In *ACM Multimedia Systems Journal*, 4(6):346-356, 1996

RCSP and Stop-and-Go: A Comparison of Two Non-Work-Conserving Disciplines For Supporting Multimedia Communication

Hui Zhang
School of Computer Science
Carnegie Mellon University
hzhang@cs.cmu.edu

Edward W. Knightly
ECE Department
Rice University
knightly@ece.rice.edu

Abstract

To support emerging real-time applications, high speed integrated services networks need to provide end-to-end performance guarantees on a per-connection basis in a networking environment. In addition to the issue of how to allocate resources to meet diverse QOS requirements in a single switch, resource management algorithms also need to account for the fact that traffic may get burstier and burstier as it traverses the network due to complex interaction among packet streams at each switch. To address this problem, several non-work-conserving packet service disciplines have been proposed. A non-work-conserving server is one that may be idle even when there are packets available to be sent. By holding packets under certain conditions, non-work-conserving servers fully or partially reconstruct the traffic pattern of the original source inside the network, and prevents the traffic from becoming burstier.

In this paper, we compare two non-work-conserving service disciplines: Stop-and-Go and Rate-Controlled Static Priority or RCSP. Stop-and-Go uses a multi-level framing strategy to allocate resources in a single switch and to ensure traffic smoothness throughout the network. RCSP decouples the server functions by having two components: a regulator to control traffic distortion introduced by multiplexing effects and load fluctuations in previous servers, and a static priority scheduler to multiplex the regulated traffic. We compare the two service disciplines in terms of traffic specification, scheduling mechanism, buffer space requirement, end-to-end delay characteristics, connection admission control algorithms, and achievable network utilization. The comparison is first done analytically, and then empirically by using two 10-minute traces of MPEG compressed video.

1 Introduction

High speed networking has introduced opportunities for new multimedia applications such as video conferencing, scientific visualization, and medical imaging. These applications have stringent network performance requirements in terms of combinations of parameters such as throughput, delay, delay-jitter, and loss-rate. To support these new applications, networks need to provide real-time communication services that allow network clients to transport information with performance guarantees expressed in terms of these parameters. It has been argued that a connection-oriented architecture with explicit resource allocation and connection admission control is needed to offer such a real-time service [7]. In a packet-switching network, packets from different connections will interact with each other at each multiplexing point. Without proper control, these interactions may adversely affect the network performance experienced by clients. The service disciplines at the switching nodes, which control the order in which packets are serviced, determine how packets from different connections interact with each other.

Service disciplines and associated performance problems have been widely studied in the contexts of hard real-time systems and queueing systems. However, results from these studies are not directly applicable to integrated-services networks for the following reasons. Analyses of hard real-time systems usually assume a single server environment, periodic jobs, and the job delay bounded by its period [27]. However, network traffic is bursty, and the delay constraint for each individual connection is independent of its bandwidth requirement. In addition, bounds on *end-to-end* performance need to be guaranteed in a *networking* environment, where traffic dynamics are far more complex than in a single server environment. Queueing analysis is also difficult to apply to this problem since it is often intractable for realistic traffic models. Moreover, classical queueing analyses usually study *average* performance for *aggregate* traffic [16] – in integrated-services networks, performance bounds need to be derived on a *per-connection* basis [6, 20]. In addition to the challenge of providing end-to-end per-connection performance guarantees to heterogeneous, bursty traffic sources, service disciplines must be *simple* enough to be implemented at very high speeds.

A service discipline can be classified as either work-conserving or non-work-conserving. With a work-conserving discipline, a server is never idle when there is a packet to send. With a non-work-conserving discipline, the server may be idle even when there are packets waiting for transmission. Recent studies suggest that non-work-conserving disciplines have some unique characteristics that make them suitable for providing performance guarantees in packet switching networks [34, 36]. In this paper, we compare two representative non-work-conserving disciplines proposed in the context of high speed networks: Stop-and-Go [10] and Rate-Controlled Static Priority or RCSP [32]. Although both Stop-and-Go and RCSP can be used to provide statistical guarantees [12, 35], only deterministic performance guarantees are considered in this paper.

The remainder of the paper is organized as the follows. We first discuss the background and motivate the need for non-work-conserving disciplines in Section 2. We then review the two disciplines and compare them by casting them into the same framework of rate-controlled service disciplines in Section 3. In Section 4, we discuss the admission control algorithms, end-to-end delay characteristics, and buffer space requirements for the two disciplines. In Section 5, we quantitatively compare the maximum network utilization that can be achieved by the two disciplines. Finally, in Sections 6, Section 7, and 8 we respectively examine implementation issues, review related work, and conclude.

2 Background

In order to provide end-to-end performance guarantees on a per connection basis, a connection-oriented and reservation-based architecture is needed [7]. In such an architecture, there are two phases in a communication session: connection establishment and data transfer. During the connection establishment, the client first specifies its end-to-end traffic and performance parameters to the network. The network then translates them into local parameters, and performs a set of connection admission control tests with the local parameters at each switch. The new connection is accepted only if there are enough resources to guarantee its performance at all switches along the path. During data transfers, each switch will transmit packets from different connections according to a packet service discipline. By ensuring that the local performance requirements are met at each switch, the end-to-end performance requirements can be satisfied. Notice that there are two levels of control in this paradigm: the connection admission control at the connection level, and the service discipline at the packet level. A complete solution needs to specify both the service discipline and the associated connection admission control conditions.

A switch can provide local performance guarantees to a connection only when the traffic on that connection behaves according to its specified traffic characteristics. However, network load fluctuations at previous switches may distort the traffic pattern of a connection and cause an instantaneous higher rate at some switch even when the connection satisfies the client-specified rate constraint at the entrance to the network. Since a connection's local performance bounds can be guaranteed only if its input traffic at the switch satisfies a certain traffic characterization, traffic pattern distortions may make it difficult to guarantee local performance bounds at the switches inside the network.

One solution to this problem is to *characterize* the traffic pattern distortion inside the network, and derive the traffic characterization at the entrance to each switch from characterizations of the source traffic and of the traffic pattern distortions [1, 4, 19, 25]. In general, characterizing traffic inside the network is difficult. In networks with *work-conserving* service disciplines, even in situations when the traffic inside the network can be characterized, the worst-case traffic is usually more bursty inside the network than that at the entrance. This property is independent of the traffic model being used. In [4], a deterministic fluid model (σ, ρ) is used to characterize a traffic source. A source is said to satisfy (σ, ρ) if during any time interval of length u, the amount of its output traffic is less than

$\sigma + \rho u$. In such a model, σ is the maximum burst size, and ρ is the average rate. If the traffic of connection j is characterized by (σ_j, ρ_j) at the entrance to the network, its characterization will be $(\sigma_j + \Delta\sigma_j^{i-1}, \rho_j)$ at the entrance to the i^{th} switch along the path, where $\Delta\sigma_j^{i-1} = \sum_{h=1}^{i-1} \rho_j d_{h,j}$ and $d_{h,j}$ is the local delay for the connection at the h^{th} switch. Compared to the characterization of the source traffic, the maximum burst size at the i^{th} switch increases by $\sum_{h=1}^{i-1} \rho_j d_{h,j}$. This maximum burst size grows monotonically along the path of the connection.

