To appear in Journal of High Speed Networking

Rate-Controlled Service Disciplines*

Hui Zhang
Lawrence Berkeley Laboratory
1 Cyclotron Road, MS: 50B-2239
Berkeley, CA 94720

Domenico Ferrari
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720

Abstract

We propose a class of non-work-conserving service disciplines, called the Rate-Controlled Service Disciplines. When coupled with suitable admission control algorithms, Rate-Controlled Service Disciplines can provide end-to-end deterministic and statistical performance guarantees on a per-connection basis in an arbitrary topology packet-switching network. The key feature of a rate-controlled service discipline is the separation of the server into two components: a rate-controller and a scheduler. This separation makes it possible to obtain end-to-end performance characteristics by applying single node analysis at each switch. It also has several other distinct advantages: it decouples the allocation of bandwidths and delay bounds, uniformly distributes the allocation of buffer space inside the network to prevent packet loss, and allows arbitrary combinations of rate-control policies and packet scheduling policies. Rate-controlled service disciplines provide a general framework within which most of the existing non-work-conserving disciplines can be naturally expressed. One discipline in this class, called Rate-Controlled Static Priority (RCSP), is particularly suitable for providing performance guarantees in high speed networks. It achieves simplicity of implementation as well as flexibility in the allocation of bandwidths and delay bounds to different connections.

1 Introduction

Future high-speed networks will have to support real-time communication services, which allow clients to transport information with performance guarantees expressed in terms of delay, delay jitter, throughput and loss rate bounds. 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 [11]. However, in a packet-switching network, packets from different connections will interact with each other at each switch; 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 in the context of integrated-services networks. Analyses of hard real-time systems usually assume a *single server* environment, *periodic* jobs, and the job delay bounded by its *period* [24]. However, the 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 often intractable for realistic traffic models. Also, classical queueing analyses usually study *average* performance for *aggregate* traffic [16, 27], while in integrated-services networks *performance bounds* need to be derived on a *per-connection* basis [8, 19]. In addition to the challenge of providing end-to-end per-connection performance guarantees to heterogeneous, bursty traffic, service disciplines must be *simple* enough that they can be implemented at very high speeds.

In this paper, we present a new class of service policies called *rate-controlled service disciplines.* This class of service disciplines is *non-work-conserving*, i.e., a server may be idle even when there are packets to be transmitted. Although non-work-conserving disciplines were seldom studied in the past, our research shows that non-work-conserving rate-controlled service disciplines have several advantages that make them suitable for supporting guaranteed performance communication in a high speed networking environment. In particular, rate-controlled service disciplines,

*This research was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Digital Equipment Corporation, Hitachi, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing official policies, either expressed or implied, of the U.S. Government or any of the sponsoring organizations.

when coupled with suitable admission control algorithms, can provide end-to-end per-connection deterministic and statistical performance guarantees in arbitrary topology simple networking and internetworking environments. The key feature of a rate-controlled service discipline is the separation of the server into two components: a rate-controller and a scheduler. This separation has several distinct advantages: it decouples the allocation of bandwidths and delay bounds, uniformly distributes the allocation of buffer space inside the network to prevent packet loss, and allows arbitrary combinations of rate-control policies and packet scheduling policies.

In Section 2, we discuss the guaranteed performance service model and review some of the issues that arise when designing service disciplines to provide performance guarantees. In Section 3, we describe the proposed rate-controlled service disciplines and derive the conditions to bound the end-to-end delay for a connection that traverses a network with rate-controlled servers. In Section 4, we propose an implementation for one discipline in this class, called Rate-Controlled Static Priority (RCSP). We believe that the implementation is simple enough to be able to run at very high speed. In Section 5, we show that most of the proposed non-work-conserving disciplines can be implemented by rate-controlled service disciplines with the appropriate choices of rate controllers and schedulers. Finally, we summarize the results in Section 6.

2 Background

2.1 Service Model

We assume that there are two types of services offered by an integrated services network: guaranteed service and non-guaranteed service. The guaranteed service provided by an integrated-services network should have the following properties [10]:

(1) *The service interface is general.* Instead of supporting only a fixed class of applications, the network should provide a parameterized interface abstraction that allows the client to specify a continuum of values and an unrestricted combination of values for its specification of both traffic characteristics and performance requirements. One example of such a general service interface [8], which we will use in this paper, consists of the following traffic parameters: $Xmin$, minimum packet inter-arrival time, $Xave$, minimum average packet inter-arrival time, I, averaging interval over which $Xave$ is computed, and $Smax$, maximum packet size; and the following end-to-end performance parameters: D, delay bound, Z, deadline violation probability bound, W, end-to-end buffer overflow probability bound, and J, end-to-end delay jitter bound.

(2) *The solution is applicable to a wide variety of internetworking environments.* Most end-to-end communication will go through several networks. Guaranteed-performance communication services that could not be easily implemented in a wide spectrum of internetworks would have limited value.

(3) *The guarantees are quantitative, and mathematically provable.* It is our belief that the clients of a guaranteed performance service require predictable performance. Computer networks are *strongly coupled non-linear* systems with complex interactions between different entities. Although simulation can give insights into the performance of a network, results from simulations of smaller networks are not easily extended to larger networks due to the non-linear nature of the network. In particular, it has been observed that adding a single extra node may make a stable system unstable [12]. These considerations make the problem more challenging and favor solutions where the results are guaranteed in general environments. As long as the guarantees are *a priori*, they can either be deterministic or statistical [8]. To achieve this, we believe that the network should manage resources explicitly; this entails (a) connection admission control with resource reservation; and (b) connection-oriented communication with pre-computed routes.

2.2 Service Disciplines

We consider the paradigm proposed in [11] for providing guaranteed service to clients in a packet switching network: before communication starts, the client specifies its traffic characteristics and performance requirements to the network; the client's traffic and performance parameters are translated into local parameters, and a set of connection admission control conditions are tested at each switch; the new connection is accepted only if its admission would not cause the performance guarantees made to other connections to be violated; during data transfers, each switch will service packets from different connections according to a 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: connection admission control at the connection level, and service discipline at the packet level. We believe that a complete solution needs to specify both the service discipline and the associated connection admission control conditions. Some other researchers have proposed solutions for just one of these two problems.

