not violated. To see why this is so, consider a (b, r, p) flow passing through two hypothetical fluid systems $\mathcal{S}_1$ and $\mathcal{S}_2$, where $\mathcal{S}_n$ delays the flow by $\frac{C_n}{R} + D_n$, $n = 1, 2$, and then serves it at the rate R (see Figure 2). It is intuitively clear (and easy to show) that the flow conforms to a $(b, r, p \wedge R)$ envelope when it departs $\mathcal{S}_1$, where $p \wedge R \equiv \min(p, R)$. Hence, if we insert a reshaper with envelope $A'(\tau)$, where

$$A'(\tau) = \min\{L + (p \wedge R)\tau,\ b + r\tau\}, \quad \tau \geq 0. \tag{6}$$

between $\mathcal{S}_1$ and $\mathcal{S}_2$, then this reshaper would have no effect on the flow, whatsoever.

Next, consider the same (b, r, p) flow traversing two network elements NE 1 and NE 2 in tandem, where NE n exports the parameters (C_n, D_n), $n = 1, 2.$. The upper bound on the end-to-end delay guaranteed to that flow with reservation R is given by $\overline{D}_2$ in (2), which is actually the delay experienced by the flow in the tandem of hypothetical fluid systems described above. Now, when we insert the $(b, r, p \wedge R)$ reshaper between NE 1 and NE 2, the delay experienced by some packets could increase. However, note that each packet arrives earlier at the reshaper after passing NE 1 than after passing $\mathcal{S}_1$. As the action of the reshaper is monotone, each packet then enters NE 2 earlier than it would enter $\mathcal{S}_2$. And finally, each packets departs NE 2 earlier than it would $\mathcal{S}_2$. Hence, the end-to-end delay of each packet is smaller under the configuration NE 1–Reshaper–NE 2, than under the configuration $\mathcal{S}_1$–Reshaper–$\mathcal{S}_2$. This shows that introducing the reshaper between NE 1 and NE 2 does not cause the end-to-end delay guarantee to be violated.

We may easily extend the above argument to the case of a (b, r, p) flow requesting a reservation R from each of a sequence of network elements $\{\text{NE } n : 1 \leq n \leq N\}$, where NE n exports the *adspec* (C_n, D_n). In general, one can show that introducing any number of reshapers which shape the flow to a $(b, r, \widehat{p})$ flow, with $\widehat{p} \geq p \wedge R$, does not alter the end-to-end delay bound presented in (2).

3.3 Reduced Buffer requirement

A casual examination of (3) reveals that the buffer requirements at the NEs keep increasing with each hop. Thus the buffer requirements for a flow that traverses many NEs can become quite large, particularly at the last hop. Since a reshaper in effect resets all the traffic characteristics of a flow, the buffer requirements for a flow are affected only by the NEs since the last reshaping point. More precisely, if $prev(i)$ denotes the index of the last reshaping point before NE i, the buffer requirement at NE i is given by,

$$B_i = L + \frac{(p-X)}{(p-r)}(b-L) + \sum_{j=prev(i)}^{i} \left[\frac{C_j}{R} + D_j\right] X, \quad (7)$$

where X is as defined in (4), but with the summations ranging from $j = prev(i)$ to i. Comparing (3) with (7), it is clear that reshaping can significantly lower buffer requirements for a flow that traverses multiple hops.

While reshaping does lower buffer requirements in the downstream NEs, it is important to point out that the reshaper itself, will require a certain amount of buffers. If the buffers are shared between the reshaper and the scheduler, then (7) can be used to compute the total buffers that are required for a given flow at an NE, at both the reshaper and the scheduler. Note that in this case, $prev(i)$ should be interpreted as the index of the previous reshaper, i.e. $prev(i) \neq i$.

If the shaper and scheduler have separate buffers, it is still fairly straightforward to compute the buffers required by a given flow at the shaper alone. Assume that the traffic shaper being considered is at NE i, and that it reshapes the flow to an envelope $A'(\tau)$. As before, we assume that the flow was previously reshaped at NE $prev(i)$, to an envelope $A(\tau)$.

The buffer requirement at the reshaper at NE i is computed as the vertical distance between $A(\tau)$ and a shifted version of $A'(\tau)$, just as in (7), and is given by[4]

$$B_{\mathrm{LB}} = \max_{\tau \geq 0} \left\{ A(\tau) - A'(\tau - \sum_{j=prev(i)}^{i-1} \frac{C_j}{R} - D_j) \right\}. \quad (8)$$

3.4 Reduction in Jitter

Another advantage of reshaping is the reduction in the jitter introduced into the flow as a result of scheduling. To illustrate this, consider a flow passing through a single NE. Assume that all the packets of the flow have the same size L, and that packet k, $k = 0, 1, 2, \ldots$ arrives at the NE at time a_k. If this packet departs the NE at time f_k we define the jitter, $\mathrm{Jitter}_{\mathrm{NE}}(k)$, experienced by packet k, $k = 1, 2, \ldots$ to be the absolute value of the difference between the inter-departure and inter-arrival times of that packet, i.e.,

$$\mathrm{Jitter}_{\mathrm{NE}}(k) := |f_k - f_{k-1} - (a_k - a_{k-1})|. \quad (9)$$