In [19], a family of stochastic random variables is used to characterize a source. Connection j is said to satisfy a characterization $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), (\mathbf{R}_{t_3,j}, t_3)...\}$, where the $\mathbf{R}_{t_i,j}$ are random variables, and $t_1 < t_2 < \cdots$ are time intervals, if $\mathbf{R}_{t_i,j}$ is *stochastically larger* than the number of packets generated over any interval of length t_i by source j. If the traffic on connection j is characterized by $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), (\mathbf{R}_{t_3,j}, t_3)...\}$ at the entrance to the network, its characterization will be $\{(\mathbf{R}_{t_1+\Delta t_j^{i-1},j}, t_1), (\mathbf{R}_{t_2+\Delta t_j^{i-1},j}, t_2), (\mathbf{R}_{t_3+\Delta t_j^{i-1},j}, t_3), ...\}$ at the h^{th} switch, where $\Delta t_j^{i-1} = \sum_{h=1}^{i-1} b_h$ and b_h is the length of the maximum busy period at the h^{th} switch. The same random variable that bounds the maximum number of packets over an interval at the *entrance* of the network, now bounds the maximum number of packets over a much *smaller* interval at the i^{th} switch. That is, the traffic characterization is more bursty at the h^{th} switch than at the entrance of the network.

In both the (σ_j, ρ_j) and $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), (\mathbf{R}_{t_3,j}, t_3)...\}$ analysis, the burstiness of a connection's traffic accumulates at each hop along the path from source to destination, and more resources need to be reserved for a connection with a burstier traffic characterization. For example, the amount of buffer space required to prevent packet loss for a connection must grow monotonically along the path.

Another approach to dealing with the problem of traffic pattern distortion is to *control* this distortion at each switch. By maintaining certain traffic characteristics throughout the network, non-work-conserving service disciplines such as Stop-and-Go and RCSP eliminate the problems of requiring more resources at downstream switches and characterizing traffic transformations inside the network. Also, these disciplines can provide end-to-end performance guarantees in networks of arbitrary topology.

3 Stop-and-Go and RCSP Service Disciplines

In this section, we review the traffic models used in Stop-and-Go and RCSP, and then describe and compare each discipline by casting them into the same framework of rate-controlled service disciplines [34].

3.1 Traffic Model

In order to allocate resources for each connection, sources must specify their traffic characteristics. In the literature, different traffic models have been used for different schedulers. For example, Stop-and-Go uses the (r, T) traffic model. A stream of packets is called (r, T)-smooth if during each frame of length T the total number of bits that are transmitted by the source is no more than $r \cdot T$ bits.

In the original proposal of RCSP [32], the $(Xmin, Xave, I, Smax)$ model was used [7]. In this model, $Xmin$ is the minimum packet inter-arrival time, $Xave$ is the average packet inter-arrival time over an averaging interval I, and $Smax$ is the maximum packet size. However, the RCSP algorithm is general and other traffic characterization can be used. For example, the (σ, ρ) model proposed in [4] may be used. In this case, RCSP's regulators are simply leaky buckets. Additionally, if a more elaborate Deterministic Bounding Interval Dependent (D-BIND) model is used [18], the flexibility offered by RCSP allows the admission control algorithm to accept additional connections. This will be demonstrated in Section 5 using real traffic traces.

3.2 Stop-and-Go

Stop-and-Go uses a framing strategy [10]. In such a strategy, the time axis is divided into frames, which are periods of some constant length T. Stop-and-Go defines $departing$ and $arriving$ frames for each link. At each switch, the arriving frame of each incoming link is mapped to the departing frame of the output link by introducing a constant delay θ, where $0 \leq \theta < T$. All the packets from one arriving frame of an incoming link and going to output link l are delayed by θ and put into the corresponding departing frame of l. According to the Stop-and-Go discipline, the

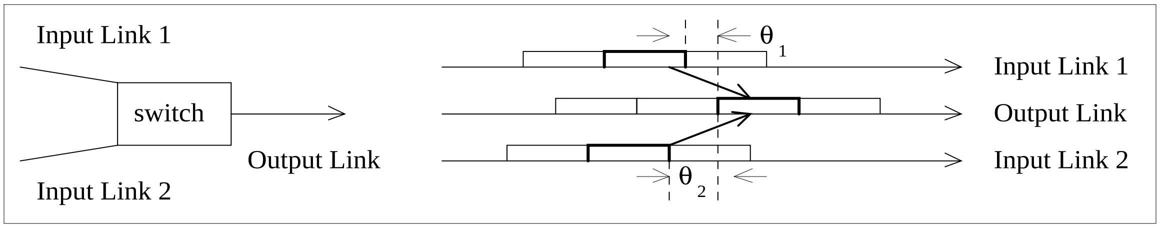


Figure 1: Synchronization between input and output links in Stop-and-Go

transmission of a packet that has arrived on any link l during a frame f should always be postponed until the beginning of the next frame. Since packets arriving during a frame f of the output link are not eligible for transmission until the next frame, the output link may be left idle even when there are packets in the switch to be transmitted. Thus, Stop-and-Go is a non-work-conserving policy.

In Stop-and-Go, bandwidth is allocated to each connection as a certain fraction of the frame time. As for delay, by using the admission control algorithms discussed in Section 4, Stop-and-Go ensures that all packets coming on one arriving frame of the input link will always go out on the next departing framing of the output link.

The framing mechanism also limits the traffic-pattern distortion and maintains (r, T) smoothness throughout the network. This is stated in the following proposition.

Proposition 1 *Consider a connection that traverses a cascade of Stop-and-Go servers. If the connection satisfies* (r, T) *smoothness at the entrance to the network, and each server ensures that packets coming in on one arriving frame of the input link will always go out on the next departing frame of the output link, the connection will satisfy* (r, T) *smoothness at each of the servers throughout the network.*

By maintaining traffic smoothness throughout the network, end-to-end delay bounds can be guaranteed in a network of arbitrary topology as long as each server can ensure local delay bounds for (r, T) smooth traffic. As discussed in [34], one of the most important advantages of non-work-conserving disciplines like Stop-and-Go is that they greatly simplify the analysis in a networking environment by allowing a single node analysis to be extended to networks of arbitrary topology.

The framing strategy also introduces the problem of coupling between delay bound and bandwidth allocation granularity. The delay of any packet at a single switch is bounded by a multiple of frame times. To reduce the delay, a smaller T is desired. However, since T is also used to specify traffic, it is tied to bandwidth allocation granularity. Assuming a fixed packet size P, the minimum granularity of bandwidth allocation is $\frac{P}{T}$. To have more flexibility in allocating bandwidth, or a smaller bandwidth allocation granularity, a larger T is preferred. It is clear that low delay bound and fine granularity of bandwidth allocation cannot be achieved simultaneously in a framing strategy like Stop-and-Go.

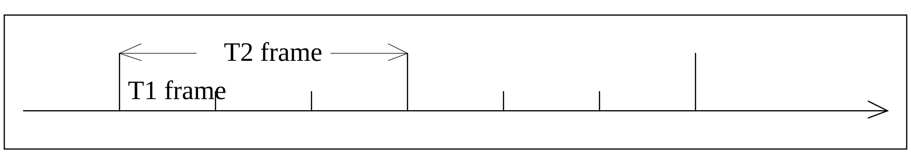


Figure 2: Two levels of framing with $T_2 = 3T_1$

To get around this coupling problem, a generalized version of Stop-and-Go with multiple frame sizes is proposed [11]. In the generalized Stop-and-Go, the time axis is divided into a hierarchical framing structure as shown in Figure 2. For n level framing with frame sizes $T_1, \cdots, T_n$, and $T_{m+1} = K_m T_m$ for $m = 1, \cdots, n-1$, packets on a level p connection need to observe the Stop-and-Go rule with frame size T_p. That is, packets which have arrived at an output link during a T_p frame, will not become eligible for transmission until the start of next T_p frame. Also, for two packets

