On the Effectiveness of Buffer in Deterministic Services *

Pawan Goyal and **Harrick M. Vin**

Pawan Goyal	**Harrick M. Vin**
goyal@research.att.com	vin@cs.utexas.edu
AT&T Labs - Research	Department of Computer Sciences
Networking and Distributed Systems Center	University of Texas at Austin
Florham Park, NJ 07932	Taylor Hall 2.124, Austin TX 78712

Abstract

We study the utility of buffer at switches in increasing the achievable utilization of a network providing deterministic guarantee. To determine the increase in utilization, we classify packet scheduling algorithms into two classes. Only one of these classes can utilize additional buffers to increase the achievable utilization. We experimentally determine the difference in achievable utilization of these classes. Our experiments demonstrate that contrary to intuition*, in most cases, additional buffers do not lead to higher achievable utilization of a network.*

1 Introduction

Integrated services networks support a diverse set of applications (e.g., data, audio, interactive video, stored video applications). The traffic characteristics as well as Quality of Service (QoS) requirements of these diverse applications vary significantly. To meet the QoS of requirements of the applications, a network has to manage two resources: link bandwidth and packet buffers. While several packet scheduling algorithms for managing link bandwidth have been developed and analyzed, the issues that arise in managing packet buffers

*A preliminary version of this paper appeared in the Proceedings of Network and Operating System Support for Digital Audio and Video, July 1998. This research was supported in part by IBM Graduate Fellowship, IBM Faculty Development Award, Intel, the National Science Foundation (Research Initiation Award CCR-9409666), NASA, Mitsubishi Electric Research Laboratories (MERL), and Sun Microsystems Inc.

for providing QoS guarantees have not been adequately investigated and are the subject of investigation of this paper.

To illustrate the issues that arise in managing buffers, consider a fluid flow model of the sources and the network. The switches in the network may either have no buffer or finite buffer. If the switches have no buffer, then fluid loss occurs whenever the aggregate arrival rate of flows at the output link of a switch exceeds the link capacity. In case of finite buffer, fluid is buffered when aggregate rate exceeds the capacity, and loss occurs only when the buffer overflows. Consider both types of networks when they provide deterministic guarantees on QoS parameters such as delay and loss. If the network does not have any buffers, then burstiness of a source may have to be smoothed at the edge of the network to ensure that loss does not occur. Such smoothing may introduce delay. Consequently, to meet the end-to-end delay requirements, resources greater than what is necessary may have to be reserved. On the other hand, if the network has buffers, then such smoothing and the consequent delay may not be necessary. This may enable the network to meet the delay requirement of a larger set of flows.

Thus, intuitively, it appears that use of buffer in a network should increase the achievable utilization of fluid flow networks. Hence, the question of interest is: Is the intuition correct for packet-switching networks? If yes, how much does the achievable utilization increase by adding buffer? Answers to these questions would determine the source rate control mechanism, which controls the smoothness of a flow, that should be employed in a network. If buffer in network does not yield higher utilization or the increase in utilization is not sufficient to offset the additional cost, a rate controller may completely smooth a flow to generate a constant bit rate (CBR) flow. If the buffer in the network yields increase in the achievable utilization, a source rate control mechanism may not smooth a flow at all or smooth it to appropriately tradeoff increase in utilization with increase in buffer requirement. Thus, the effectiveness of buffer in increasing achievable utilization would determine the appropriate buffering alternative, i.e., to buffer at source or in network.

In this paper, we take a step towards answering these questions. To determine if buffers at switches, henceforth referred to as servers, increase the achievable utilization of a network providing deterministic guarantees, we classify the packet scheduling algorithms as either belonging to the class of Guaranteed Rate (GR) scheduling algorithms or Rate Controlled Service Disciplines (RCSD). Whereas the scheduling algorithms in RCSD can utilize additional buffers to possibly increase the achievable utilization, the achievable utilization of GR scheduling algorithms is unaffected by the available buffer space. We experimentally determine the difference in achievable utilization of RCSD and GR servers when RCSD servers employ additional buffers. Our experiments demonstrate that *contrary to intuition*, in most cases, additional buffers in a network of RCSD servers do not lead to higher achievable utilization. This experimental result indicates that if a source desires deterministic bounds on packet delay, then the traffic should be completely smoothed at the source.

The rest of the paper is structured as follows. We formulate the problem under investigation and present the background for it in Section 2. We present our experimental methodology in Section 3. The experimental results are presented in Section 4 and then discussed in Section 5. Finally, Section 6 summarizes our results.

2 Background and Problem Formulation

The sequence of packets transmitted by a source is referred to as a flow and is used synonymously with a connection. A flow is serviced by a sequence of servers. Let there be K servers on the path of a flow and let the i^{th} server on the path be server i. For simplicity of exposition, let the propagation delay between the servers be 0 [1]. To provide deterministic guarantees on packet delay and loss, each server employs a packet scheduling algorithm. The achievable utilization of a server depends on the packet scheduling algorithm. Hence, to determine the effectiveness of buffer at servers in increasing achievable utilization, let the scheduling algorithms proposed in the literature be classified into two classes:

- Guaranteed Rate scheduling algorithms: The class of Guaranteed Rate (GR) scheduling algorithms is defined based on *expected arrival time* of a packet. Let p_f^j be the j^{th} packet of flow f. Then, expected arrival time of packet p_f^j at server i, denoted by $EAT^i(p_f^j)$, is defined as:

  $$EAT^i(p_f^j) = \max\left\{A^i(p_f^j), EAT^i(p_f^{j-1}) + \frac{l_f}{R_f}\right\} \tag{1}$$

  where $j \geq 1$, R_f is the rate reserved for flow f, l_f is the length of packets of flow f [2], $A^i(p_f^j)$ is the arrival time of p_f^j at server i, and $EAT^i(p_f^0) + \frac{l_f}{R_f}$ is defined to be 0.

  Scheduling algorithm at server i belongs to GR if it guarantees that packet p_f^j will be transmitted by $EAT^i(p_f^j) + \alpha_f^i$ where α_f^i depends on the scheduling algorithm as well as the server and the flow characteristics [6]. Several algorithms such as Virtual Clock [15, 18], Weighted Fair Queuing [1], Start-time Fair Queuing [8], Self Clocked Fair Queuing [4], Leap-forward Virtual Clock [14], Frame-based Fair Queuing [13], Delay EDD [7, 17] belong to the GR class. It has been shown in [6] that if

[1] All the subsequent analysis is applicable when the propagation delay is non-zero but bounded. However, use of non-zero propagation delay would clutter the presentation without providing any insight into the problem under investigation.

[2] For ease of exposition, we have assumed that the length of the packets of a flow does not vary.

each of the server on the path of a flow employs a scheduling algorithm in GR, then the end-to-end delay of packet p_f^j is given as:

$$d_f^j \leq EAT^1(p_f^j) - A(p_f^j) + \sum_{n=1}^{n=K} \alpha_f^n \tag{2}$$

where $A(p_f^j)$ is the time at which packet p_f^j is generated at the source.