Note that the jitter of each packet is 0 if the stream is unperturbed, i.e., if the same delay is experienced by each packet. Averaging the jitter experienced by individual packets across the entire stream, we obtain a measure, $\mathrm{Jitter}_{\mathrm{NE}}$, of the effect of the NE on the flow, i.e.,

$$\mathrm{Jitter}_{\mathrm{NE}} := \limsup_{j \to \infty} \frac{1}{j} \sum_{k=1}^{j} \mathrm{Jitter}_{\mathrm{NE}}(k). \quad (10)$$

In the context of the G service, the question is how much jitter can the NE introduce into the flow, subject to the constraint that the NE does not delay any packet by more than D? To answer this we need to find a tight upper-bound on $\mathrm{Jitter}_{\mathrm{NE}}$, i.e., find $\max\{\mathrm{Jitter}_{\mathrm{NE}}\}$ over all possible departure sequences $\{f_0, f_1, \ldots\}$, subject to the condition $a_k + D \geq f_k \geq a_k$, and $f_{k+1} \geq f_k$, for all $k = 1, 2, \ldots$.

For the simple case of a constant rate stream with inter-packet spacing α ($a_k = \alpha \cdot k$), we can show that the above maximum is $(2\alpha\lfloor\frac{D}{\alpha}\rfloor)/(\lfloor\frac{D}{\alpha}\rfloor+1)$, where $\lfloor\cdot\rfloor$ denotes the floor function. Further, this maximum is actually attained for certain departure sequences. Now, consider the more general case where the flow passes through a sequence of N NEs. Let $\overline{D}_n$ be the maximum delay that any packet of the flow may experience in traversing the first n NEs. Then the maximum long term average jitter that could be introduced into the flow is $(2\alpha\lfloor\frac{\overline{D}_n}{\alpha}\rfloor)/(\lfloor\frac{\overline{D}_n}{\alpha}\rfloor + 1)$.

The effect of reshaping on the jitter may now be seen by placing a reshaper between the n-th and $(n+1)$-th NEs in the above sequence. The envelope of the reshaper $A(\tau)$ is assumed to be the same as that of the original stream, i.e., $A(\tau) = L + \frac{L}{\alpha}\tau$. Let $a_k = \alpha \cdot k$ be the time of arrival of packet k to the first NE, f_k the time at which it departs NE n and arrives at the reshaper[5], and g_k the time at which it departs from the reshaper. Now, the jitter, $\mathrm{Jitter}_{\mathrm{R}}(k)$, experienced by packet k after passing through the reshaper is

$$\mathrm{Jitter}_{\mathrm{R}}(k) := |g_k - g_{k-1} - (a_k - a_{k-1})|. \quad (11)$$

It is easy to see that once some packet experiences the delay of $\overline{D}_n$ in the first n NEs, i.e., for some packet j, $f_j = a_j + \overline{D}_n$, then all subsequent packets $j+1, j+2, \ldots$ experience no more jitter after passing through the reshaper, i.e., $\mathrm{Jitter}_{\mathrm{R}}(k) = 0$ for all $k = j+1, j+2, \ldots$. This follows from two observations. The first is that $g_k \leq a_k + \overline{D}_n$ for any $k = 1, 2, \ldots$, as the reshaper never increases the maximum end-to-end delay. From this observation, we deduce that $g_j = f_j$. The second observation is that the inter-packet spacing after the reshaper is at least α, i.e., $g_{k+1} \geq g_k + \alpha$, for $k = 1, 2, \ldots$. Combining the above, we have

$$a_{j+1} + \overline{D}_n \geq g_{j+1} \geq g_j + \alpha = a_j + \overline{D}_n + \alpha = a_{j+1} + \overline{D}_n. \quad (12)$$

From this we have $g_{j+1} = g_j + \alpha$, and hence $\mathrm{Jitter}_{\mathrm{R}}(j+1) = 0$. The chain of inequalities in (12) can be applied inductively to show that $g_{j+2} = g_{j+1} + \alpha$, and so on. Thus no subsequent packet experiences any jitter. In fact, extending this reasoning we may easily show that

$$\sum_{k=1}^{j} \mathrm{Jitter}_{\mathrm{R}}(k) \leq \overline{D}_n, \quad (13)$$

for any j. Thus, the reshaper smooths the flow back to an almost constant rate flow. This means that the long term average jitter past the reshaper is 0, and only the delay elements past the reshaping point contribute to the end-to-end jitter. In other words, instead of the end-to-end bound $(2\alpha\lfloor\frac{\overline{D}_N}{\alpha}\rfloor)/(\lfloor\frac{\overline{D}_N}{\alpha}\rfloor + 1)$, on the average end-to-end jitter without the reshaper, we have the smaller quantity $(2\alpha\lfloor\frac{\overline{D}_N - \overline{D}_n}{\alpha}\rfloor)/(\lfloor\frac{\overline{D}_N - \overline{D}_n}{\alpha}\rfloor + 1)$. Further, the size of the playback buffer required by the receiver is reduced to $L\lfloor\frac{\overline{D}_N - \overline{D}_n}{\alpha}\rfloor$, rather than $L\lfloor\frac{\overline{D}_N}{\alpha}\rfloor$ which would have been required without the reshaper.