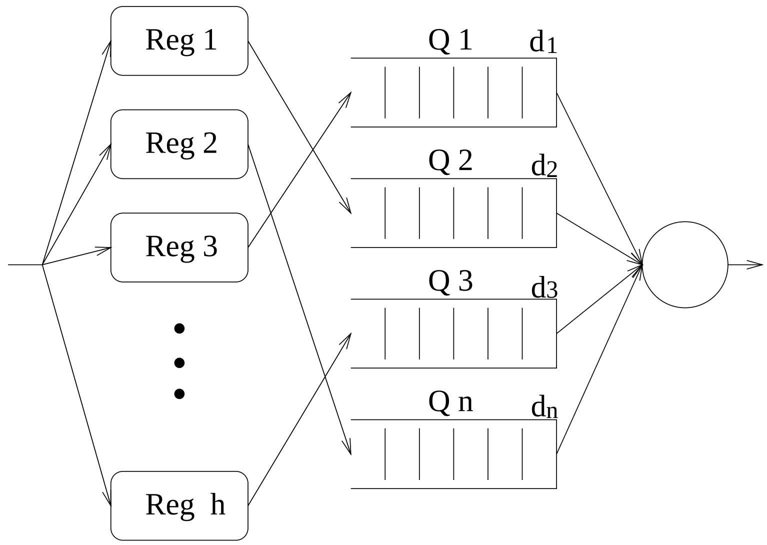


Figure 3: Rate-Controlled Static-Priority Queueing

with different frame sizes, the packet with a smaller frame size has non-preemptive priority over the packet with a larger frame size.

With multi-frame Stop-and-Go, it is possible to provide low delay bounds to some channels by putting them in frames with a smaller frame time, and to allocate bandwidth with fine granularity to other channels by putting them in levels with a larger frame time. However, the coupling between delay and service quantum still exists within each frame.

3.3 Rate-Controlled Static-Priority

A Rate-Controlled Static-Priority server differs from a Stop-and-Go server in that it uses *two* components to allocate delay bounds and bandwidth instead of one framing structure [32]. These two components are a rate controller and a static-priority scheduler. The rate controller shapes the input traffic from each connection into the desired traffic pattern by assigning an eligibility time to each packet. The scheduler then orders the transmission of eligible packets from all the connections. The architecture of the RCSP server is shown in Figure 3.

Conceptually, a rate controller consists of a set of regulators corresponding to each of the connections traversing the switch. Upon the arrival of each packet, the regulator assigns an eligibility time for the packet, and holds the packet until that time before handing it to the scheduler. Different ways of calculating the eligibility time of a packet will result in different types of regulators so that a source's traffic pattern may be partially of fully reconstructed. One possible regulator for the RCSP scheduler is the leaky bucket, which is based on enforcing the (σ, ρ) traffic specification.

A second possible RCSP regulator is a delay-jitter (DJ) regulator. In this case, the scheduler absorbs the delay variation introduced by the previous switch so that at the input to the priority queues, the original traffic pattern is completely reconstructed (with the exception of packets that have been dropped or lost). The eligibility time of a packet for a DJ regulator is defined with reference to the eligibility time of the same packet at the immediately upstream switch. The definition assumes that the queueing delays of packets on the connection, and the link delay from the upstream switch to the current switch, are bounded. Let d_{i-1} be the local delay bound for the connection in the scheduler at switch $i-1$, and $\overline{\pi}_i$ be the maximum link delay from switch $i-1$ to switch i. For a delay-jitter controlling regulator, ET_i^k, the eligibility time of the k^{th} packet on a connection that traverses switch i is defined as:

$$ET_0^k = AT_0^k \tag{1}$$

$$ET_i^k = ET_{i-1}^k + d_{i-1} + \overline{\pi_i}, \quad i > 0 \tag{2}$$

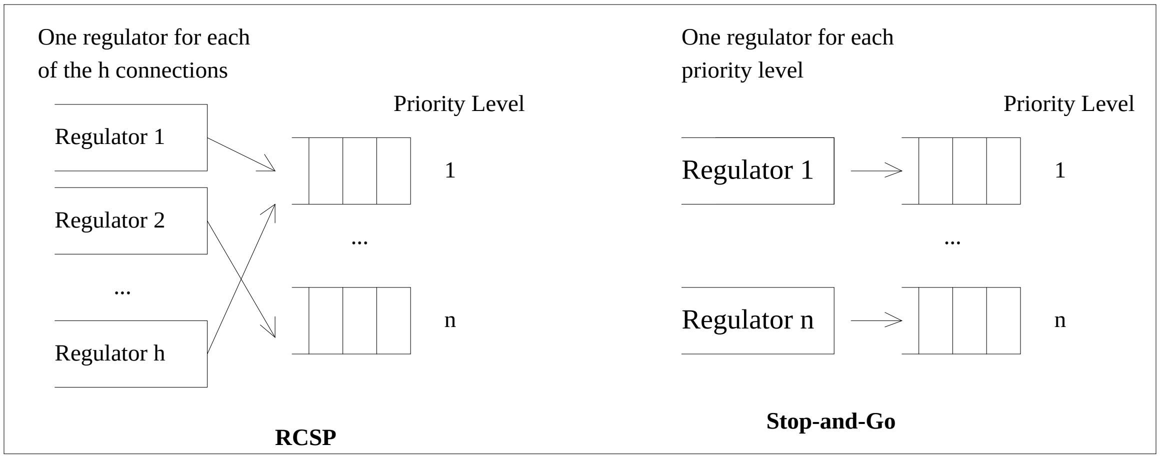


Figure 4: Stop-and-Go and RCSP

where switch 0 is the source of the connection, and AT_0^k is the arrival time of the k^{th} packet at the entrance to the network.

Notice that no traffic pattern or traffic model is assumed in the definition. That is, by turning the network into a constant delay line, the DJ regulators reconstruct the exact original traffic pattern of the source which is independent of the manner in which the source is defined or parameterized.

For a DJ regulator, it is easy to show that the following holds:

$$ET_i^{k+1} - ET_i^k = AT_0^{k+1} - AT_0^k \quad \forall k, i \geq 0 \tag{3}$$

This leads to the following proposition:

Proposition 2 *Consider a connection that traverses a cascade of RCSP servers with DJ regulators. If deterministic delay bounds can be provided at the scheduler of each RCSP server, the traffic pattern of the connection at the output of each rate-controller is exactly the same as the traffic pattern of the connection at the* entrance *to the network.*

This proposition is more general than Proposition 1. It applies to any traffic specification, rather than just (r, T) smoothness. In the original discussion of RCSP [32], the traffic model $(Xmin, Xave, I, Smax)$ is used. In [18], a more accurate D-BIND model is proposed. We will show in Section 5 that RCSP's flexibility of allowing the use of more accurate traffic models will increase the number of connections that can be admitted into the network.

The second component of an RCSP server is the scheduler. The scheduler services packets using a non-preemptive static-priority discipline which non-preemptively chooses packets in FCFS order from the highest-priority non-empty queue. Non-real-time packets are serviced only when there are no real-time packets; their service order is not specified.

3.4 Framing vs. Decoupling of Rate-Control and Scheduler

There are many similarities between Stop-and-Go and RCSP. As discussed in previous sections, both disciplines maintain traffic characteristics throughout the network by holding packets inside the network and both disciplines employ multiple priority levels to meet diverse QOS requirements for different connections. Also, as will be shown in Section 4, both disciplines, when used with associated connection admission control algorithms, can guarantee end-to-end delay and delay-jitter bounds in networks of arbitrary topology. However, an important difference between the two disciplines is that Stop-and-Go uses one mechanism, the framing strategy, to allocate both bandwidths and delay bounds. Alternatively, RCSP decouples the two functions by using two components, a rate-controller and a scheduler. In this section, we discuss the implications and tradeoffs of this important difference.