Recently, several new service disciplines, which aim to provide different qualities of service to different connections, have been proposed. Among them are Delay Earliest-Due-Date (Delay-EDD) [11, 15, 34], Virtual Clock [33], Fair Queueing [7] and its weighted version, also known as Generalized Processor Sharing [22], Jitter Earliest-Due-Date (Jitter-EDD) [26], Stop-and-Go [13], and Hierarchical Round Robin (HRR) [14]. They are closely related, but also have some important differences [31]. Figure 1 shows a taxonomy that classifies the existing solutions along two dimensions: (1) how the service discipline allocates, explicitly or implicitly, different delay bounds and bandwidths to different connections in a single switch; (2) how the service discipline handles traffic distortions in a network of switches. The first issue relates to the design of a single server: how to allocate delay bound and bandwidth among

		Delay/Bandwidth Allocation	
Interaction of Multiple Servers		**Sorted Priority Queue**	**Multi-level Framing**
	Control Distortion	Jitter-EDD	HRR Stop-and-Go
	Accommodate Distortion	Delay-EDD Virtual Clock Fair Queueing General Processor Sharing	

Figure 1: Taxonomy of service disciplines

different connections. The objective of the allocation of delay bound and bandwidth is that, with a certain scheduling discipline, a connection can be guaranteed to receive a certain throughput, and each packet on that connection can be guaranteed to have a bounded delay. There are two approaches to allocating delay/bandwidth to different connections in a single switch: the sorted priority queue mechanism and the framing strategy. As discussed in [31], a multi-level framing strategy introduces dependencies between delay and bandwidth allocation. Such a solution cannot support low delay/low bandwidth applications efficiently. A sorted priority queue avoids this coupling, but has a higher degree of complexity. The insertion operation in a sorted priority queue has an $O(\log N)$ complexity [17], which makes it difficult to implement the algorithm at very high speeds[1].

The second issue concerns the interaction between different switches along a path. A switch can provide local performance guarantees to a connection only when the traffic on that connection is well-behaved. 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 local performance bounds can be guaranteed for a connection only if the connection's input traffic to 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 [4, 1, 23, 18].

In general, characterizing traffic inside the network is difficult. In networks with *work-conserving* service disciplines, even in the situations when traffic inside the network can be characterized, the worst-case traffic is usually more bursty inside the network than that at the entrance. This 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 $\Delta\sigma_j^{i-1} = \sum_{i'=1}^{i-1} \rho_j \overline{d}_{i',j}$ and $\overline{d}_{i',j}$ is the local delay for the connection at the $i'-th$ switch. Compared to the characterization of the source traffic, the maximum burst size at switch i increases by $\sum_{i'=1}^{i-1} \rho_j \overline{d}_{i',j}$. This maximum burst size grows monotonically along the path of the connection.

[1] In [3], it was reported that a high-speed sequencer chip has been implemented to support service disciplines such as Virtual Clock; however, the chip can only sort a maximum of 256 packets [3]. It has yet to be seen whether a high-speed implementation can be built to support a sorted priority queue mechanism in an environment with large numbers of connections and packets.

In [18], a family of stochastic random variables is used to characterize a source. Connection j is said to satisfy a characterization $\{(R_{t_1,j}, t_1), (R_{t_2,j}, t_2), (R_{t_3,j}, t_3)...\}$, where the $R_{t_i,j}$ are random variables, and $t_1 < t_2 < \cdots$ are time intervals, if $R_{t_i,j}$ 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 $\{(R_{t_1,j}, t_1), (R_{t_2,j}, t_2), (R_{t_3,j}, t_3)...\}$ at the entrance to the network, its characterization will be $\{(R_{t_1+\Delta t_j^{i-1},j}, t_1), (R_{t_2+\Delta t_j^{i-1},j}, t_2), (R_{t_3+\Delta t_j^{i-1},j}, t_3), ...\}$ at the $i'-th$ switch, where $\Delta t_j^{i-1} = \sum_{i'=1}^{i-1} b_{i'}$ and $b_{i'}$ is the maximum busy period at switch i'. 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 switch j. I.e., the traffic is burstier at switch j than at the entrance.

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

Another approach to deal with the problem of traffic pattern distortions is to control the distortions to traffic patterns at each switch. In Jitter-EDD, Stop-and-Go, and HRR, traffic patterns are partially or completely reconstructed at each switch so as to offset the effects of network load fluctuations and of the interactions between switches. Reconstructing the traffic pattern at each node requires *non-work-conserving disciplines*. With a non-work-conserving discipline, the server may be idle even when there are packets waiting to be sent. Non-work-conserving disciplines were seldom studied in the past mainly due to two reasons. First, in most of previous performance analyses, the major performance indices are the *average* low delay of all packets and the *average* high throughput of the server. With a non-work-conserving discipline, a packet may be held in the server even when the server is idle. This may increase the average delay of packets and decrease the average throughput of the server. Secondly, most previous studies assumed a single server environment. The potential advantages of non-work-conserving disciplines in a network of servers were therefore not realized. In integrated-services networks, end-to-end delay *bounds* are to be guaranteed on a per-connection basis in a *network* of servers. Since guaranteeing end-to-end delay bounds requires *worst case* analysis in a networking environment, the above reasons for not using non-work-conserving disciplines are not significant any more. In this paper, we will show that a class of non-work-conserving disciplines called rate-controlled service disciplines have some distinct advantages that make them suitable for providing performance guarantees in packet-switching integrated-services networks.

3 Rate-Controlled Service Disciplines

As discussed previously, a service discipline for integrated-services packet-switching networks should address two issues: (1) how to allocate different delay bounds and bandwidths to different connections, and (2) how to deal with traffic distortions within the network.

In this section, we present a new class of service disciplines, called *rate-controlled service disciplines*, which address the problem by separating the server into two components: a rate controller and a scheduler. The rate controller monitors the traffic on each connection and forces the traffic to obey the desired traffic pattern. The scheduler orders transmissions of packets on different connections. Thus, the rate controller allocates bandwidth and controls traffic distortion, whereas the scheduler allocates service priorities to packets and controls the delay bounds of connections.

The rest of this section is organized as follows: in Section 3.1, we present the rate-controlled service discipline by describing the two components of a rate-controlled server: the rate controller and the scheduler; in Section 3.2, we give a theorem that states the end-to-end delay characteristics of connections in a network with rate-controlled servers, assuming that local delay bounds can be guaranteed in the scheduler of each of the rate-controlled servers; we present the proof of the theorem in Section 3.4 after we prove an important lemma that establishes the delay characteristics in the two-node case in Section 3.3; in Section 3.5, we give the conditions for providing local delay bounds in a Static Priority (SP) scheduler.