Thus, reshaping plays a significant role in bounding the average jitter as well as the size of the playback buffer required to smooth constant rate flows. An important issue

[4] We have assumed that the C_i and D_i exported by NE i, are associated with the output and, therefore, have not included them in the buffer requirement in the shaper. If there is some *variable* internal latency in the NE i, before the packet gets to the reshaper, then it must be included in (8)

[5] Without loss of generality, we assume 0 propagation delays, as these do not contribute to jitter.

to note is that while it is sufficient to reshape at the last NE to reap the full benefits of jitter reduction, we significantly reduce the size of the reshaping buffer by doing so at each node.

4 RCS Discipline

The previous section outlined the advantages afforded by reshaping a flow at intermediate NEs. Motivated by these benefits we now consider reshaping the flow at each NE along its path. We refer to the combination of the shapers (for each flow) and the scheduler as the Rate Controlled Service (RCS) discipline. This is a generalized form of the RCS disciplines introduced in [22], a comprehensive study of which can be found in [9] as well. In general, any scheduler can be used in combination with the shapers, and [21] describes an implementation with a Static Priority scheduler. However, it is well established that the Non-preemptive Earliest Deadline First (EDF) policy is optimal in terms of guaranteeing deadlines [8], and so when we use the term RCS discipline, we implicitly assume that the EDF policy is being used. Our first task is to examine whether the RCS discipline can be used in the framework of [16], and if so, what the exported C and D parameters are in this case. Before we do that, let us quickly state the schedulability check for the EDF policy in a form that is easy to represent graphically. If there are M flows multiplexed onto an output link of speed γ, each with envelope $A_m(\tau)$, and with an associated deadline of d_m, $1 \le m \le M$, the following inequality has to be checked at the NE, to ensure that a feasible schedule (one that does not violate any deadlines) exists, [8],

$$\sum_{m=1}^{M} A_m(\tau - d_m) + \widehat{L} \le \gamma\tau, \quad \tau \ge 0, \tag{14}$$

where $\widehat{L}$ is the MTU of the link and for convenience, we have set $A_m(\tau) := 0$ for $\tau < 0$. Graphically, this amounts to checking whether the sum of the appropriately shifted traffic envelopes, lies at least $\widehat{L}$ units below the straight line passing through the origin with a slope of γ. An illustration of this check for the example considered in Section 4.3, can be found in Figure 3.

4.1 Parameters exported by the RCS discipline

We consider a (b, r, p) flow that has reserved a rate of R and focus on the output link of a single NE that uses an RCS discipline. The parameters exported by an NE that uses the RCS discipline, clearly depend on the choice of shaper envelope that is used to do the reshaping. In Section 3, it was shown that reshaping the traffic to an envelope of $(b, r, \widehat{p})$, where $\widehat{p} \ge p \wedge R$, does not impact the delay bounds given in (2). A flow with a smaller envelope, can in general, get better deadlines from the EDF scheduler, and so we choose the reshaper envelope as $(b, r, p \wedge R)$. The EDF scheduler, needs to have a deadline[6] associated with a packet in order to decide the order of packet transmissions. If a packet is released from the reshaper at time s_k, then we assign a deadline, f_k, given by

$$f_k = s_k + L/R + \widehat{L}/\gamma, \tag{15}$$

where γ is the speed and $\widehat{L}$ is the MTU of the outgoing link. Using (14) it can be checked (see [9] for details) that as long as the sum of the reserved rates (R) of all the flows are less than the link speed, γ, the above assignment of deadlines always results in a feasible schedule.

By definition of the EDF scheduler, it is clear that the maximum delay encountered by a packet from the time it is released from the reshaper, until it reaches the next hop, is no more than $L/R + L/\gamma + T_{\text{prop}}$ units of time, where T_{prop} is the propagation delay on the outgoing link. According to the definition of the exported parameters (see Section 2.1), it follows that the C and D terms exported by this NE are given by:

$$\begin{aligned} C &= L, \\ D &= \widehat{L}/\gamma + T_{\text{prop}}. \end{aligned} \tag{16}$$

In a heterogenous environment, only some of the NEs could be using the RCS discipline. These NEs export the above C and D values, which can then be substituted in (2) to obtain an upper bound on the end-to-end delay for any given flow.

Now, in addition, assume that a flow traverses NEs $1, 2, \ldots, N$, that are all using the RCS discipline. The end-to-end delay experienced by a packet from this flow is the total propagation delay as well as the sum of the delays encountered at both the shaper and the scheduler at each NE. In [9] it was shown that for the end-to-end delay bound, it is necessary to only include an upper bound on the shaper delay at the first shaper on the path. Thus the end-to-end delay bound $\overline{D}_N$ can be written as,

$$\overline{D}_N = \begin{cases} \dfrac{(b-L)}{R}\dfrac{(p-R)}{(p-r)} + \displaystyle\sum_{i=1}^{N}\left[\frac{L}{R} + \frac{\widehat{L}_i}{\gamma_i} + T_i\right] & \text{if } p > R, \\ \displaystyle\sum_{i=1}^{N}\left[\frac{L}{R} + \frac{\widehat{L}_i}{\gamma_i} + T_i\right] & \text{if } p \le R, \end{cases} \tag{17}$$

where T_i, $\widehat{L}_i$, and γ_i, are respectively, the propagation delay, MTU, and link speed at NE i. It can be readily verified that (17) results in a slightly tighter bound on the end-to-end delay than what is obtained by substituting (16) in (2).