As shown in [34], both Stop-and-Go and RCSP belong to a class of non-work-conserving disciplines called rate-controlled service disciplines. A rate-controlled server has two components: a rate-controller and a scheduler. Combinations of different rate-controllers and schedulers result in different rate-controlled disciplines. RCSP is one instance in this class with delay-jitter-controlled regulators and a static priority scheduler. Stop-and-Go can also be implemented using a rate-controlled service discipline as defined in Proposition 3. By casting Stop-and-Go into the framework of rate-controlled service disciplines, it is easier to see the similarities and differences between Stop-and-Go and RCSP.

Proposition 3 *A Stop-and-Go server with n frame sizes ($T_1 < T_2 < \ldots < T_n$) can be implemented by a rate-controlled service discipline with a variation of delay-jitter controlling regulators, which we call DJ_s regulators, and an n-level static priority scheduler. In a DJ_s regulator, the eligibility time for packet k at the i^{th} switch along the path is defined as follows:*

$$ET_i^k = AT_i^k + Ahead_{i-1}^k + \theta \tag{4}$$

where $Ahead_{i-1}^k$ is the amount of time the packet is ahead of schedule in the $i-1^{th}$ switch, and θ is the synchronization time between the framing structures on the input and output links. Each pair of input and output links in a switch may have a different value of θ. In the static priority scheduler, the delay bound associated with level m is T_m, $1 \leq m \leq n$.

Although the above implementation of Stop-and-Go is very similar to RCSP, there are also important differences. Figure 4 shows an RCSP server and a Stop-and-Go server. As can be seen, in an RCSP server, there is a regulator for each connection, and the regulated traffic on each connection can be assigned to *any* priority level in the scheduler. Alternatively, in a Stop-and-Go server, regulators are associated with priority levels in the scheduler. In fact, there is a one-to-one correspondence between the regulator and the priority level. Packets from one regulator can only go to the queue of the corresponding priority level. This introduces a coupling between the allocations of bandwidth and delay bounds. The traffic has to be specified with respect to the frame size that corresponds to the connection's assigned priority level. Since the frame size is also the local delay bound, the coupling between the traffic specification and the delay allocation implies that the admission control algorithm has to be based on a busy period argument, which tends out to produce looser bounds when compared to more elaborate analysis [4, 33]. This will be discussed in more detail in Section 4.3.

Because of the framing, there are dependencies among the local delay bounds at each priority level in a Stop-and-Go server. In particular, $T_{m+1} = K_m T_m$ must hold, with $1 \leq m < n$, and K_m being an integer. In addition, the delay bound allocations for each connection in different switches are coupled with one another. In [11], a connection has to have the same frame size in all the switches. In [36], a looser requirement is presented: the frame times of a connection along the path should be non-decreasing. None of these restrictions apply to RCSP. The impact of flexibility of allocating delay bounds inside the network on network utilization was studied in [23].

4 Admission Control Conditions and End-to-End Delay Properties

As discussed in Section 2, a service discipline alone cannot provide performance guarantees. Admission control algorithms are also needed to ensure that the network has enough resources to meet the performance requirements of all the connections. Different service disciplines have different corresponding admission control conditions. In this section, we first review the admission control conditions of Stop-and-Go and RCSP. We then compare the conditions by using deterministic fluid model analysis developed in [4].

4.1 Stop-and-Go

In Stop-and-Go, the connection admission control algorithm needs to ensure that packets from an incoming frame of an input link will always go out on the next departing frame of the output link. The following theorem gives the condition.

Theorem 1 *Consider a Stop-and-Go server of n priority levels with frame sizes $T_1, \cdots, T_n$. Let C_q be the set of the connections at level q, and the j^{th} connection in C_q satisfies the traffic specification (r_j^q, T_q). For a link speed l, and a maximum packet size of $\overline{Smax}$, any packet arriving in a T_m frame of an incoming link will be serviced before the end* *of the next T_m frame of the output link if*

$$\sum_{q=1}^{m}\sum_{j\in C_q} r_j^q T_m + \overline{Smax} \leq l T_m. \tag{5}$$

The proof is in [11]. Intuitively, $\sum_{q=1}^{m}\sum_{j\in C_q} r_j^q T_m$ is the maximum of bits that can arrive during an interval of length T_m from all connections with priority equal to or larger than a connection at T_m level, and lT_m is the maximum number of bits that can be transmitted during a interval of length T_m. The inequality ensures that the maximum busy period of packets with a priority equal to or larger than level m is less than T_m. The theorem then follows directly from a busy period argument: the maximum busy period is an upper bound on the delay of any work-conserving policies (notice that the service policy for all eligible packets is work-conserving).

The next theorem gives the end-to-end delay property of a connection in a network of Stop-and-Go servers.

Theorem 2 *Consider a connection that traverses n Stop-and-Go switches connected in cascade with π_i being the link delay between the $i-1^{th}$ and the i^{th} switch. If the connection is assigned to the frame of size T and Theorem 1 holds for the T-sized frame at all switches, the end-to-end delay and delay jitter of the connection is bounded by $D + \sum_{i=2}^{n}\pi_i$ and T, where $nT \leq D \leq 2nT$ holds.*

The proof is given in [11].

4.2 RCSP

In RCSP, the admission control algorithm needs to ensure that local delay bounds can be guaranteed for each connection at the scheduler. In order to perform admission control tests, the traffic characteristics must be specified for each guaranteed performance connection traversing the server. As mentioned in Section 3.3, many traffic models can be used. In [32, 33], admission control conditions were given for the $(Xmin, Xave, I, Smax)$ model and in [18] for the D-BIND model. The following theorem gives the algorithm control condition for RCSP using a general traffic constraint function $b(\cdot)$, where $b_j(u)$ is defined to be the maximum number of bits that can arrive on connection j during *any* interval of length u. Different bounding traffic models such as $(Xmin, Xave, I, Smax)$, D-BIND, and (σ, ρ) have different corresponding traffic constraint functions (see [17]).

Theorem 3 *Consider a Static Priority scheduler with n priority levels. Let C_q be the set of the connections at level q, and the j^{th} connection in C_q satisfies the traffic constraint function $b_{q,j}(\cdot)$. For a link speed l, and a maximum packet size of $\overline{Smax}$, the maximum delay of any packet at priority level m is bounded above by d^m, where*

$$d^m = max\{u : u \geq 0, b'_m(u) \geq l \times u\}$$

and $b'_m(\alpha)$ is defined for all α by

$$b'_m(\alpha) = \max_{\beta\geq 0}\{\overline{Smax} + \sum_{j\in C_m} b_{m,j}(\beta) + \sum_{q=1}^{m-1}\sum_{j\in C_q} b_{q,j}(\alpha+\beta) - l\times\beta\} \tag{6}$$

The proof is by extension of the results of [4]. Details may also be found in [31].

The end-to-end delay of a packet consists of the link delays the packet experienced and the residence times of the packet in each of the switches along the path. The residence time of a packet in a switch with rate-controlled servers has two components: the *holding* time in the regulator and the *waiting time* in the scheduler. Theorem 3 only bounds the waiting time in the scheduler. The next theorem, proven in [34], states that the end-to-end delays of all the packets on a connection can be bounded, as long as the delays on the links and the delays at each of the schedulers can be bounded. Holding packets in the rate controllers will *not* increase the *end-to-end delay bound* of the connection.