- Rate Controlled Service Disciplines: The class of Rate Controlled Service Disciplines (RCSD) is defined based on the concept of shapers [3]. A shaper is a network element that ensures that its output traffic satisfies some burstiness constraints. If $AP^i(t_1, t_2)$ denotes the number of bits that are output from shaper S^i in time interval $[t_1, t_2]$, then shaper S^i delays the departure time of packets to ensure that $AP^i(t_1, t_2) \leq F^i(t_2 - t_1)$, where the function F^i characterizes the shaper S^i.

  Scheduling algorithm at server i belongs to the RCSD class with shaper S_f^i and delay d_f^i if it ensures that the departure time of a packet from the server is at most d_f^i more than its departure time from shaper S_f^i. RCSD class contains work conserving as well as non-work conserving algorithms (see [3, 16] for some examples). It has been shown in [3] that if each of the server on the path of a flow employs a scheduling algorithm in RCSD, then the network guarantees that the maximum end-to-end delay of any packet, denoted by d_f, is given as:

$$d_f = D_f^{\overline{S}} + \sum_{n=1}^{n=K} d_f^n \tag{3}$$

  where $D_f^{\overline{S}}$ is the maximum delay incurred by the source traffic at shaper $\overline{S}_f$, which is a composition of shapers $S_f^1, .., S_f^K$. Intuitively, $\overline{S}_f$ is a shaper composed by sequentially connecting shapers $S_f^1, .., S_f^K$ (see [3] for a precise definition of composition of shapers).

Observe that the bound on the departure time of a packet in GR at server i can be interpreted as being at most α_f^i more than the departure time in a shaper that delays a packet till its expected arrival time. Hence, if scheduling algorithm at server i belongs to GR, it also belongs to RCSD with $d_f^i = \alpha_f^i$ and shaper S_f^i that delays packet till its expected arrival time. However, since arbitrary shapers may be employed in RCSD, the converse is not true, i.e., a scheduling algorithm in RCSD may not belong to GR. Hence, GR is a proper subset of RCSD.

The main difference between GR and RCSD class is that if the burstiness of a source is completely smoothed at the edge of a network to reduce the buffer requirement in the network, then whereas the end-to-end delay does not increase in case of a network of GR servers, it may increase in case of a network

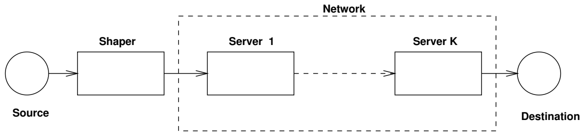

Figure 1: Network with Traffic Shaper

of RCSD servers. To observe this, let a shaper be employed at the edge of the network, i.e., between the source and the first server (see Figure 1). Let the departure time of packet p_f^j from such a shaper, denoted by $L^S(p_f^j)$, be equal to its expected arrival time, i.e., $L^S(p_f^j) = EAT^S(p_f^j)$. Such a shaper generates a smooth constant bit rate (CBR) flow. Now consider the end-to-end delay in such a system in a network of GR and RCSD servers.

- Network of GR servers: The maximum delay incurred by a packet consists of two components - the delay in the shaper and the delay in the network. The delay in the shaper of packet p_f^j is $L^S(p_f^j) - A(p_f^j) = EAT^S(p_f^j) - A(p_f^j)$. Using (2), we conclude that the maximum delay of packet p_f^j in the network is $\widehat{EAT}^1(p_f^j) - \widehat{A}^1(p_f^j) + \sum_{n=1}^{n=K} \alpha_f^n$ where $\widehat{EAT}^1(p_f^j)$ and $\widehat{A}^1(p_f^j)$ are the expected arrival time and actual arrival time of p_f^j at server 1 when the shaper is employed. Hence, we conclude:

$$\begin{aligned} d_f^j \quad \leq \quad & EAT^S(p_f^j) - A(p_f^j) + \widehat{EAT}^1(p_f^j) \\ & -\widehat{A}^1(p_f^j) + \sum_{n=1}^{n=K} \alpha_f^n \end{aligned} \tag{4}$$

  Since propagation delay has been assumed to be 0, the arrival time of a packet at the shaper is the same as the arrival time at the first server in a network without the shaper. Hence, $EAT^S(p_f^j) = EAT^1(p_f^j)$. Furthermore, due to the use of shaper, $\widehat{EAT}^1(p_f^j) = \widehat{A}^1(p_f^j)$. Hence, we get:

$$d_f^j \leq EAT^1(p_f^j) - A(p_f^j) + \sum_{n=1}^{n=K} \alpha_f^n \tag{5}$$

  Since (5) is same as (2), we conclude that the end-to-end delay does not change due to the presence of the shaper.

- Network of RCSD servers: Let the maximum delay incurred by packets in the shaper at the edge be denoted by D_f^C. Also, let the maximum delay incurred by the packets at shaper $\overline{S}_f$ after being shaped at the edge be $\widehat{D}_f^{\overline{S}}$. Hence, maximum end-to-end delay is given as:

$$\widehat{d}_f = D_f^C + \widehat{D}_f^{\overline{S}} + \sum_{n=1}^{n=K} d_f^n \tag{6}$$

Now, $D_f^C + \widehat{D}_f^{\overline{S}}$ may be greater than the delay $D_f^{\overline{S}}$ experienced by packets at shaper $\overline{S}_f$ when traffic is not shaped at the edge. To observe this, let $\overline{S}_f$ be a (kl_f, r_f) leaky bucket shaper and flow f conform to (kl_f, r_f) leaky bucket. A shaper is a (σ, r) leaky bucket shaper, where σ is the burstiness and r the average rate, if it ensures that for all intervals $[t_1, t_2]$:

$$AP(t_1, t_2) \leq \sigma + r(t_2 - t_1) \tag{7}$$

A flow conforms to (σ, r) leaky bucket if it does not incur any delay in a (σ, r) leaky bucket shaper. Then, since flow f conforms to the shape enforced by the shaper $\overline{S}_f$, $D_f^{\overline{S}} = 0$. However, since D_f^C is given as $\frac{(k-1)l_f}{r_f}$ [6], if $k > 1$, $D_f^C + \widehat{D}_f^{\overline{S}} > 0$. Hence, $\widehat{d}_f$ may be greater than d_f. Thus in case of RCSD, the end-to-end delay may increase due to the presence of the shaper.

Hence, if buffer requirement in the network is reduced to that of a CBR flow by employing a shaper at the network edge then:

- In GR networks the delay does not increase. Hence, increasing the available buffer space does not effect the achievable utilization.
- In RCSD networks the delay may increase. Hence, to ensure that end-to-end delay requirements of the flow are met, higher resources may have to be reserved which may result into loss in achievable utilization.