4.2 Comparison with WFQ

Other scheduling disciplines that have often been suggested in the literature for providing per-flow delay (and throughput) guarantees, are those based on Weighted Fair Queueing (WFQ) [7, 14, 15]. In particular, the G Service specification lends itself readily to an implementation of WFQ schedulers at the output links of the NEs. With WFQ used as the scheduler, it follows (see [14, 15] for a justification) that the values for the C and D terms that have to be exported by the NE are the same as those in (16). Thus, (2) again provides a bound on the end-to-end delay experienced by the flow. There are, however, a couple of important caveats. One, is that the RCS discipline reshapes the traffic at the each NE to a smaller envelope than the original *flowspec*. While it is shown in Section 3.2 that this has no impact on the guaranteed end-to-end delay bound given by (2), it may clearly increase the average delay experienced by packets. On the other hand, the reshaping that takes place at every hop as a consequence of the RCS discipline, results in lower buffer requirements per flow at each NE. Additionally, the G service flows, can result in fairly low utilization of the link bandwidth. The RCS discipline, allows other flows to make use of this excess link bandwidth, albeit with larger delay guarantees. The WFQ discipline, by itself, cannot really allow other flows to make use of this unused bandwidth.

[6] Here deadline represents the time by which the last bit of the packet is guaranteed to have completed transmission

In the next example, we illustrate the above issues using a reasonable representative set of flows. Subsequently, we outline how with an RCS discipline the excess bandwidth can be utilized to support another type of service that we call a Guaranteed Rate service. Finally, we show how with a simple extension, the WFQ policy can also efficiently support the GR service.

4.3 Example

As an example, consider an OC-3 (155 Mb/s) output link at an NE. For simplicity, we assume that the G service traffic at this link is comprised of only the 3 types of flows that are listed in Table 1. For voice, we assume a standard 64Kb/s constant bit rate flow, while for Stored Video, we use typical values from MPEG traces of a movie like Star Wars, that roughly correspond to 3 Mb/s average rate and a burst size of around 100 Kbytes [13]. The Video Conference flow we consider, has an average rate of about 1.5 Mb/s and a maximum burst size of 10 Kbytes.. We assume that the peak rate for both types of Video flows is only limited by the speed of the media to which the source is attached (say 10-base T Ethernet), i.e., 10Mb/s. The maximum packet size for voice is limited to 100 byte packets, while for both the Video flows, we limit the packet size to 1500 bytes (Ethernet MTU). With each of the flows, there is an associated end-to-end delay requirement, that needs to be translated into a reservation rate R. In order to make this translation it is necessary to make some assumptions about the propagation delay and the number of hops traversed by each of the flows. For simplicity, we make identical assumptions for each of the flows, *viz.* we assume that each of the flows traverses 5 hops, with a total propagation delay of 20ms. We also assume that the MTU on all links traversed by the flows is 1500 bytes. The *flowspecs* for each of the flows are listed in Table 1.

Subtracting the propagation delay from the end-to-end delay requirement, for each of the flows, we obtain the allowable end-to-end queueing delays. Substituting the end-to-end queueing delay in (17), we can solve for the rate, R, that needs to be reserved for each flow. Table 2 lists the flows' delay requirements along with the corresponding rate, R, that needs to be reserved for each of them.

Traffic Type	L kB	b kB	r Mb/s	p Mb/s
64 Kb/s Voice	0.1	0.1	0.064	0.064
Video Conference	1.5	10	0.5	10
Stored Video	1.5	100	3	10

Table 1: Flow characteristics for 3 types of flows.

Traffic Type	e2e Delay (ms)	R (Mb/s)
64 Kb/s Voice	50	0.162
Video Conference	75	2.32
Stored Video	100	6.23

Table 2: End-to-end delay requirements and the rate reserved for each flow

If the entire 155 Mb/s of bandwidth could be used for G service traffic and the NEs were using the RCS discipline with EDF as the scheduler, then from (14) it can be verified that the following mix of traffic is feasible: 200 Voice flows, 26 Video Conference flows, 10 Stored Video flows.

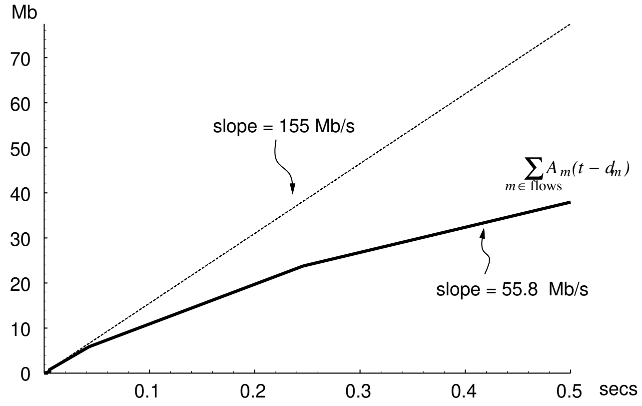

Figure 3: Schedulability check for EDF for traffic mix in Table 1.