Theorem 4 *Consider a connection passing through n switches connected in cascade, with $\overline{\pi}_i$ and $\hat{\pi}_i$ being the upper and lower bounds on the delay of the link from the $(i-1)^{th}$ to the i^{th} switch. Assume that the scheduler of the i^{th} switch* *can guarantee that the delays of all the packets on the connection be bounded by d_i as long as the connection's input traffic to the scheduler satisfies the given $b_j(\cdot)$ constraint. If the traffic on the connection obeys the $b_j(\cdot)$ constraint at the entrance to the first switch, the end-to-end delay and the delay jitter of each of the connection's packets is bounded by $\sum_{i=1}^{n} d_i + \sum_{i=2}^{n} \overline{\pi}_i$ and $d_n + \overline{\pi}_n - \hat{\pi}_n$, respectively.*

Notice that in the theorem, we assume a model of links with *bounded*, but possibly *variable* delay. This is important for an internetwork environment, in which a link may be a subnetwork such as ATM or FDDI networks. It is possible to bound delay over these subnetworks; however, the delays for different packets will be *variable*.

4.3 Admission Control and Fluid Model Analysis

The admission control criteria above may be compared intuitively for the simple case when there is only one priority level. As described below, for the case of one priority level, the fundamental difference between the admission control bounds and thus the resulting network utilization is that Stop-and-Go relies on a busy period bound while RCSP relies on a backlog bound.

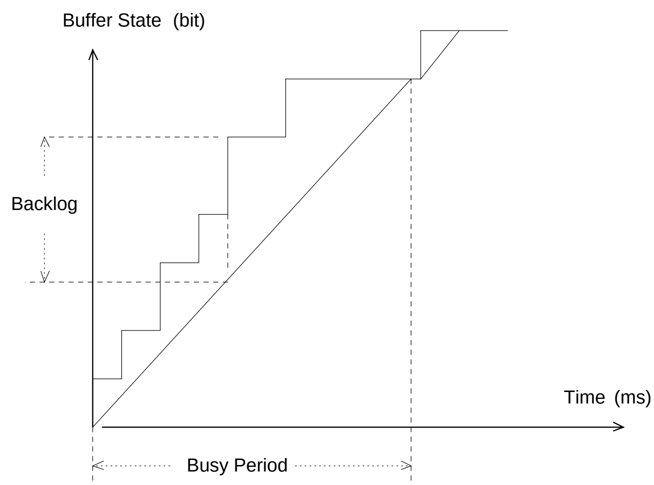


Figure 5: Backlog and Busy Period

Figure 5 illustrates this difference. The horizontal axis is time and the vertical axis is bits. The upper curve represents the total number of bits that have arrived in the queue by time t and the lower curve represents the total number of bits *transmitted* by time t. The difference between the two curves is the number of bits currently in the queue, or the *backlog* function. When the backlog function returns to zero (the two curves meet) there are no bits in the queue and thus a busy period has ended. Two key observations follow [4]:

- the maximum busy period provides an upper bound on delay for any work-conserving server;
- the maximum backlog divided by link speed provides an upper bound on delay for a FCFS server.

Delay bounds for other policies can also be expressed as the function of the two curves [1, 4, 17, 24].

The admission control criteria of Stop-and-Go relies on a bound on the busy period. That is, Stop-and-Go ensures that a busy period is bounded by a frame time so that during each frame time, all packets that arrived in the previous frame are guaranteed to be served. Note that in this case, to ensure that the busy period is bounded by T, the admission control criteria is that the total number of bits that arrive on all connections in an interval of length T is less than T times the link speed, as in (5).

Alternatively, since RCSP decouples the rate-controller and the scheduler, tighter analysis can be applied on the scheduler. For the case of one priority level, the delay bound is the maximum backlog divided by the link speed. For the case of multiple priority levels, the delay bound is a function of the link speed and the traffic constraint function as shown in (6). Notice that in both cases, the resulted delay bound is tighter than the bound based on the busy period because the maximum busy period is an upper bound on delay for *any* work-conserving server.

In order to provide a deterministic delay bound at a server, a bound is needed on each traffic source so that the total number of bits that arrive at the server in any time interval also has a deterministic bound. This source-constraint is specified by the client during connection establishment time using a traffic model, and is used by the network to calculate the delay bound. A tighter model will result in a lower constraint curve and thus a smaller backlog bound and hence a smaller delay bound since the maximum delay bound is simply the maximum backlog times the link speed. Since RCSP decouples the rate-controller and the scheduler, and the rate-controller can implement any regulating functions, different traffic models can be used. In Section 5, we will show quantitatively that, for RCSP, using the tighter D-BIND model will result in a higher network utilization than using the $(Xmin, Xave, I, Smax)$ model. In Stop-and-Go, since the traffic specification is tied to the framing structure, a more informative traffic model does not help.

4.4 Buffer Space Requirement

The maximum buffer size needed by a connection to prevent packet loss at a switch can be determined using the maximum residence time of packets at the switch and the maximum rate packets can arrive. In Stop-and-Go, for a connection with the (r, T) specification, the required buffer space is $3rT$. In RCSP, for a connection with constraint function $b(\cdot)$, the required buffer space is $b(d_i + d_{i-1} + \overline{\pi}_i - \hat{\pi}_i)$, where d_i and d_{i-1} are the respective local delay bounds for the current and the immediately up-stream switch; $\overline{\pi}_i$ and $\hat{\pi}_i$ are upper and lower bounds on the link-delay between the two switches. In particular, if the $(Xmin, Xave, I, Smax)$ traffic model is used and if the link delay is constant, the amount of buffer space required is $\lceil \frac{d_{i-1}+d_i}{Xmin} \rceil Smax$.

Notice that the buffer space requirement for a connection in Stop-and-Go and RCSP depends only on the local delay bounds at the current and the previous switches (in Stop-and-Go, they are both T). In contrast, for work-conserving policies, more buffer space is needed at downstream nodes due to the potential accumulated distortion to the traffic inside the network. For example, if a Delay-EDD scheduler is used, and the $(Xmin, Xave, I, Smax)$ traffic model is adopted, the amount of buffer space required at the i^{th} switch along the path traversed by the connection is $\lceil \frac{\sum_{h=1}^{i} d_h}{Xmin} \rceil Smax$, where d_h is the local delay bound at the h^{th} switch [7, 36].

Stop-and-Go and RCSP require less buffer space not only inside the network, but also at the destination node. In order to provide an isochronous service, $b(J)$ amount buffer space is needed at the destination where $b(\cdot)$ is the traffic constraint function, and J is the maximum end-to-end delay jitter. The end-to-end delay-jitters for Stop-and-Go and RCSP are the frame time and the last hop local delay bound respectively, while the end-to-end delay jitters for work-conserving policies are usually much larger.

5 Utilization Comparison with MPEG Traffic Traces

In this section, we use two 10 minute MPEG video traces to investigate the link utilizations that are achievable with the two disciplines. The two traces were chosen in that they likely represent the extremes in the spectrum of video types: one trace is taken from a series of advertisements where scenes are constantly and quickly changing in colorful, fluctuating surroundings. The second trace is taken from a lecture that has only two alternating scenes: a speaker and his transparencies. When the camera is focused on the speaker, there is some movement as the speaker moves and the camera zooms and pans. When the camera is focused on the transparencies, the transparency may change or be written on by the speaker in which case, motion of the hand, pen, and ink alter the scene.