Hence, we conclude that the available buffer space does not effect the achievable utilization of GR networks but may effect that of RCSD networks. Therefore, the effectiveness of buffer in increasing the utilization of a network can be determined by evaluating the difference in achievable utilization of a network of RCSD and GR servers when RCSD servers employ additional buffers. Since RCSD requires additional buffers only when the shaper $\overline{S}_f$ is a non-CBR shaper, henceforth we will assume that non-CBR shapers are employed in RCSD.

To gain an intuitive understanding for the difference in achievable utilization of a network of GR and RCSD servers, let us first consider a single server scenario. Let the flows conform to leaky buckets, i.e., flow f conforms to a (σ_f, r_f) leaky bucket. Furthermore, for scheduling algorithms in GR, let $\alpha_f^i = \frac{l_f}{R_f} + \frac{l_{max}^i}{C^i}$ where C^i is the capacity of server i and l_{max}^i is the maximum length of packets served by server i (scheduling algorithms such as WFQ and Virtual Clock satisfy this assumption [6]). Since EDF has the largest schedulability region, let the RCSD scheduling algorithms employ EDF scheduler and a (σ_f, r_f) leaky bucket shaper for flow f.

Consider two flows f and m that conform to leaky bucket with parameters $(10pkt, 1pkt/s)$ and $(1pkt, 1pkt/s)$, respectively. Let both flows f and m require delay of $5.5s$ and be served by a single server with capacity $2pkts/s$. Then, from the results in [3], it can be shown that a RCSD server satisfies the deadline requirements of both the flows. On the other hand, in case of GR, from (2) and [6] we know that flow f would have to reserve at least rate R_f such that:

$$\frac{\sigma_f - l_f}{R_f} + \frac{l_f}{R_f} + \frac{l_{max}}{C} = 5.5 \tag{8}$$

Hence, $\frac{10}{R_f} + 0.5 = 5.5$ which yields $R_f = 2pkt/s$, which is the capacity of the server. Hence, the server would not be able to accept flow m. Thus, the achievable utilization is higher in case of RCSD.

Though achievable utilization is higher in case of a single server for RCSD, it is not clear whether the same result holds in a network of servers. To observe this , let flow f have an end-to-end delay requirement of d_f. Since we have assumed that the shaper employed is the same as the input traffic, $D_f^{\overline{S}} = 0$. Hence, from (3) we get:

$$d_f = \sum_{n=1}^{n=K} d_f^n \tag{9}$$

If we assume each of the server is equally loaded, then the delay of the flow at each server is $\frac{d_f}{K}$. Thus, *intuitively*, for a given end-to-end delay, the resource requirement at each server for a flow increases linearly with the number of servers on the path. On the other hand, in a network of GR servers, we know from (2) and [6] that the reserved rate $R_f \geq r_f$ should be such that:

$$d_f \geq \frac{\sigma_f - l_f}{R_f} + K\frac{l_f}{R_f} + \sum_{i=1}^{i=K} \frac{l_{max}^i}{C^i} \tag{10}$$

Then, if σ_f is large, i.e., when RCSD servers may be expected to have larger utilization, R_f increases sub-linearly with the number of servers on the path. Hence, intuitively, whereas resource requirement increases linearly with the number of servers in a RCSD network, it increases sub-linearly in a GR network. Consequently, RCSD networks will have higher utilization only if the reduction in resource requirement yielded by additional buffer is sufficient to offset the difference in the linear and sub-linear increase. Otherwise, RCSD networks may not have higher utilization and, contrary to intuition, may have lower utilization.

A theoretical analysis to determine the difference in achievable utilization of GR and RCSD networks has remained elusive. Hence, to determine the difference we conduct a large number of experiments. In the next section, we describe our experimental methodology and present the experimental results in Section 4.

3 Experimental Methodology

To experimentally determine the difference in achievable utilization of a network of GR and RCSD servers, we assume that all the flows conform to leaky bucket. (Since the traffic model used by the Internet and ITU standard bodies is leaky bucket, this is not a restrictive assumption.). Hence, flow f is characterized by the quintuple $(\sigma_f, r_f, l_f, d_f, K_f)$ where σ_f,r_f,l_f,d_f, and K_f denote the burstiness, rate, packet length, end-to-end delay requirement, and the number of servers on the path of flow f, respectively. We make the following assumptions for GR and RCSD networks:

- GR: We assume that for GR servers $\alpha_f^i = \frac{l_f}{R_f} + \frac{l_{max}^i}{C^i}$ where C^i is the capacity of server i and l_{max}^i is the maximum length of packets served by server i (we remove this restriction in Section 4.2). Hence, rate $R_f \geq r_f$ that would satisfy the end-to-end delay requirements of flow f can be determined by (10). We assume that R_f is reserved for flow f at each of the servers.

- RCSD: We assume that RCSD servers employ EDF scheduler and a (σ_f, r_f) leaky bucket shaper for flow f. Furthermore, we assume each of the server is equally loaded and hence the delay of flow f at each server is $\hat{d}_f = \frac{d_f}{K_f}$.

We assume that the capacity of each server is C. The schedulability conditions that we employ for GR and RCSD are as follows:

- GR: We ensure that aggregate rate reserved by the flows at a server is less than the server capacity. Hence, if Q flows are served by a server, then the schedulability condition is $\sum_{i=1}^{i=Q} R_i \leq C$.

- RCSD: We use the schedulability condition presented in [2], which is a generalization of Theorem 1 of [3] for networks that have variable packet sizes, to determine the schedulability region. Specifically, flows $[1..Q]$ are schedulable if $\sum_{i=1}^{i=Q} r_i \leq C$ and the following set of inequalities hold:

$$\sum_{i=1}^{i=k} l_i + \sum_{i=1}^{i=k} (\sigma_i - l_i) + l_{max} \leq \hat{d}_k \left(C - \sum_{i=1}^{i=k-1} r_i \right) + \sum_{i=1}^{i=k-1} r_i \hat{d}_i \quad 1 \leq k \leq Q \tag{11}$$

Without loss of generality we have assumed that $\hat{d}_1 \leq .... \leq \hat{d}_Q$. If all the flows have the same packet length l, then we use Theorem 1 of [3] and ensure that the following set of inequalities hold:

$$\min\{(k+1)l, Ql\} + \sum_{i=1}^{i=k} (\sigma_i - l) \leq \hat{d}_k \left(C - \sum_{i=1}^{i=k-1} r_i \right) + \sum_{i=1}^{i=k-1} r_i \hat{d}_i \quad 1 \leq k \leq Q \tag{12}$$

To experimentally evaluate the difference in achievable utilization between RCSD and GR networks, we first assume that there are only two types of flow at the servers: low throughput and high throughput flows (experimental methodology for heterogeneous flows is presented in Section 4.3). Let $r_l < r_h$ and the quintuples $(\sigma_l, r_l, l_l, d_l, K_l)$ and $(\sigma_h, r_h, l_h, d_h, K_h)$ characterize the low and high throughput flows, respectively. We employ the schedulability conditions presented above to compute the schedulability region for RCSD and GR servers. The schedulability region is given by a set of tuples (n_h, n_l) such that n_h and n_l are the maximum number of high and low throughput flows, respectively, that can be admitted simultaneously (Figure 2 shows a hypothetical schedulability region for RCSD and GR servers). To compare schedulability regions of GR and RCSD and hence determine the difference in achievable utilization between GR and RCSD, we define two metrics: *average utilization gain* and *maximum utilization gain*. To define these metrics, we first define the functions $U^r(i)$ and $U^g(i)$ that compute the achievable utilization of a RCSD and GR server, respectively, when i high throughput flows are admitted.