Figure 3 graphically depicts the schedulability check of (14) at an NE for the traffic mix listed above. The straight line passing through the origin has a slope of 155 Mb/s, corresponding to the speed of the OC-3 link. The other curve is the sum of all the traffic envelopes, i.e.

$$\sum_{m \in \text{flows}} A_m(t - d_m),$$

where d_m is given by the $L/R + \widehat{L}/\gamma$ value corresponding to flow m. It is clear from Figure 3, that quite a bit of link bandwidth remains available for flows that do not have too stringent delay requirements. In particular, observe that the sum of the reserved rate for the above sample traffic mix is around 155 Mb/s which is the link capacity. However, if you consider the average throughput, it is only 55.8 Mb/s, resulting in a utilization that is about 1/3. This poor utilization, is not peculiar to the type of traffic mix that we have chosen. On the contrary, any flow that has a reserved rate, R, that is significantly larger than r, contributes to the low utilization of the link. For the flows that we have considered in our example the reserved rate R varies with the maximum packet size that is assumed for each flow. Table 3 lists the rate R that needs to be reserved for each flow, assuming different maximum packet sizes, with everything else remaining the same. However, note that we assume that there is no fragmentation, i.e., the MTU on all the links is large enough to accommodate the maximum packet size of all the flows. This is unlikely to be true in practice for large packet sizes, e.g., 50kB. In such cases, fragmentation would take place so that the maximum packet size to consider would then be the MTU on the path of the flow.

Going back to the example, it is possible to add a flow with a deadline of 111 ms, and a throughput of upto 99 Mb/s, without affecting the maximum delays seen by the other flows that have been considered so far. It is this available capacity that we feel should be utilized by a new service, that would provide the same rate guarantees as the G service, but with a much looser delay guarantee. We refer to this as the Guaranteed Rate service, and in the next section, describe it in some detail.

Max. Pkt Size L (kB)	Reservation Rate R (Mb/s)		
	64 Kb/s Voice	Video Conf.	Stored Video
0.1	0.16	1.40	5.91
0.5	0.81	1.66	6.00
1.0	1.62	1.99	6.11
1.5	2.43	2.32	6.23
5.0	8.35	4.87	7.07
10.0	17.50	9.15	8.36
25.0	50.96	24.71	16.31
50.0	140.37	57.01	35.77

Table 3: Variation of the Reserved Rate with the packet size

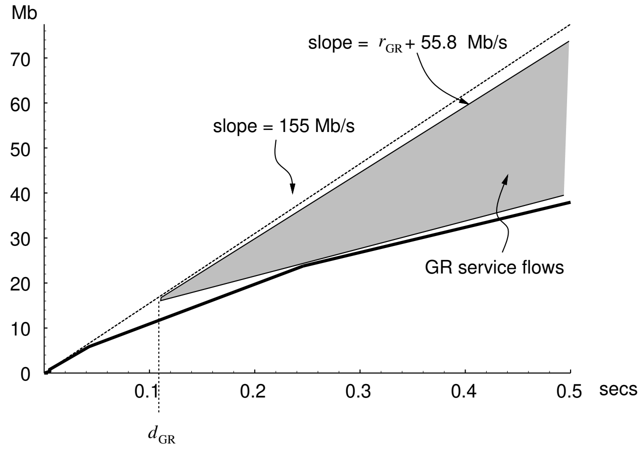

Figure 4: Schedulability check for EDF for traffic mix in Table 1 with the additional GR service flows.

5 Guaranteed Rate service

In this section, we describe a service that provides rate guarantees to flows, but unlike the G service, does not provide explicit delay guarantees. More precisely, a user making a guaranteed rate (GR) reservation for x Mb/s will be able to obtain a throughput of x Mb/s when measured over a fairly large period of time. This differs from a best effort type of service in that the user is *guaranteed* a certain amount of bandwidth that it will receive. No explicit guarantee is made as to the packet loss, but the buffers in the NEs should be engineered so that the packet loss can be made to be fairly small.

There are a number of applications that would benefit from a GR service. For example, an http session may be satisfied with such a kind of service guarantee, where the requested bandwidth depends on the size of the file that is being transmitted. For a large file, say a compressed movie clip, it would be preferable to have a much larger "bandwidth pipe" so that the entire file can be received in a few seconds. On the other hand it would be wasteful to have such a large pipe for a small file that contained only ascii text. With the G service, bandwidth and delay guarantees are coupled, and therefore it is not appropriate for applications that require only bandwidth guarantees. This is because such a coupling is typically more expensive in terms of resources than if only bandwidth guarantees are provided. In general, the GR service is suitable for most applications that like to have bandwidth guarantees, but for whom the G service would be an overkill.

At this point, it is worthwhile to contrast the GR service with the Controlled Load (CL) service proposed in [20]. The CL Service, requires the user to provide a *flowspec* that may be used for admission control, but once a flow is accepted, it expects to see an "unloaded" network [20]. In other words, each flow will experience fairly small delays (and losses) as long as it conforms to its *flowspec*. The GR service on the other hand does not promise small end-to-end delays, instead it only guarantees a certain throughput over some reasonable period of time. As a result, the delays experienced by a GR flow can temporarily be fairly large. This is because in order to use most of the bandwidth left available by the G service but without impacting the delay guarantees of the G service, it is necessary that the GR service be willing to tolerate occasional large delays. However, because of the worst case nature of the G service guarantees, instances where many G service packets are present and need to be sent out ahead of GR packets, should be relatively rare. Another difference with the CL service is that because the rate requested by GR flows is considered a hard guarantee, network elements are allowed to reshape GR flows to their TSpec. This TSpec is expected to be similar to that of the G service, with the peak rate and burst size used essentially as information to the network for the purpose of call admission and buffer sizing.