3.1 Rate-Controlled Service Disciplines

Table 1 shows the notation used in this paper. In discussions when there is no ambiguity, some subscripts or superscripts are omitted for simplicity.

As shown in Figure 2 (a), a rate-controlled server has two components: a rate controller and a scheduler. The rate controller shapes the input traffic on each connection into the desired traffic pattern by assigning an eligibility time to each packet; the scheduler orders the transmission of eligible real-time packets from all the connections. In the following discussion, we assume that real-time packets have non-preemptive priority over non-real-time packets. The scheduling policies discussed below only apply to the real-time packets. The service order of non-real-time packets is not specified, and is irrelevant in this discussion.

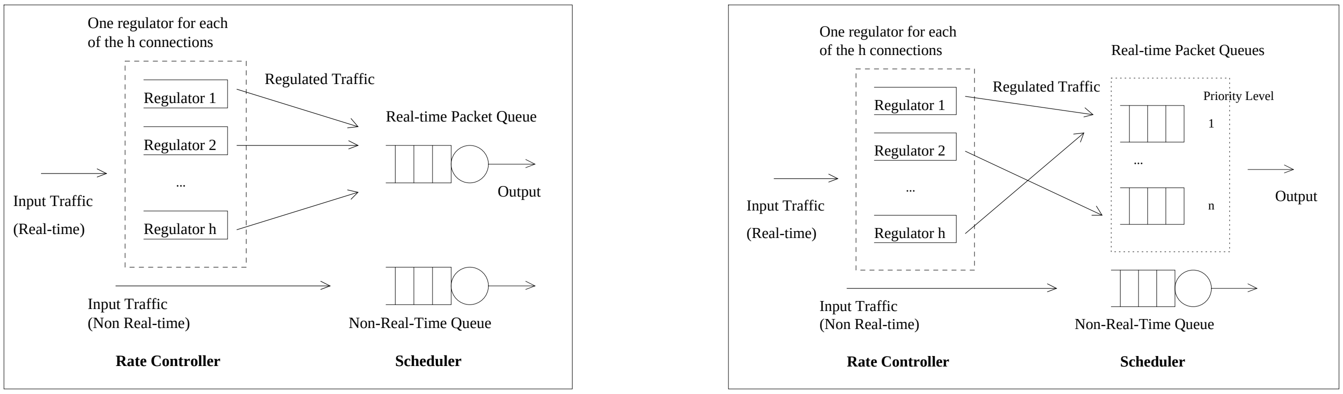
a bar is used to denote an upper bound	j denotes the connection number
d denotes the local delay	i denotes the switch number
D denotes the end-to-end delay	k denotes the packet number

$d_{i,j}^k$	delay of the k^{th} packet on connection j at the i^{th} switch along its path
$\overline{d}_{i,j}$	local delay bound for connection j at the i^{th} switch along its path
$\overline{D}_j$:	end-to-end delay bound for connection j
$\pi_{i,j}^k$:	link delay between the $(i-1)^{th}$ and the i^{th} switch for the k^{th} packet on connection j
$\overline{\pi}_{i,j}$:	maximum link delay between the $(i-1)^{th}$ and the i^{th} switch for all packets on connection j

Table 1: Notation used in this paper

Conceptually, a rate controller consists of a set of regulators corresponding to each of the connections traversing the switch; each regulator is responsible for shaping the traffic of the corresponding connection into the desired traffic pattern. Regulators control the interactions between switches and eliminate jitter. Many types of regulators are possible; they will be discussed in Section 5. In this section, we present two types of regulators: **(1)** *rate-jitter controlling regulator*, or RJ regulator, which controls rate jitter by partially reconstructing the traffic pattern; and **(2)** *delay-jitter controlling regulator*, or DJ regulator, which controls delay jitter by fully reconstructing the traffic pattern.

The regulator achieves this control by assigning each packet an eligibility time upon its arrival and holding the packet till 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.



(a) (b)

Figure 2: Rate-Controlled Service Discipline

For a $(Xmin, Xave, I)$ RJ regulator, where $Xmin \le Xave < I$ holds, the eligibility time of the k^{th} packet on a connection at the i^{th} switch along its path, ET_i^k, is defined with reference to the eligibility times of packets arriving earlier at the switch on the same connection:

$$ET_i^k = -I, \quad k < 0 \tag{1}$$

$$ET_i^1 = AT_i^1 \tag{2}$$

$$ET_i^k = max(ET_i^{k-1} + Xmin,\ ET_i^{k-\lfloor \frac{I}{Xave} \rfloor + 1} + I,\ AT_i^k), \quad k > 1 \tag{3}$$

where AT_i^k is the time the k^{th} packet on the connection arrived at switch i. (1) is defined for convenience so that (3) holds for any $k > 1$.

If we consider the sequence of packet eligibility times at switch i, $\{ET_i^k\}_{k=1,2,...}$, it always satisfies the $(Xmin, Xave, I)$ traffic characterization.

Proposition 1 *Assume a connection traverses a $(Xmin, Xave, I)$ RJ regulator, and the maximum packet size of the connection is bounded by $Smax$. The output traffic of the regulator satisfies the $(Xmin, Xave, I, Smax)$ specification.*

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 $\overline{d}_{i-1}$ 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:

$$ET_0^k = AT_0^k \tag{4}$$

$$ET_i^k = ET_{i-1}^k + \overline{d}_{i-1} + \overline{\pi}_i, \quad i > 0 \tag{5}$$

where switch 0 is the source of the connection.

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

$$ET_i^{k+1} - ET_i^k = AT_0^{k+1} - AT_0^k \quad \forall k, i \geq 0 \tag{6}$$

i.e., the traffic pattern on a connection at the output of the regulator of every switch traversed by the connection is exactly the same as the traffic pattern of the connection at the *entrance* to the network.

Proposition 2 *Assume a connection traverses a set of switches with rate-controlled servers having DJ regulators. If its traffic obeys the specification Θ at the entrance to the network, the output traffic from the connection's regulator, or the input traffic into the connection's scheduler at each switch, will also obey Θ.*

This proposition is more general than Proposition 1. It applies to any traffic specification Θ, rather than just $(Xmin, Xave, I, Smax)$. For example, Θ can be the (σ, ρ) model as proposed in [4], or even stochastic models like Markov-Modulated Poisson Process (MMPP) models [21]. As discussed in [28, 32], this property of completely reconstructing the traffic pattern allows us to extend local statistical performance bounds to end-to-end statistical performance bounds.