Let $L^r(i)$ and $L^g(i)$ be the maximum number of low throughput flows that are admitted in RCSD and GR when i high throughput flows are admitted, respectively. Also, let n_h^r and n_h^g be the maximum number of high throughput flows that can be admitted by RCSD and GR server, respectively. We need to define $U^r(i)$ and $U^g(i)$ for $i \in [1..N]$ where $N = \max\{n_h^g, n_h^r\}$. Consider the definition of $U^r(i)$. Clearly, when $i \le n_h^r$, $U^r(i)$ is:

$$U^r(i) = \frac{r_h i + r_l L^r(i)}{C} \quad i \le n_h^r \tag{13}$$

If $n_h^r \le n_h^g$, then we will require $U^r(i)$ for $i > n_h^r$ also. In such a case, the number of high throughput flows that should be used in determining utilization is n_h^r. Furthermore, since $L^r(i)$ is not defined for $i > n_h^r$, $L^g(i)$ should be chosen as the number of low throughput flows. However, since at most $L^r(n_h^r)$ low throughput flows may be admissible along with n_h^r high throughput flows, we conclude that $\min\{L^r(n_h^r), L^g(i)\}$ number of low throughput flows should be used in the computation of $U^r(i)$ when $i > n_h^r$. Hence, we get:

$$U^r(i) = \frac{r_h n_h^r + r_l \min\{L^r(n_h^r), L^g(i)\}}{C} \quad n_h^r < i \le N \tag{14}$$

The function $U^g(i)$ is defined analogously. Note that in the definition of $U^g(i)$, the actual rates of the flows, and not the reserved rates, are used.

We now define the two metrics as follows:

- *Average utilization gain*: This metric, denoted by U, is defined to capture the expected increase in utilization yielded by a RCSD server. It is defined as:

$$U = \frac{1}{N}\left(\sum_{i=1}^{i=N} \frac{U^r(i) - U^g(i)}{U^g(i)}\right) \tag{15}$$

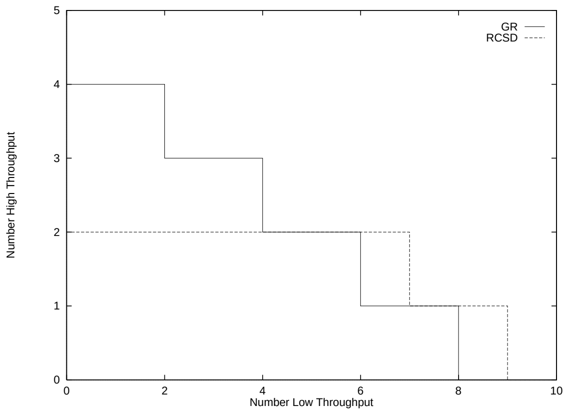

Figure 2: Example Schedulability Regions for GR and RCSD

- *Maximum average utilization gain*: This metric, denoted by U^+ is defined to capture the maximum expected increase in utilization yielded by a RCSD server. It is based on the observation that since GR class is a proper subset of RCSD class, it is always possible for a RCSD server to achieve the utilization of a GR server. Hence, for determining maximum expected increases in utilization, any loss in utilization yielded by RCSD may be ignored. Specifically, it is defined as:

$$U^+ = \frac{1}{N}\left(\sum_{i=1}^{i=N} \frac{\max\{0, (U^r(i) - U^g(i))\}}{U^g(i)}\right) \tag{16}$$

We are now ready to define our experimental methodology. In an experiment, we choose a set of low and high throughput flows. For each pair of low and high throughput flows chosen from their respective sets, we determine the schedulability region in RCSD and GR. For each pair, we compute U and U^+. Thus, we get a set of values for the two metrics. We then determine a cumulative distribution function for the metrics and employ it to determine the difference in achievable utilization between GR and RCSD servers. In the next section, we present and analyze the results of some of the experiments that we have conducted.

4 Experimental Evaluation

4.1 Basic Experiments

We construct several sets of low and high throughput flows and choose several of their combinations. In a particular set, all the flows have equal number of servers on their path. Hence, let $S_l^{i,k}$ and $S_h^{i,k}$ denote the i^{th} low and high throughput set, respectively, with all flows having k servers on their path. Let l_l and l_h be the length of low and high throughput flow packets, respectively. Also, let Σ_l, Φ_l, and D_l^i be set of

Type	Burstiness Σ	Rate Φ (Kb/s)	Delay D^1 (ms)	D^2 (ms)	D^3 (ms)
Low	$\{l_l, 10l_l, 20l_l\}$	$\{32, 64, 128\}$	$\{25, 50, 75, 100, 200\}$	$\{25, 50, 75, 100\}$	$\{100, 125, 150, 175, 200\}$
High	$\{l_h, 10l_h, 20l_h\}$	$\{640, 1280, 2560\}$	$\{25, 50, 75, 100, 200\}$	$\{100, 125, 150, 175, 200\}$	$\{25, 50, 75, 100\}$

Table 1: Parameters for the experiments

burstiness, rate, and delay values for the low throughput flows. Then, the set $S_l^{i,k}$, $i \in [1..3]$, is defined as follows:

$$S_l^{i,k} = \{(\sigma, r, d, l_l, k)| \;\; \sigma \in \Sigma_l \wedge r \in \Phi_l \wedge d \in D_l^i\} \tag{17}$$

The set $S_h^{i,k}$ is defined analogous to $S_l^{i,k}$. Observe that flows in set $S_l^{1,k}..S_l^{3,k}$ differ only in their delay values. The sets Σ, Φ, and D^i for low and high throughput flows are defined in Table 1. (The delay and throughput values for the low and high throughput flows are chosen so that they correspond to audio and video flows, respectively). We let the number of servers on the path be either $1, 3, 5, 10$, or 15. Note that since the average number of hops in Internet is 17, the number of servers we choose is conservative [10]. We conduct three experiments:

- EXP1: $S_l^{1,k}$ and $S_h^{1,k}$ are chosen to be the low and high throughput sets. In this case, some flows in the low throughput set have higher delay requirement than the flows in the high throughput set, and vice versa.
- EXP2: $S_l^{2,k}$ and $S_h^{2,k}$ are chosen to be the low and high throughput sets. In this case, flows in the low throughput set always have lower delay requirement than the flows in the high throughput set.
- EXP3: $S_l^{3,k}$ and $S_h^{3,k}$ are chosen to be the low and high throughput sets. In this case, flows in the high throughput set always have lower delay requirement than the flows in the low throughput set.