A short segment of the trace of the advertisement sequence is shown in Figure 6 which depicts the instantaneous bit rate vs. frame number. The bandwidth of this trace is smaller than for many others because of the small frame size of 160 by 120. The general shape of the traces may be explained in terms of the mechanisms used in the MPEG standard. The coder generates three types of frames: I frames that use only *Intraframe* compression, and P and B frames that are transmitted between I frames and use *interframe* compression. While P frames (Predicted frames) are coded based on only past frames, B frames (Bidirectional frames) are coded based on both past and a future frame. With P and

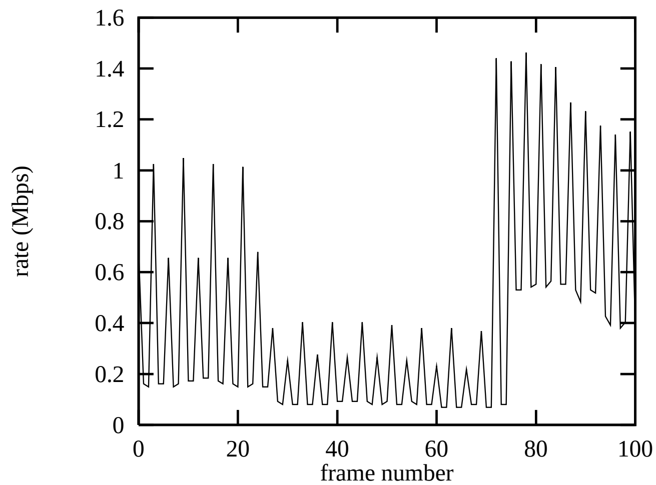


Figure 6: MPEG Video Trace

B frames, higher compression ratios can be achieved since the interframe coding makes use of motion compensation techniques. More details of the MPEG algorithm may be found in [9].

It is assumed that the entire frame is transmitted per frame time (as opposed to introducing additional delay by smoothing over several frames) so that Figure 6 is shows the frame sizes multipled by the frame rate (30 fps). Additionally, it is assumed that each frame is fragmented into 48 byte ATM cells with the cells being transmitted at equally spaced intervals over the frame time ($\frac{1}{30}^{th}$ of a second).

From the traces, we calculate the traffic model parameters for the various traffic models of Section 3.1. We then calculate the maximum number of connections that can be multiplexed on a T3 (45 Mbps) line with the given characterization. We consider three combinations of schedulers and traffic models (abbreviating Stop-and-Go as SG): RCSP/D-BIND, RCSP/Xmin, and SG/(r,T). For RCSP, as noted in Section 4, a tighter source model can result in more accepted connections and thus higher network utilization. For this reason, for RCSP we investigate the maximum number of connections that can be accepted for both the $(Xmin, Xave, I, Smax)$ model of [7] as well as the tighter D-BIND model of [18]. Alternatively, for Stop-and-Go, the admission control analysis can only make use of information on the maximum number of bits that a source will transmit in an interval of length T, where T is the frame time.

Figures 7(a) and 7(b) show the maximum number of connections that can be accepted for the advertisement sequence and the lecture sequence under the various scheduling schemes. The horizontal axes show the delay bound and the vertical axes show the maximum number of connections that can be multiplexed given the delay bound constraint. The three curves represent the three combinations of servers and traffic models described above.

Focusing first on the lecture sequence of Figure 7(a), there are several things to note. First, the general trend of the curves is that with an increasing delay bound, more connections can be accepted. Also, not shown is the fact that a peak-rate allocation scheme would result in 29 connections accepted. The ratio of the number of accepted connections to 29 what is termed in [18] as the Deterministic Multiplexing Gain (DMG). That is, even though all packets are deterministically guaranteed to meet their loss and delay bounds, sources may be multiplexed beyond their peak rate. For the RCSP/D-BIND curve, even for small delay bounds, DMG's significantly greater than 1 are achievable. For example, for a delay bound of 10 msec, 38 connections may be multiplexed for a DMG of 1.31. By a 40 msec delay bound, the DMG is 2.24.

The RCSP/Xmin curve is based on the $(Xmin, Xave, I, Smax)$ specification with I chosen to be 3 frame times or 100 msec. Note that this I, and the rest of the specification is fixed for all delay bound calculations. That is, for the SG calculations, we allow the admission control algorithm to choose the optimal value of I ($Xave$ and r should be viewed as functions of I). Thus, the RCSP/Xmin curve uses the bound in [33]. Once again, as the delay bound increases, so does the number of connections accepted until the point of scheduler saturation. The maximum DMG for this traffic

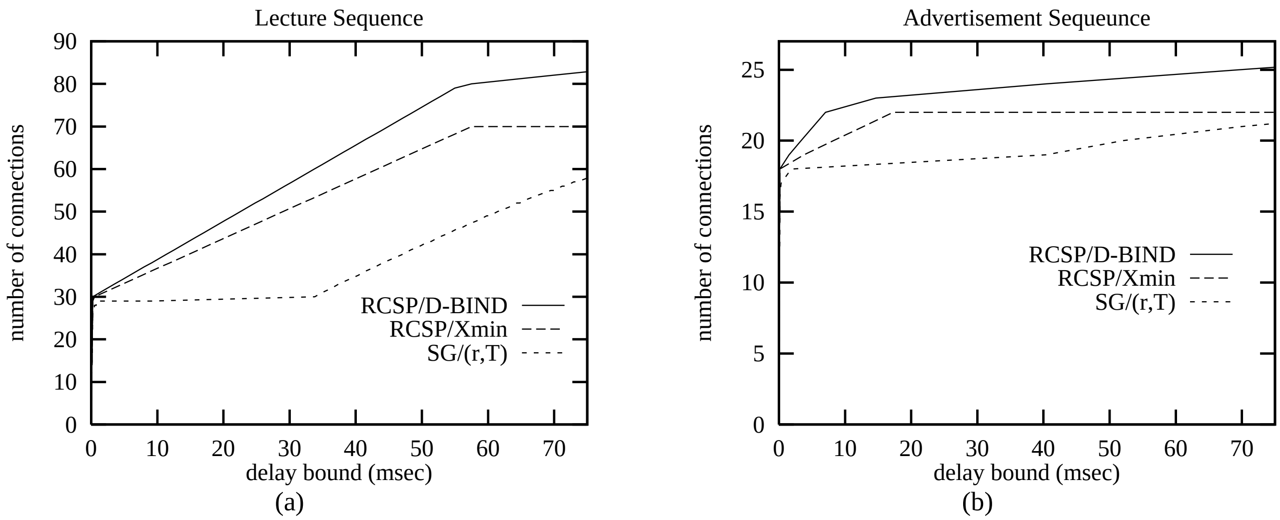


Figure 7: Connections Accepted vs. Delay Bound for SG and RCSP

specification and scheduler is 2.41 achieved at a queueing delay bound of 58 msec. As expected, the RCSP/Xmin curve is below the RCSP/D-BIND curve since the difference is that the D-BIND curve uses a more accurate source model with the same scheduler and analysis techniques.