From Propositions 1 and 2, we can see that both types of regulators enforce the traffic specification requirement for each connection, so that the traffic going into the scheduler will always satisfy the traffic specification.

Any scheduler can be used in a rate-controlled server as long as it can provide local delay bounds for connections under certain admission control conditions. Many schedulers, including the most common First Come First Served (FCFS) discipline, can be used. A rate-controlled server with an SP scheduler is called a Rate-Controlled Static Priority (RCSP) server. This is shown in Figure 2 (b). We will give the admission control conditions for the Static Priority (SP) scheduler in Section 3.5.

3.2 End-To-End Delay Characteristics

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. In this section, we will show 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 in the rate controllers will *not* increase the *end-to-end delay bound* of the connection. Formally, we have the following theorem (End-to-End Theorem):

Theorem 1 *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 between the $i-1^{th}$ and the i^{th} switch. Assume that the scheduler of switch i can guarantee that the delays of all the packets on the connection be bounded by $\overline{d}_i$ as long as the connection's input traffic to the scheduler satisfies the given $(Xmin, Xave, I, Smax)$ specification. If the traffic on the connection obeys the $(Xmin, Xave, I, Smax)$ specification at the entrance to the first switch,*

1. *the end-to-end delay for any packet on the connection is bounded by $\sum_{i=1}^{n} \overline{d}_i + \sum_{i=2}^{n} \overline{\pi}_i$ if rate-jitter controlling regulators are used;*
2. *the end-to-end delay and the delay jitter for any packet are bounded by $\sum_{i=1}^{n} \overline{d}_i + \sum_{i=2}^{n} \overline{\pi}_i$ and $\overline{d}_{i_n}$, respectively, if delay-jitter controlling regulators are used;*
3. *reservation of the following amount of buffer space for the connection at switch i will prevent packet loss: $(\lceil \frac{\overline{d}_{i-1} + \overline{\pi}_i - \hat{\pi}_i}{Xmin} \rceil + \lceil \frac{\overline{d}_i}{Xmin} \rceil) Smax, \quad i = 1, \ldots, n;\ \overline{d}_0 = 0$*

Before we give the proof, we first briefly discuss the implications of the result. At first glance, the result seems trivial, because it states that an end-to-end delay bound can be guaranteed if local delay bounds can be guaranteed at each scheduler and link. However, several things should be noticed:

- If a switch's server does not include a rate controller, but has only the scheduler, the result will not hold. Under the assumptions of the theorem, a local delay bound can be guaranteed for a connection only if the connection's *input traffic to the scheduler* satisfies the $(Xmin, Xave, I, Smax)$ specification. Even though the traffic satisfies that specification at the entrance to the network, without rate controllers the traffic pattern will be distorted inside the network, and the $(Xmin, Xave, I, Smax)$ specification may *not* be satisfied in subsequent schedulers.
- The theorem just assumes that there is an upper bound on delays in the scheduler; the holding times in the rate controllers have not be accounted for. The proof of the theorem needs to establish that holding in the rate controllers will not increase the end-to-end delay bound.
- The theorem holds for *any* schedulers that can guarantee local delay bounds for connections satisfying certain traffic specifications. This has two implications: 1) any scheduler that can guarantee delay bounds to connections at a single switch can be used in conjunction with a rate controller to form a rate-controlled server, which can then be used to provide end-to-end performance bounds in a networking environment; 2) rate-controlled servers with different schedulers can be used in a network, and end-to-end performance bounds can still be guaranteed. This is particularly important in an internetworking environment, where heterogeneity is unavoidable.
- The theorem assumes links with bounded, but possibly *variable*, delay. This is important for an internetworking environment, in which links connecting switches may be subnetworks such as ATM or FDDI networks. It is possible to bound delay over these subnetworks; however, the delays for different packets will be *variable*. In contrast, some of the existing solutions proposed to bound end-to-end delay assume a network model with *constant* delay links [5, 6, 18, 13, 1].
- The theorem provides both end-to-end delay bounds and non-trivial end-to-end delay jitter bounds. With a bounded-delay-jitter network service, clients need to reserve much less buffer space to obtain an end-to-end isochronous service [28].
- The theorem also gives the buffer space requirement to prevent packet loss for each connection. Unlike work-conserving disciplines, where more buffer space is needed at downstream switches to accommodate potentially burstier traffic, rate-controlled service disciplines needs uniformly distributed buffer space inside the network to prevent packet loss. This saves the overall buffer space requirement inside the network when a connection traverses more than two hops.

3.3 Two-Node Lemma

In the following lemma, we establish the delay characteristics of two switches connected in cascade, when each type of regulator is used.

Lemma 1 *Consider a connection with traffic specification $(Xmin, Xave, I, Smax)$ traversing two switches using rate-controlling service disciplines, and labeled $i-1$ and i, respectively. For the k^{th} packet on the connection, let d_{i-1}^k be its delay in the scheduler of switch $i-1$, π_i^k its link delay from the $(i-1)^{th}$ to the i^{th} switch, and h_i^k its holding time in the regulator of switch i. If $d_{i-1}^k \le \overline{d}_{i-1}$ and $\pi_i^k \le \overline{\pi}_i$ for all k's, we have that*

1. *if a delay-jitter controlling regulator is used at switch i,*

$$d_{i-1}^k + h_i^k + \pi_i^k = \overline{d}_{i-1} + \overline{\pi}_i \tag{7}$$

2. *if a rate-jitter controlling regulator is used at switch i,*

$$d_{i-1}^k + h_i^k + \pi_i^k \le \overline{d}_{i-1} + \overline{\pi}_i \tag{8}$$

Proof.