We assume that the capacity of each server is $50Mb/s$ and conduct these experiments for network environments with fixed as well as variable size packets (leading to the computation of 52650 schedulability regions).

- Networks with fixed packet size: We choose $l_l = l_h = 512$ bits. Figures 3 and 4 plot the cumulative distribution function for the metrics U and U^+, respectively, for different values of k. Figure 3 shows that $70 - 80\%$ of the time, average gain is less than 0, and it is at most 0.15 in all the experiments when $k > 1$. Figure 4 shows that maximum average gain also is at most 0.15.

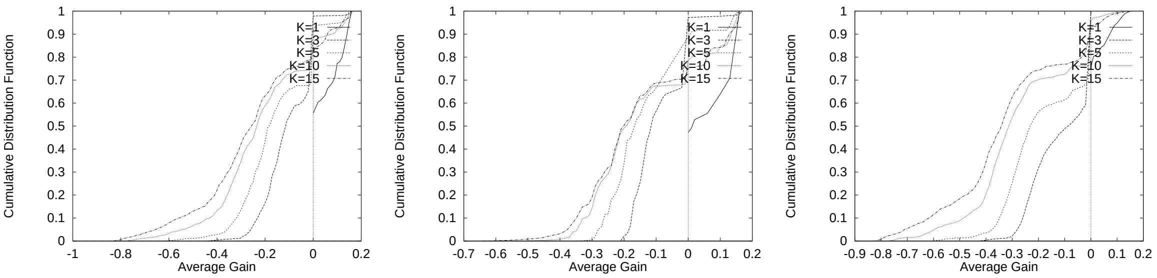

Figure 3: Fixed size packet networks: Average gain in (a) EXP1, (b) EXP2, and (c) EXP3

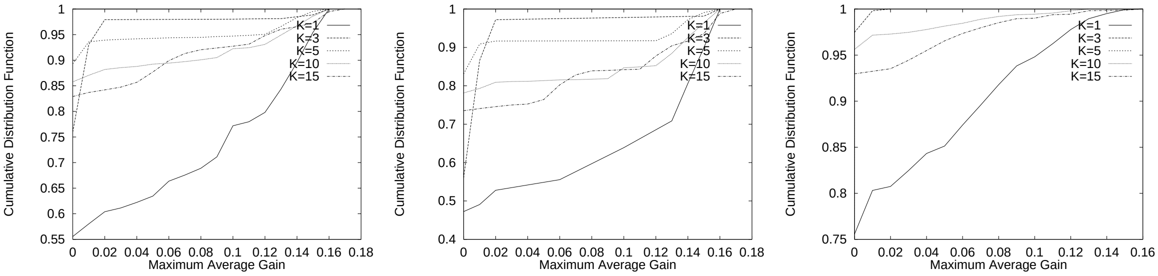

Figure 4: Fixed size packet networks: Maximum Average gain in (a) EXP1, (b) EXP2, and (c) EXP3

- Networks with variable packet size: In networks with variable packet size, we expect low throughput applications (such as audio) to have small packet size and high throughput applications such as video to have large packet sizes. Hence, we choose $l_l = 512$ bits and $l_h = 8000$ bits. Figures 5 and 6 plot the cumulative distribution function for the metrics U and U^+, respectively, for different values of k. Figure 5 shows that $85 - 90\%$ of the time average gain is less than 0 in all the experiments when $k > 1$. Figure 6 shows that 90% of the time the maximum average gain is less than 0.02 in all the experiments when $k > 1$. It also shows that sometimes, the maximum average gain is as high as 1.

From the above experiments, we conclude, for both type of networks, that: (1) as per the intuition presented in Section 2, the average gain and maximum average gain decrease with increases in the number of servers on the path; (2) $70 - 80\%$ of the time, the average gain is less than 0; (3) $85 - 90\%$ of the time, the maximum average gain is less than $0.02 - 0.03$; and (4) $10 - 20\%$ of the time, the maximum average gain is non-negligible.

A further examination of the experimental results for the $10 - 20\%$ cases, in which the maximum average gain was non-negligible, demonstrated that most of the significant increase in utilization of RCSD occurred when both the low and high throughput flows have burstiness equal to the packet length, i.e., the flows are CBR. Since theoretically the achievable utilization of GR and RCSD should be same when all the flows are CBR, this seems to be a contradictory result. However, this is only an apparent contradiction.

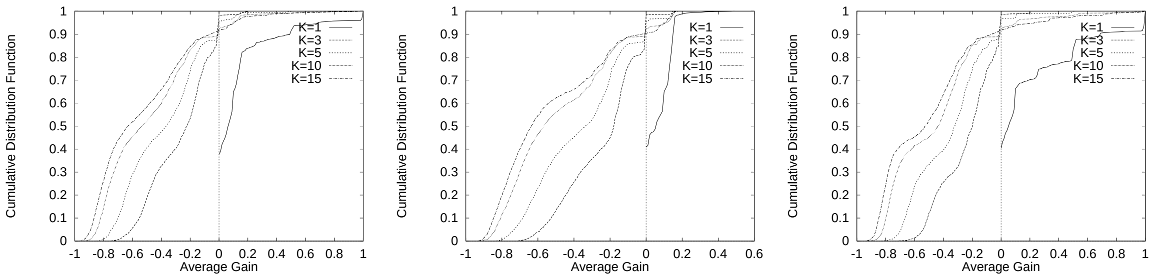

Figure 5: Variable size packet networks: Average gain in (a) EXP1, (b) EXP2, and (c) EXP3

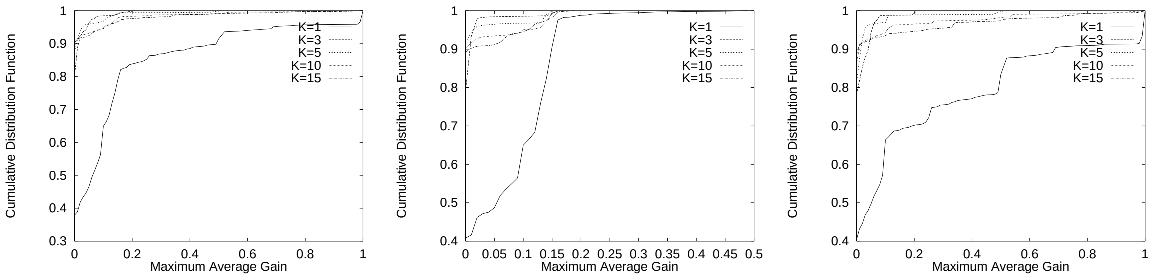

Figure 6: Variable size packet networks: Maximum Average gain in (a) EXP1, (b) EXP2, and (c) EXP3