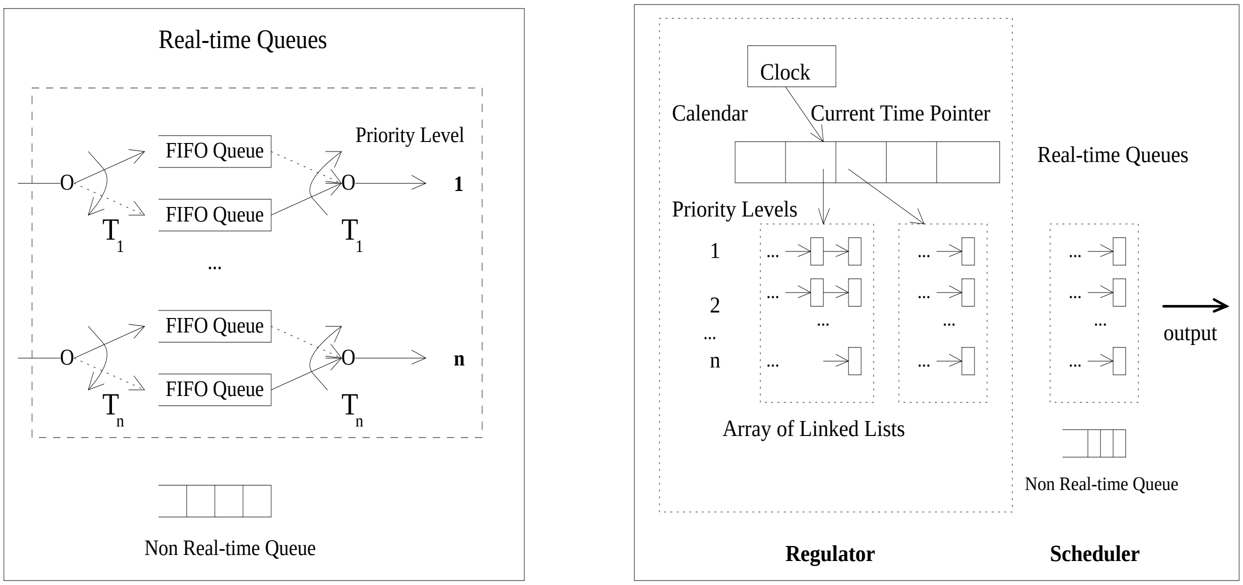
Finally for Figure 7(a), we explain the shape of the SG/(r,T) curve which represents the maximum number of connections that can be accepted by a SG scheduler for a given frame size or queueing delay bound T. First, note that the SG scheduler is not able to do better than peak-rate allocation (29 accepted connections) until the frame time T is greater than 33 msec. This may be explained in the following manner. Since the video frame rate is 30 fps, sources can send at their peak rate for 33 msec, 1/30th of a second, until the entire video frame is transmitted (again, the peak rate represents transmission of the largest video frame). Thus, if 29 connections are multiplexed, the busy period bound for FCFS will be 33 msec. However, the analysis of RCSP shows that the maximum backlog of the queue will result in a much smaller delay bound. Thus, by a delay bound of 33 msec, RCSP/D-BIND has accepted 59 connections. SG cannot accept a 30th connection until the SG frame time T is greater than 1/30th of a second. The intuitive reason for this is that the largest video frame (and peak rate) is caused by transmission of a large I frame. Since I video frames are immediately followed by B video frames which tend to be much smaller, for a T greater than 1/30th of a second, r (which is a function of T) may be decreased allowing SG to accept more connections. The DMG is therefore at most 1 for delays less than 1/30th of a second and at 63 msec, where the RCSP/D-BIND DMG peaks at 2.79, the SG/(r,T) DMG is lagging behind at 1.72.

Thus, the RCSP scheduler with the D-BIND model is more efficient than the RCSP scheduler with the Xmin model since the former represents a tighter constraint on the source. However, both techniques result in higher network utilizations than is achievable for SG/(r,T). The reason for this is that the framing strategy of SG/(r,T) requires a busy period bound rather than the tighter backlog bound.

Figure 7(b) shows the same sequence of curves for the advertisements sequence. As expected, the trends are the same as for Figure 7(b) but the DMG is less. The reason for this is that the intense action and colors of the advertisement sequence results in a very bursty deterministic traffic specification (see [18] for further details). Thus, this source is more difficult to multiplex allowing approximately a 33% increase over a peak rate reservation scheme.

6 Implementation Issues

To implement Stop-and-Go, mechanisms are needed at both the link level and at the queue management level. At the link level, a framing structure is needed, and there is a synchronization requirement such that the framing structure is the same at both the sending and the receiving ends of the link. At the queue management level, two FIFO queues are



(a) Stop-and-Go (b) RCSP

Figure 8: Implementation of RCSP and Stop-and-Go

needed for each priority level, one storing the eligible packets ready to be transmitted, the other storing the packets that won't be eligible until the end of the current frame time. Mechanisms are needed to swap the two FIFO queues at the start of each frame time. Also, the set of FIFO queues with eligible packets need to be serviced according to a non-preemptive static priority policy. This is shown Figure 8 (a).

RCSP seems to be more complex than Stop-and-Go since it requires traffic regulation on a per connection basis. However, the conceptual decomposition of the rate controller into a set of regulators in RCSP does not imply that there must be multiple physical regulators in an implementation; a common mechanism can be shared by all logical regulators. Each regulator has two functions: computing the eligibility times for incoming packets on the corresponding connection, and holding packets until they become eligible. Eligibility times for packets from different connections are computed using the same formula with different parameters; holding packets is equivalent to managing a set of timers. One mechanism for managing timers is the calendar queue [3]. Figure 8(b) shows an example implementation of RCSP based on a modified version of a calendar queue. A calendar queue consists of a clock and a calendar, which is a pointer array indexed by time. Each entry in the calendar points to an array of linked lists indexed by priority levels. The clock ticks at fixed time intervals. Upon every tick of the clock, the linked lists in the array indexed by the current time are appended at the end of the scheduler's linked lists. Packets from the linked list of one priority level in the rate-controller are appended to the linked list of the same priority level in the scheduler. The scheduler just selects the first packet at the highest priority queue that is non-empty. As can be seen, the data structures used in the proposed implementation are simple: arrays and linked lists. The operations are all constant-time ones: array indexing, insertion at the tail of a linked list, deletion from the head of a linked list. Another implementation based on two-dimensional shifters is proposed in [22].

Both Stop-and-Go and RCSP need a mechanism to service packets according to a non-preemptive static priority policy, which is easy to implement in high speed switches. In fact, even early implementations of commercial ATM switches have at least two priority levels [2]. Experimental ATM switches have more priority levels. The queue module in the Xunet-2 [8] switch supports 16 priority levels [14].

One important difference between Stop-and-Go and RCSP is that RCSP requires the computation of the eligibility time to be performed on a per packet basis while Stop-and-Go needs only per frame processing. Regardless, rate control in RCSP may still be feasible. For example, a 1 Gbps link sending out 53 byte cells must process cells at the rate of approximately 2.4 million cells per second. A 50 MIPS processor is thus allowed 20 instructions per cell, which is more than enough to compute the eligibility time of a cell. To perform delay-jitter control, RCSP also requires time-stamping on a per packet basis. In an ATM network where each packet is only 53 bytes, the time-stamping of each packet may be too expensive. To eliminate this overhead, rate-jitter control regulators rather than delay-jitter control regulators can be used in the RCSP server. Rather than calculating eligibility time of a packet based on the eligibility time of the same packet in the previous server as done in delay-jitter controlling regulators, rate-jitter controlling regulators calculate the eligibility time of a packet based on the eligibility times of packets arriving earlier at the server on the same connection [34], therefore, time-stamping packets at each switch are not needed. Examples of rate-jitter controlling regulators are the popular leaky bucket [28] and the dual leaky bucket mechanism.

7 Related Work

There have been a number of new service disciplines proposed to support Quality of Service in the context of high speed networks. These service disciplines may be classified as either work-conserving or non-work-conserving.