Let ET_{i-1}^k and ET_i^k be the eligibility times for the k^{th} packet at switch $i-1$ and i, respectively. DT_{i-1}^k is the departure time of the k^{th} packet from switch $i-1$, and AT_i^k is the arrival time of the k^{th} packet at switch i. We have $d_{i-1}^k = DT_{i-1}^k - ET_{i-1}^k$, $h_i^k = ET_i^k - AT_i^k$, and $\pi_i^k = AT_i^k - DT_{i-1}^k$, Combining the three equations, we have

$$d_{i-1}^k + h_i^k + \pi_i^k = ET_i^k - ET_{i-1}^k \tag{9}$$

1. For the case of a delay-jitter controlling regulator, from (9) and $ET_i^k - ET_{i-1}^k = \overline{d}_{i-1} + \overline{\pi}_i$, which is (5), we immediately have

$$d_{i-1}^k + h_i^k + \pi_i^k = \overline{d}_{i-1} + \overline{\pi}_i$$

2. For the case of a rate-jitter controlling regulator, we will prove the lemma by induction with respect to k.
Step 1. When k=1, from (2), we have $ET_i^1 = AT_i^1$. It follows that $h_i^1 = ET_i^1 - AT_i^1 = 0$. Also, since $d_{i-1}^1 \le \overline{d}_{i-1}$ and $\pi_i^1 \le \overline{\pi}_i$ hold, we have

$$d_{i-1}^1 + h_i^1 + \pi_i^1 \le \overline{d}_{i-1} + \overline{\pi}_i \tag{10}$$

So, (8) holds for k=1.
Step 2. Assume that (8) holds for the first k packets; we now consider the $(k+1)^{st}$ packet. From (3) we have

$$ET_i^{k+1} = max(ET_i^k + Xmin, ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I, AT_i^{k+1}), \quad k > 1 \tag{11}$$

Since ET_i^{k+1} is the maximum of the three quantities at the right hand side of (11), we consider each of the following three cases in turn: **(a)** $ET_i^{k+1} = AT_i^{k+1}$, **(b)** $ET_i^{k+1} = ET_i^k + Xmin$, and **(c)** $ET_i^{k+1} = ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I$.
Case **(a):** If $ET_i^{k+1} = AT_i^{k+1}$, we have $h_i^{k+1} = ET_i^{k+1} - AT_i^{k+1} = 0$. Also from $d_{i-1}^{k+1} \le \overline{d}_{i-1}$ and $\pi_i^{k+1} \le \overline{\pi}_i$, it immediately follows that

$$d_{i-1}^{k+1} + h_i^{k+1} + \pi_i^{k+1} \le \overline{d}_{i-1} + \overline{\pi}_i \tag{12}$$

Case **(b):** If $ET_i^{k+1} = ET_i^k + Xmin$, we have

$$\begin{aligned}
d_{i-1}^{k+1} + h_i^{k+1} + \pi_i^{k+1} &= ET_i^{k+1} - ET_{i-1}^{k+1} && (13)\\
&= (ET_i^k + Xmin) - ET_{i-1}^{k+1} && (14)\\
&\le (ET_i^k + Xmin) - (ET_{i-1}^k + Xmin) && (15)\\
&= ET_i^k - ET_{i-1}^k && (16)\\
&\le \overline{d}_{i-1} + \overline{\pi}_i && (17)
\end{aligned}$$

(15) holds due to $ET_{i-1}^{k+1} \ge ET_{i-1}^k + Xmin$. (17) is derived from the assumption that (8) holds for the first k packets.
Case **(c):** If $ET_i^{k+1} = ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I$, we have

$$\begin{aligned}
d_{i-1}^{k+1} + h_i^{k+1} + \pi_i^{k+1} &= ET_i^{k+1} - ET_{i-1}^{k+1} && (18)\\
&= (ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I) - ET_{i-1}^{k+1} && (19)\\
&\le (ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I) - (ET_{i-1}^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I) && (20)\\
&= ET_i^{k-\lfloor\frac{I}{Xave}\rfloor+1} - ET_{i-1}^{k-\lfloor\frac{I}{Xave}\rfloor+1} && (21)\\
&\le \overline{d}_{i-1} + \overline{\pi}_i && (22)
\end{aligned}$$

(20) holds due to the fact that the traffic pattern into the scheduler of switch $i-1$ also satisfies the $(Xmin, Xave, I, Smax)$ specification, hence $ET_{i-1}^{k+2} \ge ET_{i-1}^{k-\lfloor\frac{I}{Xave}\rfloor+2} + I$. (22) is derived from the assumption that (8) holds for the previous k packets.

From (a), (b) and (c), we have proven that (8) holds for the $k+1^{st}$ packet if it holds for the first k packets. Thus, (8) holds for any packet on the connection. **Q.E.D.**

The Two-Node Lemma is an important step in establishing Theorem 1. Intuitively, a packet is held in a regulator only when the packet was transmitted ahead of schedule by the previous switch, or when the packet experienced less delay over the link than the maximum link delay. The amount of holding time in the regulator is never greater than the amount of time the packet is ahead of schedule plus the difference between the maximum link delay and the actual link delay. Thus, holding does not increase the *accumulative delay bound*. The result is obvious for delay-jitter controlling regulators, since the eligibility time of a packet at a switch is defined with respect to its eligibility time in the previous switch. In a sense, Equations (4) and (5), which are the definition of the eligibility time of a packet for delay-jitter controlling regulators, have already had the result built in. However, for a rate-jitter controlling regulator, the result is non-trivial, since the definition of the eligibility time of a packet is based on the packet spacing requirement in the same switch.

The idea of holding a packet by the amount of time it is ahead of schedule in the previous switch was first proposed in [26]. The relationship between the regulators defined in [26] and the DJ & RJ regulators is discussed in Section 5.

3.4 Proof of the End-To-End Theorem

With Lemma 1 proven, we are now ready to prove Theorem 1.

Proof of Theorem 1.

For the first two parts of the theorem, consider the end-to-end delay of the k^{th} packet on the connection, $D^k = \sum_{i=1}^{n}(h_i^k + d_i^k) + \sum_{i=0}^{n-1}\overline{\pi}_i$. Rearranging the terms, we have $D^k = h_1^k + \sum_{i=2}^{n}(d_{i-1}^k + h_i^k + \overline{\pi}_i) + w_n^k$. If the traffic obeys the $(Xmin, Xave, I, Smax)$ characterization at the entrance to the first switch, there is no holding time in the first regulator, or $h_1^k = 0$. The D_k can be further simplified to be:

$$D^k = \sum_{i=2}^{n-1}(d_{i-1}^k + h_i^k + \overline{\pi}_i) + d_n^k \tag{23}$$

(1) If rate-jitter controlling regulators are used, according to Proposition 1 the traffic obeys the $(Xmin, Xave, I, Smax)$ characterization at the entrance to each of the schedulers. From Lemma 1, we have

$$d_{i-1}^k + h_i^k + \pi_i^k \le \overline{d}_{i-1} + \overline{\pi}_i \tag{24}$$