To observe this, recall that we had restricted the GR class to algorithms for which α^i was given as $\alpha^i_f = \frac{l_f}{R_f} + \frac{l^i_{max}}{C^i}$. This restriction included algorithms that allocate only rate (e.g., WFQ, Virtual Clock). However, GR class contains algorithms such as Delay EDD that separate rate and delay allocation and can have $\alpha^i_f < \frac{l_f}{R_f} + \frac{l^i_{max}}{C^i}$, or $\alpha^i_f > \frac{l_f}{R_f} + \frac{l^i_{max}}{C^i}$. Such algorithms have higher achievable utilization than the algorithms that allocate only rate. The unrestricted GR class that contains Delay EDD has the same achievable utilization as the RCSD class when the flows are CBR. Thus, to determine whether the non negligible increase in utilization of RCSD server is due to the additional buffer employed by it or the restriction that we had placed on GR, in the next section, we repeat the experiments with the unrestricted GR class.

4.2 Experiments with Unrestricted GR Class

In the unrestricted GR class, the reserved rate $R_f \geq r_f$ that will meet the end-to-end delay requirement of flow f can be derived using:

$$d_f = \frac{\sigma_f - l_f}{R_f} + K\hat{d}_f \tag{18}$$

where we have assumed that $\alpha^i_f = \hat{d}_f$. Observe that there are several combinations of R_f and $\hat{d}_f$ that will satisfy (18). Hence, to determine both R_f and $\hat{d}_f$, we need one more constraint.

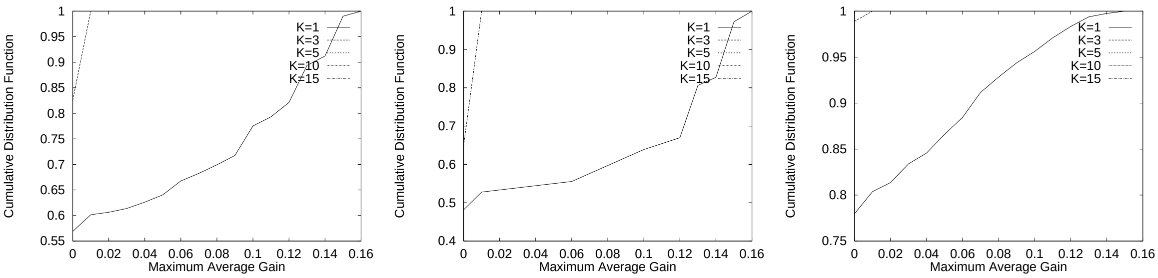

Figure 7: Fixed size packet networks: Maximum Average gain with Delay EDD in (a) EXP1, (b) EXP2, and (c) EXP3

To develop this constraint, we will assume that the scheduling algorithm at each of the servers is Delay EDD. The schedulability conditions for Delay EDD for variable size and fixed size packet networks are given by (11) and (12), respectively, with l_f and R_f substituted for σ_f and r_f, respectively. From an examination of the schedulability conditions, we conclude that neither very high value of R_f nor a very small value of $\hat{d}_f$ are desirable. Hence, to balance these two tradeoffs we choose minimum value of $R_f \geq r_f$ that will satisfy (18) while ensuring that:

$$\hat{d}_f \geq \epsilon \frac{l_f}{R_f} \tag{19}$$

where $\epsilon \leq 1$.

We determine the schedulability region for Delay EDD as follows. Since the set of flows that are schedulable by rate only allocation algorithms are also schedulable by Delay EDD, we first determine a tuple (n_h, n_l) as in Section 3 for GR. We then choose $\epsilon = 0.1, 0.3, 0.5, 0.7$, and 0.9 and determine the maximum number of low throughput flows, $\hat{n}_l \geq n_l$, that can be accepted along with n_h high throughput flows. This procedure when repeated for all values of n_h gives the schedulability region for Delay EDD.

We repeat the experiments presented in Section 4.1 to compare the achievable utilization in GR and RCSD. We find that the average gain becomes less than 0 in all the experiments for $k > 1$. Hence, we only plot the maximum average gain for fixed and variable length packet networks in Figures 7 and 8, respectively. As the figures demonstrate, the maximum average gain reduces significantly for $k > 1$. It is always less than 0.01 for fixed size packet networks and 0.05 for variable size packet networks.

We repeat all the experiments that we have presented so far by increasing all the delay values by 100ms. Figures 9 and 10 plot the maximum average gain for fixed size and variable size packet networks, respectively. As the figures demonstrate, there is no significant change in the results.

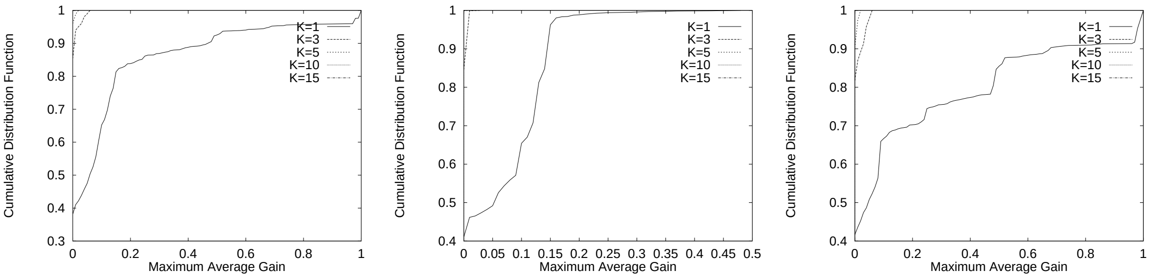

Figure 8: Variable size packet networks: Maximum Average gain with Delay EDD in (a) EXP1, (b) EXP2, and (c) EXP3

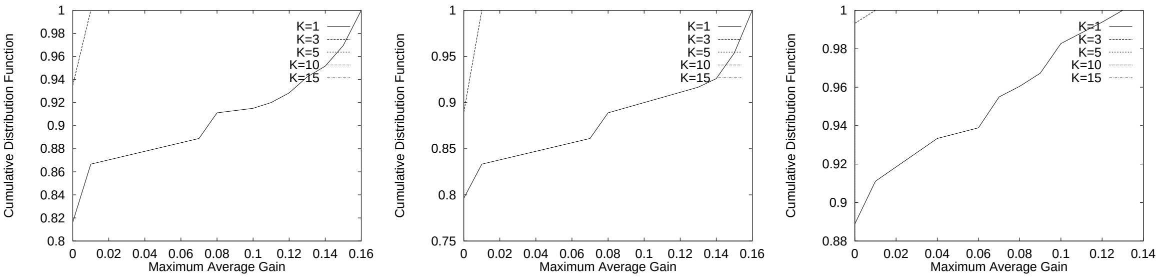

Figure 9: Fixed size packet networks with delay increased by 100 ms: Maximum Average gain with Delay EDD in (a) EXP1, (b) EXP2, and (c) EXP3