Jitter-EDD [30], which uses the two-component structure with delay-jitter-control regulators and the Earliest-Due-Date scheduler, is one of the first non-work-conserving disciplines proposed. It has many of the desirable properties possessed by Stop-and-Go and RCSP. For example, Jitter-EDD provides per-connection end-to-end delay and delay-jitter bounds and allows the buffer space requirements to be uniformly distributed across the network. However, it is unclear how to implement an EDD scheduler efficiently at high speeds. In [29], another rate-controlled discipline is proposed with a First-Come-First-Served scheduler. Though much simpler than EDD, a FCFS scheduler provides only one delay bound and thus cannot efficiently support the diverse QOS requirement in the future integrated services networks. Hierarchical Round Robin [13] is a service discipline that also uses a multi-level framing strategy. It differs from Stop-and-Go in that it does not use synchronized framing structure across links, thus it cannot provide tight end-to-end delay-jitter bounds. Besides exact service disciplines, a number of analytical models have also been proposed to study non-work-conserving policies, for example, the affine server in [21] and the AIRPORT server in [5].

Among the work-conserving disciplines proposed are: Virtual Clock [37], variations of the Earliest-Due-Date algorithms [7, 15], and Generalized Processor Sharing [24]. All of them use a sorted priority queue mechanism [36], which makes it difficult to implement in high-speed switches. Also, in a network with work-conserving disciplines, more resources such as buffer space are needed in downstream switches due to traffic pattern distortions inside the network. As well, more buffer space is needed at the destination to provide an isochronous service [26, 31].

8 Conclusion

This paper compares two packet service disciplines proposed to support guaranteed performance service in a connection-oriented packet-switching network: Rate-Controlled Static Priority (RCSP) and Stop-and-Go. There are many similarities between these two disciplines: both disciplines maintain certain traffic characteristics throughout the network; both disciplines employ multiple priority levels to allocate multiple local delay bounds to different connections; both disciplines, when used with their corresponding admission control algorithms, can provide end-to-end delay and delay-jitter guarantees in networks of arbitrary topology; and both disciplines require less buffer space both inside the network and at the destination node compared to work-conserving disciplines. The main difference between the two disciplines is that Stop-and-Go uses a single strategy (framing) to allocate delay bounds and bandwidth, while RCSP decouples the server into two components, a rate-controller and a scheduler. This decoupling in RCSP allows more flexible allocation of bandwidth and delay bounds, which results in higher link utilizations. Two MPEG video traces were used in the paper to quantitatively compare the efficiency of Stop-and-Go and RCSP in terms of link utilization. The analysis showed that because Stop-and-Go's framing strategy requires the busy period to be less than a frame time, the RCSP server can multiplex more connections for a given delay bound.

References

[1] A. Banerjea and S. Keshav. Queueing delays in rate controlled networks. In *Proceedings of IEEE INFOCOM'93*, pages 547–556, San Francisco, CA, April 1993.

[2] E. Biagioni, E. Cooper, and R. Sansom. Designing a practical ATM LAN. *IEEE Network Magazine*, pages 32–39, March 1993.

[3] R. Brown. Calendar queues: A fast O(1) priority queue implementation for the simulation event set problem. *Communications of the ACM*, 31(10):1220–1227, October 1988.

[4] R. Cruz. A calculus for network delay, part I : Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–121, January 1991.

[5] R. Cruz. Service burstiness and dynamic burstiness measures: A framework. *Journal of High Speed Networks*, 1(2):105–127, 1992.

[6] D. Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, November 1990.

[7] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[8] A. G. Fraser, C. R. Kalmanek, A. E. Kaplan, W. T. Marshall, and R. C. Restrick. Xunet2: A nationwide testbed in high-speed networking. In *Proceedings of IEEE INFOCOM'92*, pages 582–589, Firenze, Italy, May 1992.

[9] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[10] S. Golestani. Congestion-free transmission of real-time traffic in packet networks. In *Proceedings of IEEE INFOCOM'90*, pages 527–542, San Francisco, CA, June 1990.

[11] S. Golestani. A stop-and-go queueing framework for congestion management. In *Proceedings of ACM SIGCOMM'90*, pages 8–18, Philadelphia PA, September 1990.

[12] S. Golestani. Duration-limited statistical multiplexing of delay-sensitive traffic in packet networks. In *Proceedings of IEEE INFOCOM'91*, April 1991.

[13] C. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks. In *Proceedings of IEEE GLOBECOM'90*, pages 12–20, San Diego, CA, December 1990.

[14] C. Kalmanek, S. Morgan, and R. C. Restrick. A high performance queueing engine for ATM networks. In *Proceedings of 14th International Switching Symposium*, Yokahama, Japan, October 1992.

[15] D. Kandlur, K. Shin, and D. Ferrari. Real-time communication in multi-hop networks. In *Proceedings of 11th International Conference on Distributed Computer Systems*, May 1991.

[16] L. Kleinrock. *Queueing Systems*. John Wiley and Sons, 1975.

[17] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs for providing deterministic guarantees to VBR video traffic. In *Proceedings of ACM SIGMETRICS'95*, Ottowa, Ontario, May 1995.

[18] E. Knightly and H. Zhang. Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models. In *Proceedings of IEEE INFOCOM'95*, pages 1137–1145, Boston, MA, April 1995.

[19] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proceedings of ACM SIGMETRICS'92*, pages 128–139, Newport, Rhode Island, June 1992.

[20] J. Kurose. Open issues and challenges in providing quality of service guarantees in high-speed networks. *ACM Computer Communication Review*, 23(1):6–15, January 1993.

[21] S. Low. *Traffic Control in ATM Networks*. PhD dissertation, University of California at Berkeley, May 1992.

[22] M. Maresca, June 1993. Personal communication.

[23] R. Nagarajan, J. Kurose, and D. Towsley. Local allocation of end-to-end quality-of-service in high-speed networks. In *IFIP TC6 Task Group/WG6.4 International Workshop on Performance of Communication Systems*, pages 99–118, Martinique, January 1993.

[24] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control - the single node case. In *Proceedings of IEEE INFOCOM'92*, pages 521–530, Firenze, Italy, May 1992.

[25] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. In *Proceedings of IEEE INFOCOM'93*, pages 521–530, San Francisco, CA, March 1993.

[26] C. Partridge. Isochronous applications do not require jitter-controlled networks, September 1991. RFC 1157.

[27] J. Stankovic and K. Ramamritham. *Hard Real-Time Systems*. IEEE Computer Society Press, 1988.

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

[29] D. Verma. *Guaranteed Performance Communication in High Speed Networks*. PhD dissertation, University of CA at Berkeley, November 1991.

[30] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of Tricomm'91*, pages 35–46, Chapel Hill, North Carolina, April 1991.

[31] H. Zhang. Service disciplines for integrated services packet-switching networks. Ph.D. Dissertation. UCB/CSD-94-788, U. C. Berkeley, November 1993.

[32] H. Zhang and D. Ferrari. Rate-controlled static priority queueing. In *Proceedings of IEEE INFOCOM'93*, pages 227–236, San Francisco, CA, March 1993.

[33] H. Zhang and D. Ferrari. Improving utilization for deterministic service in multimedia communication. In *Proceedings of 1994 International Conference on Multimedia Computing and Systems,* pages 295–304, Boston, MA, May 1994.

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

[35] H. Zhang and E. Knightly. Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models. In *Proceedings of ACM SIGMETRICS'94*, pages 211–220, Nashville, TN, May 1994.

[36] H. Zhang and K. Srinivasan. Comparison of rate-based service disciplines. In *Proceedings of ACM SIGCOMM'91*, pages 113–122, Zurich, Switzerland, September 1991.

[37] L. Zhang. Virtual clock: A new traffic control algorithm for packet switching networks. In *Proceedings of ACM SIGCOMM'90*, pages 19–29, Philadelphia, PA, September 1990.