Combining (23) and (24), we have,

$$D^k = \sum_{i=2}^{n}(d_{i-1}^k + h_i^k + \overline{\pi}_i) + w_n^k \le \sum_{i=2}^{n}(\overline{d}_{i-1} + \overline{\pi}_i) + \overline{d}_n = \sum_{i=1}^{n}\overline{d}_i + \sum_{i=2}^{n}\overline{\pi}_i \tag{25}$$

(2) If delay-jitter controlling regulators are used, according to Proposition 2 the traffic obeys the $(Xmin, Xave, I, Smax)$ characterization at the entrance to each of the schedulers. From Lemma 1, we have

$$d_{i-1}^k + h_i^k + \pi_i^k = \overline{d}_{i-1} + \overline{\pi}_i \tag{26}$$

Combining (23) and (26), we have $D^k = \sum_{i=2}^{n}(\overline{d}_{i-1} + \overline{\pi}_i) + w_n^k$. Since $0 < w_n^k \le \overline{d}_n$, we have

$$\sum_{i=1}^{n-1}\overline{d}_i + \sum_{i=2}^{n}\overline{\pi}_i < D^k \le \sum_{i=1}^{n}\overline{d}_i + \sum_{i=2}^{n}\overline{\pi}_i \tag{27}$$

(3) To verify the third part of the theorem, notice that the longest times a packet can stay in the regulator and the scheduler of the i^{th} switch are $\overline{d}_{i-1} + \overline{\pi}_i - \hat{\pi}_i$ and $\overline{d}_i$, respectively; since the minimum packet inter-arrival time is $Xmin$, it follows that the maximum numbers of packets in the regulator and the scheduler are $\lceil \frac{\overline{d}_{i-1} + \overline{\pi}_i - \hat{\pi}_i}{Xmin} \rceil$ and $\lceil \frac{\overline{d}_i}{Xmin} \rceil$, respectively. **Q.E.D.**

3.5 Bounding delay for a single scheduler

In the previous section, we established the end-to-end delay characteristics of a connection traversing switches with rate-controlled service disciplines, assuming that local delay bounds can be guaranteed in the scheduler of each of the rate-controlled servers. In this section, we give admission control condition that guarantee deterministic delay bounds for a Static Priority (SP) scheduler. Conditions can also be given for other schedulers such as the First-Come-First-Served (FCFS) scheduler [4, 9, 28] and the Earliest-Due-Date-First (EDD) scheduler [11].

An SP scheduler has a number of prioritized real-time packet queues and a non real-time packet queue. Packets at priority level 1 have the highest priority. A connection is assigned to a particular priority level at the connection's establishment time; all the packets from the connection will be inserted into the real-time packet queue at that priority level. Multiple connections can be assigned to the same priority level. The scheduler services packets using a non-preemptive static-priority discipline, which 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 order is not specified. Comparing with the FCFS discipline, which is the simplest but can have only one delay bound in one server, and the EDD discipline, which can allocate a continuous spectrum of delay bounds to different connections but may be too complicated to implement at very high speeds, the SP discipline provides a good balance between simplicity of implementation and flexibility in allocating delay bounds. The combination of a rate controller and an SP scheduler is called a Rate-Controlled Static Priority (RCSP) server [29]. This is shown in Figure 2 (b).

By limiting the number of connections at each priority level via certain admission control conditions, the waiting time of each packet at a priority level can be bounded. The following theorem proved in [29] gives the admission control condition for an SP scheduler.

Theorem 2 *Let $\overline{d^1}, \overline{d^2}, \ldots, \overline{d^n}$ ($\overline{d^1} < \overline{d^2} < \cdots < \overline{d^n}$) be the delay bounds associated with each of the n priority levels, respectively, in a Static Priority scheduler. Assume that C_q is the set of connections at level q, and, that the j^{th} connection in C_q has the traffic specification ($Xmin_j^q, Xave_j^q, I_j^q, Smax_j^q$). Also assume that the link speed is l, and the size of the largest packet that can be transmitted onto the link is $\overline{Smax}$. If*

$$\sum_{q=1}^{m} \sum_{j \in C_q} \lceil \frac{\overline{d^m}}{Xmin_j^k} \rceil Smax_j^k + \overline{Smax} \leq \overline{d^m} l \tag{28}$$

the waiting time of a real-time packet at level m is bounded by $\overline{d^m}$.

Although condition (28) can guarantee delay bounds in an SP scheduler, since the delay bound calculation is solely based on the $Xmin$ of each connection instead of the $Xave$, it has the limitation that the sum of the peak rates of all real-time connections cannot exceed the link speed. This will result in a low average utilization of the connection when the peak-to-average-rate ratio is high [28, 30]. A more refined admission control condition that overcomes such a limitation is given in Theorem 3.

Theorem 3 *Assume an SP scheduler has n priority levels. Let C_q be the set of the connections at level q, and the j^{th} connection in C_q satisfy the traffic specification ($Xmin_j^q, Xave_j^q, I_j^q, Smax_j^q$). Also assume that the maximum link speed is l, and the maximum size of a packet that can be transmitted over the link is $\overline{Smax}$. If $\sum_{q=1}^{n} \mu_{ave}^q \leq 1$ where*

$$\mu_{ave}^q = \sum_{j \in C_q} \frac{Smax_j^q}{Xave_j^q \times l} \tag{29}$$

the maximum delay of any packet at priority level m is bounded above by $\overline{d^m}$, where

$$\overline{d^m} = \begin{cases} min(\overline{d^{m'}}, \overline{d^{m''}}) & \sum_{q=1}^{m-1} \mu_{peak}^q < 1 \\ \overline{d^{m''}} & \sum_{q=1}^{m-1} \mu_{peak}^q \geq 1 \end{cases} \tag{30}$$

where

$$\mu_{peak}^q = \sum_{j \in C_q} \frac{Smax_j^q}{Xmin_j^q \times l} \tag{31}$$

$$\overline{d^{m'}} = \frac{\frac{\overline{Smax}}{l} + \sum_{q=1}^{m} \sum_{j \in C_q} \frac{Smax_j^q}{l}}{1 - \sum_{q=1}^{m-1} \mu_{peak}^q} \tag{32}$$

$$\overline{d^{m''}} = \frac{\frac{\overline{Smax}}{l} + \frac{1}{l} \sum_{q=1}^{m} \sum_{j \in C_q} \frac{Smax_j^q}{Xave_j^q} [I_j^q (1 - \frac{Xmin_j^q}{Xave_j^q}) + Xmin_j^q]}{1 - \sum_{q=1}^{m-1} \mu_{ave}^q} \tag{33}$$

The theorem can be proven by applying the bounding techniques developed by Cruz [4]. The details of the proof are given in [28].