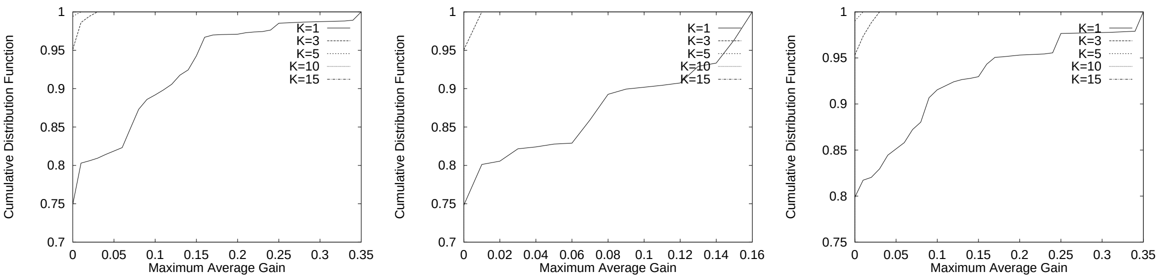

Figure 10: Variable size packet networks with delay increased by 100 ms: Maximum Average gain with Delay EDD in (a) EXP1, (b) EXP2, and (c) EXP3

4.3 Experiments with Randomly Generated Flows

In our experiments so far, we have assumed that there are only two types of flows at a server. In this section, we determine the difference in achievable utilization when the flows have heterogeneous characteristics. To achieve this objective, we generate a list of flows that have heterogeneous characteristics and then determine the subset of flows accepted in RCSD and GR using two methods:

1. We consider flows for acceptance in sequential order and determine the subset of flows that are accepted by a RCSD server using the schedulability conditions presented before. We then determine the subset of flows accepted by a GR server in two steps:

   - We determine the subset of the accepted flows that can be scheduled by a GR server using the schedulability conditions for Delay EDD and $\epsilon = 0.1, 0.3, 0.5, 0.7$, and 0.9.
   - If all the flows accepted by RCSD are accepted by GR, we determine the subset of the remaining flows that can be accepted by GR.

   Let Q^r and Q^g be the set of accepted flows in RCSD and GR, respectively. Define utilization of RCSD and GR servers, denoted by u^r and u^g respectively, as $u^r = \frac{1}{C}\sum_{i \in Q^r} r_i$ and $u^g = \frac{1}{C}\sum_{i \in Q^g} r_i$ where C is the link capacity. Then RCSD utilization gain, U^r is defined as: $U^r = \frac{u^r - u^g}{u^g}$.

2. This method is the same as the previous with the only change that roles of GR and RCSD are reversed. In this case, we define GR utilization gain, U^g as: $U^g = \frac{u^g - u^r}{u^r}$.

In an experiment, we generate a large number of different list of flows, determine U^r and U^g for each list, and determine the cumulative distribution function of both the metrics. The lists of flows are generated as follows. We choose range of burstiness factor $[b_{min}, b_{max}]$, packet lengths $[l_{min}, l_{max}]$, rate $[r_{min}, r_{max}]$, delay $[d_{min}, d_{max}]$, and number of servers on the path $[k_{min}, k_{max}]$. We then generate the quintuple $(\sigma_f, r_f, l_f, d_f, K_f)$ for a flow by choosing r_f, l_f,d_f, and K_f from their respective ranges with uniform probability. We choose burstiness factor b_f uniformly from its range and set $\sigma_f = b_f l_f$. A list of flows consists of such randomly generated flows and has at most one flow more than that required to ensure that the aggregate rate of the flows in the list is at least that of the link capacity.

We conduct 6 experiments and generate $100,000$ lists in each experiment. The ranges for all the factors other than delay are kept same in all the experiments. The ranges that we choose are: $[b_{min}, b_{max}] = [1, 20]$, $[l_{min}, l_{max}] = [512, 8196]$ bits, $[r_{min}, r_{max}] = [32, 2056]$ Kb/s, and $[k_{min}, k_{max}] = [3, 15]$. The six different ranges that we choose for delay are: $[25, 200]$ ms, $[100, 200]$ ms, $[100, 300]$ ms, $[100, 400]$ ms,

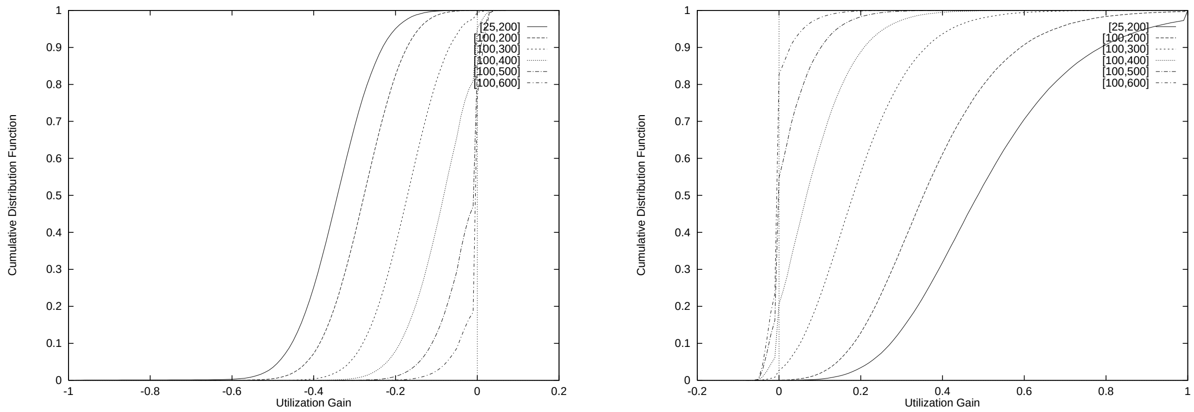

Figure 11: Randomly generated flows: (a) U^r (b) U^g

$[100, 500]$ ms, and $[100, 600]$ ms. Figures 11(a) and 11(b) plot the metrics U^r and U^g for these experiments, respectively. As both the figures demonstrate, in most cases, RCSD has lower utilization than GR, but the loss decreases with increase in the upper bound on delay. Observe that the distribution of the gain in utilization of RCSD is similar to the experiments conducted in the previous section. Hence, it appears that the results are not very correlated with the particular choice of parameters.