In the previous section we have demonstrated the availability of link bandwidth to support the GR service. However, it is not clear a priori whether this can be used to provide bandwidth guarantees over a reasonable period of time. In the next sections we investigate ways in which this service can be supported by both the RCS as well as the WFQ disciplines.

5.1 GR service with the RCS discipline

Consider Figure 4, that duplicates the schedulability check of Figure 3 for the G service flows that were considered in the previous section. In addition, Figure 4 depicts a potential GR flow that can be added to the mix of G flows, assuming it is given a local deadline (for the EDF scheduler) of $d_{\rm GR}$ and has a traffic envelope, $A_{\rm GR} := b_{\rm GR} + r_{\rm GR}\tau$[7]. This flow can be considered as representing the aggregate of all the GR service flows. It is clear from Figure 4 that as long as $r_{\rm GR} + 55.8 \leq 155$, we can find some $d_{\rm GR}$ so that no deadlines of the G service flows will be violated at the EDF scheduler. In general, the parameters $d_{\rm GR}$, $b_{\rm GR}$, and $r_{\rm GR}$ must be carefully determined to ensure that the GR service has a minimal impact on the G service flows that are carried by the NE.

The first step is to determine the local deadline, $d_{\rm GR}$, that can be assigned to the GR service flows, so that they have absolutely no impact on the delay guarantees for G service flows at this NE. Consider a set of G service flows numbered $1, \ldots, M$ that are multiplexed onto the output link that is being considered here. Let a flow m, have a *flowspec* of (b_m, r_m, p_m), and a reserved rate of R_m, $1 \leq m \leq M$. From (6), we know that the envelope for flow m at this NE is given by

$$A'_m(\tau) = \min\{L + \xi_m\tau,\ b + R\tau\}, \quad \tau \geq 0$$

[7]Note that for clarity, we do not assume any peak rate constraint for the GR flows

where $\xi_m := p_m \wedge R_m$. The deadline associated with this flow at the EDF scheduler is determined from (15) as $L/R_m + \widehat{L}/\gamma =: d_m$, where γ is the speed of the link. We assume that the deadlines for all the M flows are feasible at the EDF scheduler, which from (14) implies that the following constraint is satisfied:

$$\sum_{m=1}^{M} \min\left\{L + \xi_m(\tau - d_m)^+,\ b_m + r_m(\tau - d_m)^+\right\} + \widehat{L} \le \gamma\tau, \quad \tau \ge 0, \tag{18}$$

where $(x)^+ \equiv \max\{x, 0\}$.

Now, assume that a GR service flow with an envelope of A_{GR}, and a local deadline of d_{GR} at the output link, is multiplexed with the G service flows defined above. In order for a feasible schedule to exist, the following schedulability check has to be verified:

$$\sum_{m=1}^{M} \min\left\{L + \xi_m(\tau - d_m)^+,\ b_m + r_m(\tau - d_m)^+\right\} + b_{\mathrm{GR}} + r_{\mathrm{GR}}(\tau - d_{\mathrm{GR}})^+ + \widehat{L} \le \gamma\tau \quad \tau \ge 0, \tag{19}$$

Letting $\tau \to \infty$ in (19), we get the following condition on r_{GR}:

$$r_{\mathrm{GR}} \le \gamma - \sum_{m=1}^{M} r_m. \tag{20}$$

Given the total burst size, b_{GR}, of the aggregate GR service flows, in order for a feasible schedule to exist, the local deadline assigned to the GR service flows must satisfy,

$$d_{\mathrm{GR}} \ge \min_{t \ge 0}\left\{t : \sum_{m=1}^{M} A'_m(\tau - d_m) + A_{\mathrm{GR}}(\tau - t) + \widehat{L} \le \gamma\tau,\ \tau \ge 0\right\}$$

Assuming that $b_{\mathrm{GR}} = 100$ kBytes, and that $r_{\mathrm{GR}} = 99$ Mb/s, for the example considered in Section 4, we must have $d_{\mathrm{GR}} \ge 111$ msec. Figure 4 illustrates how the addition of this GR service flow with a deadline of around 111 msec to the schedulability check in Figure 3, still results in a feasible schedule.

From the previous discussion, it is clear that the values of d_{GR} and b_{GR} and r_{GR} are local decisions that have to be made at each NE, depending on the G service flows that it typically carries. In particular, for the example considered in Section 4.3, we computed the appropriate values for these parameters. So far, we have described the GR service and explained how it can be efficiently supported by the RCS discipline. The question remains, as to how this service, is to be supported by other service disciplines. In the next section, we demonstrate that with a minor extension, the WFQ discipline can also be made to support the GR service.