4 Implementation

In this section, we present one implementation of the RCSP queueing discipline. We believe that this implementation is simple enough to run at very high speeds.

We have shown that a RCSP server has two components, a scheduler and a rate controller. The scheduler consists of multiple prioritized FCFS queues, and the rate controller consists of a set of regulators corresponding to each connection. Notice that the conceptual decomposition of the rate controller into a set of regulators does not

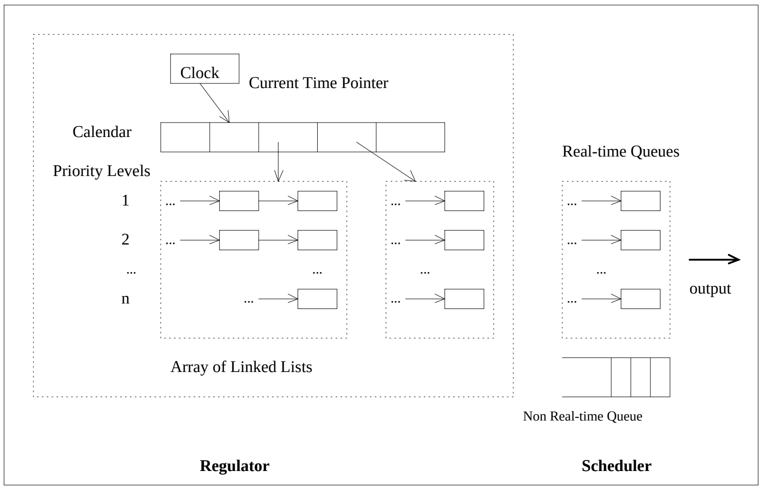


Figure 3: Implementation of RCSP

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 till they become eligible. Eligibility times for packets from different connections are computed using the same formula (as described in Section 3.2) with different parameters; holding packets is equivalent to managing a set of timers; the mechanism for managing timers, which is a calendar queue [2, 25], can be shared by all regulators.

Figure 3 shows the proposed implementation. Each of the real-time queues is implemented as a linked list. The rate controller is implemented using 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 to 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.

Upon the arrival of each packet, the eligibility time of the packet, ET, is calculated; if $\lfloor \frac{ET}{Tick} \rfloor$ is equal to the current clock time, where $Tick$ is the clock tick interval, the packet is appended at the end of the corresponding real-time queue of the scheduler; otherwise, the packet is appended to the corresponding linked list at the calendar queue entry indexed by $\lfloor \frac{ET}{Tick} \rfloor$.

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: insertion at the tail of a linked list and deletion from the head of a linked list. We believe that it is feasible to implement this in a very high speed switch.

5 Generalization and Comparison

In Section 3.1, we presented two types of regulators: the rate-jitter controlling, or RJ, regulator, and the delay-jitter controlling, or DJ, regulator. In Section 3.5, we gave conditions that guarantee delay bounds for SP schedulers. There are other rate controllers and schedulers that can be used in rate-controlled service disciplines. In this section, we show that the definition of rate-controlled service disciplines is quite general. By having different combinations of rate controllers and schedulers, we can have a wide class of service disciplines. Most of the proposed non-work-conserving disciplines, including Jitter-EDD [26], Stop-and-Go [13] and Hierarchical Round Robin [14], either belong to this class, or can be implemented by a rate-controlled service discipline with the appropriate choices of rate controllers and schedulers. In the following, we first show the corresponding rate controllers and schedulers for each of the above three disciplines, and then compare these disciplines with RCSP by showing the tradeoffs of using different rate controllers and schedulers.

5.1 Jitter-EDD, Stop-and-Go, HRR

Jitter-EDD is a rate-controlled service discipline. The rate controller in Jitter-EDD consists of regulators that control delay jitter in a network with constant link delays. We will call this type of regulator the DJ_e regulator. In a DJ_e regulator, the eligibility time for packet k at the i^{th} switch along the path is defined as follows:

$$ET_i^k = AT_i^k + Ahead_{i-1}^k \tag{34}$$

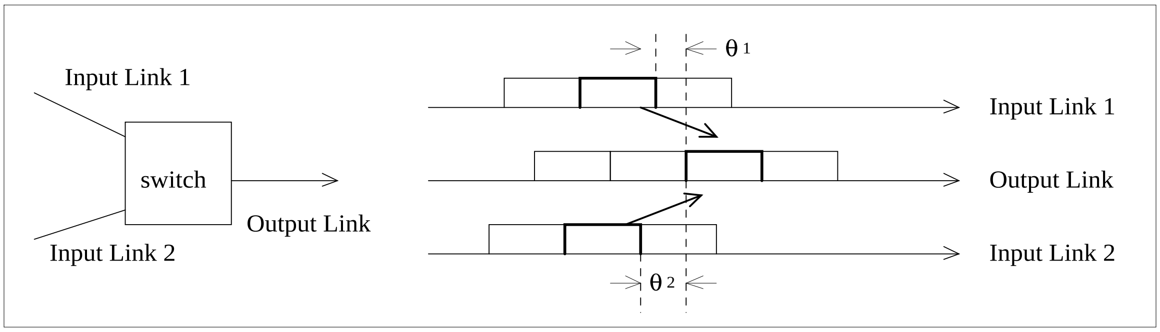


Figure 4: Synchronization between input and output links in Stop-and-Go

where $Ahead_{i-1}^k$ is the amount of time the packet is ahead of schedule in switch $i-1$, the immediate upstream switch to switch i. The scheduler in Jitter-EDD is a variation of the Earliest-Due-Date scheduler [20].

It is easy to show that, in a DJ regulator, the eligibility time for packet k at the i^{th} switch along the path can also be represented as:

$$ET_i^k = AT_i^k + Ahead_{i-1}^k + (\overline{\pi}_i - \pi_i^k) \tag{35}$$

Compared with (34), (35) has one more term: $\overline{\pi}_i - \pi_i^k$, which can be seen as the amount of time the packet is ahead of schedule in the link. In a network with constant delay links, the value of this term will always be zero. A DJ regulator differs from a DJ_e regulator in that the DJ regulator not only removes the traffic distortion introduced by the previous scheduler, but also removes the traffic distortion introduced by a variable-delay link.