5 Discussion

The experiments in the previous section demonstrate that, in most cases, RCSD networks that employ non-CBR shapers have lower achievable utilization than GR networks. In few cases that they have higher utilization, the increase is small. Since RCSD networks with non-CBR shapers have higher buffer requirement than GR networks, we conclude that additional buffer does not increase the achievable utilization of a network providing deterministic guarantees. Furthermore, contrary to intuition, use of additional buffer may decrease the achievable utilization significantly.

It may be argued that since additional buffer does increase utilization sometimes, although by very small percentage, it may be used only when it increases the utilization and not otherwise. However, an algorithm for determining when to use additional buffers such that utilization always increases is not known. Furthermore, in many of our experiments in the previous section, the schedulability region is similar in structure to the hypothetical example shown in Figure 2. In such a case, the utility of additional buffer is dependent not only on the characteristics of the flows but also the specific combination of flows that have to be supported. Thus, it may be difficult to determine utility of additional buffers in a network where the set of flows may change dynamically.

In our experiments, we have focussed on the gain in maximum achievable utilization of a network. Though buffer does not help in increasing the achievable utilization, it may be argued that it may improve other measures of network performance such as call blocking probability. However, our experiments indicate that this is unlikely as most (99% or higher) of the times, the maximum average utilization gain is 0. Unless the flow arrival process is such that the network operates most of the time in the small region where the maximum average utilization gain is non-zero, buffer would not yield reduced call blocking probabilities.

Though additional buffer at the servers does not increase achievable utilization when a network provides deterministic guarantees, it may be desirable for other reasons. If the burstiness of the sources is not smoothed at the edge and additional buffer is employed at the switches to absorb the burstiness, then due to the resultant overall work conserving nature of the systems, the packets of sources will experience smaller average delay. Furthermore, a network may be able to achieve better statistical multiplexing of best effort packets and thereby achieve higher overall throughput [12]. Finally, a network that supports deterministic service as well as a service that provides guaranteed throughput, but not delay, may be able to employ additional buffer for deterministic service to increase its overall utilization [2].

6 Concluding Remarks

A theoretical investigation to determine whether source traffic should be completely smoothed or not when a network provides deterministic guarantees has been carried out in [5]. However, [5] makes several simplifying assumptions (for example, the differences in the delay requirements of flows are not accounted for) that make it inapplicable to the more general problem that we have investigated. We are not aware of any other theoretical or experimental study that investigates the utility of buffer in a network providing deterministic guarantees.

In this paper, we took a step towards determining the effectiveness of buffer in increasing achievable utilization of networks providing deterministic guarantees. To determine the increase, we classified the packet scheduling algorithms as either belonging to the class of Guaranteed Rate (GR) scheduling algorithms or Rate Controlled Service Disciplines (RCSD). We experimentally determined the difference in achievable utilization of RCSD and GR servers when RCSD servers employ additional buffers. Our experiments demonstrated that *contrary to intuition*, in most cases, additional buffers in a network of RCSD servers do not lead to higher achievable utilization. This experimental result indicates that if a source desires deterministic bounds on packet delay, then the traffic should be completely smoothed at the source [9].

Though our experiments explored large parameter space (more than 100,000 schedulability regions and 700,000 sets of flows were generated), they are not exhaustive. Hence, they should be considered as the first step in evaluating effectiveness of buffer in increasing achievable utilization of networks providing deterministic guarantees. We plan to extend this study in several ways. For example, we plan to evaluate the effectiveness of buffer for deterministic guarantees when the sources are characterized by models such as D-BIND [11].

References

[1] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proceedings of ACM SIGCOMM'89*, pages 1–12, September 1989.

[2] L. Georgiadis, R. Guerin, V. Peris, and R. Rajan. Efficient Support of Delay and Rate Guarantees in an Internet. In *Proceedings of ACM SIGCOMM'96*, pages 106–116, August 1996.

[3] L. Georgiadis, R. Guerin, V. Peris, and K.N. Sivarajan. Efficient Network QoS Provisioning Based on per Node Traffic Shaping. In *Proceedings of INFOCOM'96*, pages 102–110, March 1996.

[4] S.J. Golestani. A Self-Clocked Fair Queueing Scheme for High Speed Applications. In *Proceedings of INFOCOM'94*, pages 636–646, April 1994.

[5] S. Gorinsky, S. Baruah, and A. Stoyenko. Traffic Reshaping in Packet-Switched Virtual-Circuit Fixed-Packet Networks. In *Proceedings of Workshop on Resource Allocation Problems in Multimedia Systems*, December 1996. Available via URL: http://www.cs.unc.edu/ jeffay/mm-wrkshp96.html.

[6] P. Goyal, S. S. Lam, and H. M. Vin. Determining End-to-End Delay Bounds In Heterogeneous Networks. *ACM/Springer-Verlag Multimedia Systems Journal*, 5(3):157–163, May 1997. Also appeared in the Proceedings of the Workshop on Network and Operating System Support for Digital Audio and Video, Pages 287-298, April 1995.

[7] P. Goyal and H. M. Vin. Generalized Guaranteed Rate Scheduling Algorithms: A Framework. In *IEEE/ACM Transactions on Networking*, August 1997.

[8] P. Goyal, H. M. Vin, and H. Cheng. Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. In *Proceedings of ACM SIGCOMM'96*, pages 157–168, August 1996.

[9] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic. In *Proceedings of ACM SIGCOMM'95*, pages 219–230, August 1995.

[10] J. D. Guyton and M. F. Schwartz. Locating Nearby Copies of Replicated Internet Servers. In *Proceedings of ACM SIGCOMM'95*, pages 288–298, August 1995.

[11] E. Knightly and H. Zhang. Traffic Characterization and Switch Utilization Using a Deterministic Bounding Interval Dependent Traffic Model. In *Proceedings of INFOCOM'95, Boston, MA*, April 1995.

[12] S. Shenker. Contribution to the Int-serv mailing list. July 1995.

[13] D. Stiliadis and A. Varma. Design and Analysis of Frame-based Fair Queueing: A New Traffic Scheduling Algorithm for Packet Switched Networks. In *Proceedings of SIGMETRICS'96*, May 1996.

[14] S. Suri, G. Varghese, and G. Chandramenon. Leap Forward Virtual Clock: A New Fair Queuing Scheme with Guaranteed Delays and Throughput fairness. In *Proceedings of INFOCOM'97*, April 1997.

[15] G.G. Xie and S.S. Lam. Delay Guarantee of Virtual Clock Server. *IEEE/ACM Transactions on Networking*, 3(6):683–689, December 1995.

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

[17] H. Zhang and S. Keshav. Comparison of Rate-Based Service Disciplines. In *Proceedings of ACM SIGCOMM*, pages 113–121, August 1991.

[18] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proceedings of ACM SIGCOMM'90*, pages 19–29, August 1990.