5.2 GR service with the WFQ service discipline

The GR service can also be offered if the WFQ scheduling discipline is used at the NE. One way in which it can be offered, is to have two priorities. One for the G service flows, and the other for the GR service flows, with the GR flows being served only if there are no packets from the G service, that are waiting to be served. Figure 5, gives a possible representation of the scheduler, with HIGH denoting the higher priority for the G service flows, and LOW denoting the lower priority queues for the GR service flows. When a packet from any of the G Service flows is present, the GR service flows (indicated by the shaded queues in Figure 5) are completely ignored in the calculation of the weights for the WFQ discipline. The GR flows are served only when there are absolutely no packets in the G service queues, at which time, the WFQ discipline only uses the weights that are in the non-empty GR service flows to schedule packet transmissions. For purposes of illustration, each flow is depicted as having a queue dedicated to it, but the buffers can be shared among the different flows. However, it is advisable to separate the buffers used for the GR and the G service, so as to ensure that the GR service has a minimal impact on the G service.

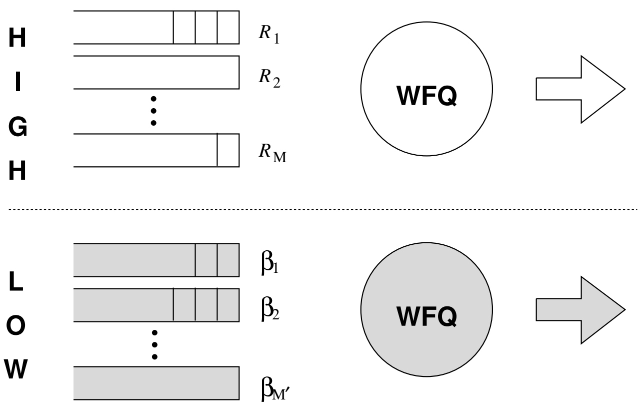

Figure 5: Prioritized, WFQ implementation of the GR service

The G service flows are depicted as having rates $R_1, R_2, \ldots R_M$, and the GR service flows are depicted as having rates $\beta_1, \beta_2, \ldots \beta_{M'}$. The schedulability check for the G service is simply,

$$\sum_{m=1}^{M} R_m \le \gamma. \tag{21}$$

For the GR service it is necessary to check that there is sufficient capacity for both the flows, *viz.*

$$\sum_{m=1}^{M} r_m + \sum_{m=1}^{M'} \beta_m \le \gamma. \tag{22}$$

Since the WFQ implementation for the GR service flows comes into play only when there are no G service packets, it is clear that there is almost no impact on the G service flows that are currently being carried by the NE. At most a G service packet has to endure an extra packet delay while the server is off serving the GR service flows. But this extra packet delay is already accounted for in the delay bounds that are known for WFQ and so there is really no difference in terms of the end-to-end delay guarantee [15, 11].

On the other hand, the worst case delay encountered by the GR service flows are affected by the G service traffic, and this can be analyzed using the service curve framework [6]. Clearly, the service curves for each of the GR service flows, strongly depends on the traffic characteristics of the G service flows. Let us assume that the G service flows can be characterized by the envelopes $A'_m(\tau)$, $1 \leq m \leq M$. Note that, these envelopes could in general, be different from the flow characteristics at the network ingress, since they have to account for the possible increase in burstiness caused by all the NEs on the path. If the flows are being reshaped at each hop, then they can be characterized by the envelopes of the reshapers themselves.

Given the traffic envelopes of the G service flows, we know that if the GR service queues are backlogged in the interval $[t, t+\tau]$, then they collectively receive a service of at least

$$S_{\mathrm{GR}}(\tau) = \left[\gamma\tau - \sum_{m=1}^{M} A'_m(\tau)\right]^+$$

Assuming that the aggregate GR service traffic is characterized by the envelope $A_{\mathrm{GR}}(\tau)$, then the delay encountered by any GR service packet is upper bounded by

$$d^*_{\mathrm{GR}} = \max_{t\geq 0} \min\left\{\tau : \tau \geq 0 \text{ and } A_{\mathrm{GR}}(t) \leq S_{\mathrm{GR}}(t+\tau)\right\}^+ . \tag{23}$$

Equation (23) is simply the horizontal distance between the traffic envelope, $A_{\mathrm{GR}}(\tau)$, and the service curve $S_{\mathrm{GR}}(\tau)$ (see Figure 1 for a graphical illustration of the delay computation). For the example considered in Section 4, we can readily compute the upper bound on the delay experienced by the GR flows to be, $d^*_{\mathrm{GR}} = 111$ msec, as before, and this is illustrated in Figure 6. The value of d^*_{GR} (or d_{GR} for the RCS discipline), gives an indication of the buffer requirements that are needed to ensure that packets from the GR service flows are not lost. Regardless of which scheduling discipline is used, i.e. RCS or WFQ, it is necessary to have sufficient buffers to ensure fairly low packet loss for the GR service flows. Since the value of d^*_{GR} (or d_{GR}) can be quite large, it may no longer be possible to engineer the buffer sizes based on a worst case analysis of the flows. A statistical model of the GR service traffic may be necessary in order to compute the buffer requirements, for some small packet loss probability and is the subject of ongoing work.

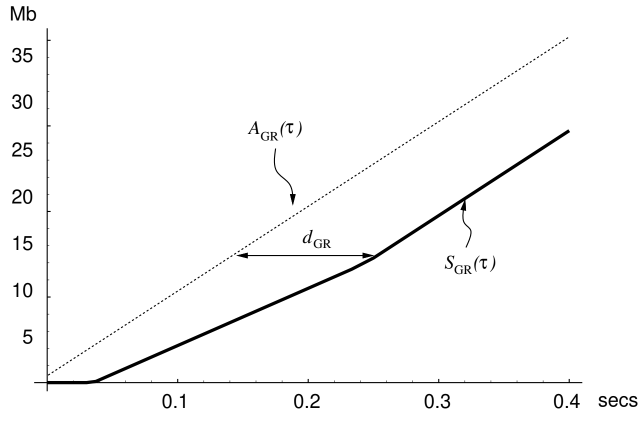

Figure 6: Illustration of the computation d^*_{GR}, for the WFQ discipline

6 Conclusions

In this paper, we have investigated the benefits of reshaping in the context of the G service. First, we established that in the context of the G service, the introduction of a reshaper does not increase the end-to-end delay bounds for a given flow. Subsequently, we established how reshapers can help lower buffer requirements at the NEs and also minimize the end to end jitter. We then leveraged this reshaping, to efficiently support the G service by using the family of RCS disciplines. Finally, we have shown how the bandwidth left unused by the G service can be exploited to provide a Guaranteed Rate service. This was done assuming either the RCS or the WFQ disciplines, although RCS disciplines were found to offer the benefit of an implementation synergetic with that of the G service.

An issue that is closely related to reshaping, is that of policing. In general, the decision of when to consider a packet non-conforming and what to do with it, is a complex one. This decision affects the size of the buffers needed to accommodate a flow, and can depend on the requirements and characteristics of the flow itself. For example, some applications may request a certain guaranteed rate, but consider it as a floor rate with the intention of sending more traffic if possible. Other applications may view this rate as the one necessary to provide a specific level of service to a flow, given that it conforms to certain traffic characteristics, i.e., *flowspec*. Investigating these different cases, and determining the range of services needed to support them effectively is the subject of ongoing work.

Acknowledgments

Many people have contributed to the material in this paper, either through discussions or exchanges of information. In particular, we are indebted to Scott Shenker and Craig Partridge for numerous conversations and email exchanges that have played a significant role in shaping the content of this paper. We also want to thank Dilip Kandlur and Victor Firoiu for many technical exchanges that have helped improve several aspects of the paper. We would like to acknowledge Fred Baker, whose views on the need for a Guaranteed Bandwidth service motivated some of the work in this paper. Finally, we would also like to acknowledge the many contributors to the Int-Serv mailing list (Int-Serv@isi.edu), whose comments often triggered investigations of some of the issues addressed in this paper.

References

[1] ATM Forum. ATM User-Network Interface Specification. Version 3.1, September 1994.

[2] F. Baker. Contribution to the Int-Serv mailing list, December 20, 1995.

[3] R. Braden, et. al. Resource ReSerVation Protocol (RSVP) - version 1, functional specification. Internet Draft, draft-ietf-rsvp-spec-12.ps, May 1996.

[4] C.-S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[5] R. L. Cruz. A calculus for network delay, Part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[6] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communication*, 13(6):1048–1056, August 1995.

[7] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking: Research and Experience*, 1:3–26, January 1990.

[8] L. Georgiadis, R. Guérin, and A. Parekh. Optimal multiplexing on a single link: Delay and buffer requirements. In *Proceedings of the IEEE INFOCOM '94*, 1994.

[9] L. Georgiadis, R. Guérin, V. Peris, and K. N. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. To appear in the August 1996 issue of the *IEEE/ACM Transactions on Networking.* Also available as Research Report RC 20064, IBM, T. J. Watson Research Center, May 1995.

[10] P. Goyal. Contribution to the Int-Serv mailing list, June 13 and July 15, 1995.

[11] P. Goyal and H. Vin. Generalized guaranteed rate scheduling algorithms: A framework. Technical Report TR-95-30, Department of Computer Sciences, University of Texas at Austin, 1995.

[12] S. Keshav. Contribution to the Int-Serv mailing list, July 13, 1995.

[13] P. Pancha and M. El Zarki. Leaky bucket access control for VBR MPEG video. In *Proceedings of the IEEE INFOCOM'95*, Boston, April 1995.

[14] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[15] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[16] S. Shenker and C. Partridge. Specification of guaranteed quality of service. Internet Draft draft-ietf-intserv-guaranteed-svc-03.txt, November 1995.

[17] S. Shenker, C. Partridge, B. Davie and L. Breslau. Specification of predictive quality of service. Internet Draft draft-ietf-intserv-predictive-svc-01.txt, November 1995.

[18] S. Shenker, C. Partridge, and J. Wroclawski. Specification of controlled delay quality of service. Internet Draft draft-ietf-intserv-control-del-svc-02.txt, November 1995.

[19] J. S. Turner. New directions in communications (or which way to the information age?). *IEEE Communications Magazine*, 24(10):8–15, October 1986.

[20] J. Wroclawski. Specification of the controlled-load network element service. Internet Draft draft-ietf-intserv-ctrl-load-svc-01.txt, November 1995.

[21] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10):1374–1396, October 1995.

[22] H. Zhang and D. Ferrari. Rate-controlled service disciplines. *Journal of High Speed Networks*, 3(4):389–412, 1994.