A Stop-and-Go server with n frame sizes ($T_1 < T_2 < ... < T_n$) 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:

$$ET_i^k = AT_i^k + Ahead_{i-1}^k + \theta \tag{36}$$

where $Ahead_{i-1}^k$ is the amount of time the packet is ahead of schedule in switch $i-1$, 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. Figure 4 illustrates this synchronization time. In a 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, the allocation of delay bounds and bandwidth is more restrictive in Stop-and-Go than in RCSP. First, the traffic has to be specified with respect to the frame size that corresponds to the priority level the connection is assigned to. This introduces a coupling between the allocations of bandwidth and delay bounds. Secondly, there are dependencies among the local delay bounds at each priority level. $T_m = h_{m,m'} \times T_{m'}$ must hold for each pair of priority levels, with $1 \leq m' \leq m \leq n$, and $h_{m,m'}$ being an integer. Thirdly, the delay bound allocations for each connection in different switches are coupled with one another. In [13], a connection has to have the same frame size in all the switches. In [31], 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.

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

$$ET_i^k = max(AT_i^k + \tau, ET_i^{k-a_i} + T_m) \tag{37}$$

where $AT_i^k + \tau$ is the beginning time of the next frame and a_i is the service quantum of the connection in switch i, which is defined to be the maximum number of packets that can be served on the connection within each frame of size T_m. In the static priority scheduler, the delay bound associated with level m is T_m, $1 \leq m \leq n$. If a connection traverses a level-m RJ_h regulator, it is assigned to the priority level m in the scheduler. This shows the coupling between delay and bandwidth allocation in HRR. In contrast, in a RCSP server, a connection can be assigned to any priority level regardless of its rate parameters.

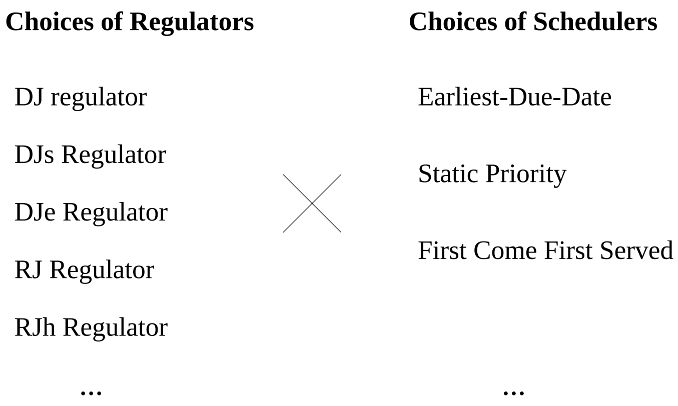


Figure 5: Generality of Rate-Controlled Service Disciplines

5.2 Generalization

The class of rate-controlled service disciplines is quite general. As has been shown in the previous section, most of the non-work-conserving disciplines that have been proposed in the literature either belong to this class or can be implemented with a rate-controlled service discipline by choosing an appropriate rate controller and an appropriate scheduler. Different combinations of regulators and schedulers will result in more service disciplines. This is shown in Figure 5.

5.3 Comparison With Work-Conserving Disciplines

As discussed in Section 2.2, many work-conserving policies have also been proposed in the literature to support guaranteed performance services. Some representatives are Virtual Clock [33], variations of the Earliest-Due-Date algorithms [11, 15, 34], and Generalized Processor Sharing [22]. Virtual Clock has been introduced to provide average throughput guarantees. No admission control algorithms have been proposed for Virtual Clock to provide end-to-end delay guarantees. It has been shown in [23] that, with GPS servers, end-to-end deterministic delay bounds can be guaranteed on a per-connection basis in an arbitrary topology network. Also, the end-to-end delay bound obtained is tighter than the simple addition of all the worst-case local delay bounds at each switch, as has been done in this paper. However, it is unclear how statistical guarantees can be provided in a network of GPS servers. In contrast, end-to-end statistical guarantees can be provided in a network of delay-jitter controlled servers by applying single node analysis at each of the switches [32], because the exact traffic pattern is maintained throughout the network.

All of the above work-conserving disciplines use a sorted priority queue mechanism. It is unclear whether this can be implemented in high-speed switches. Also, due to traffic pattern distortions inside the network, more buffer space is needed in downstream switches in a network with work-conserving disciplines.

6 Summary

In this paper, we have presented a class of non-work-conserving disciplines called rate-controlled service disciplines. A rate-controlled service discipline has two components: a rate controller and a scheduler. After the rate controller limits the distortion of the traffic introduced by load fluctuations inside the network, the scheduler orders the packets for transmission. The end-to-end delay of a packet in a network with rate-controlled servers consists of the following components: waiting times in the schedulers, holding times in the rate controllers and the link delays.

We have showed that the end-to-end delays of all the packets on a connection can be bounded, as long as the delays on links and the delays at each of the schedulers can be bounded. Thus, although holding times may increase the average delay of packets on a connection, they do not increase the end-to-end delay bounds. By keeping traffic characteristics throughout the network, end-to-end performance characteristics can be obtained by simply applying single node analysis at each switch.

The key feature of a rate-controlled service discipline is the separation of the server into two components: a rate controller and a scheduler. Such a separation has several advantages that make rate-controlled service disciplines suitable for supporting guaranteed-performance communication in a high-speed networking environment:

- End-to-end performance bounds can be obtained in a network of arbitrary topology. Also, the result applies not only in simple networks where switches are connected by physical links, but also in internetworks where switches are connected by subnetworks. The delays of packets traversing subnetworks must be *bounded*, but may be *variable*.

- By having a server with two components, we can extend results previously obtained for a single scheduler to a networking environment. Any scheduler that can provide delay bounds to connections at a single switch can be used in conjunction with a rate controller to form a rate-controlled server, which can then be used to provide end-to-end performance bounds in a networking environment.

- Given this separation of functionality into rate control and packet scheduling, we can have arbitrary combinations of rate-control policies and packet scheduling disciplines.

- Separation of rate-control and delay-control functions in the design of a server allows decoupling of bandwidth and delay bound allocation to different connections. Most existing solutions have the drawback of coupling the bandwidth/delay bound allocation — allocating of a lower delay bound to a connection automatically allocates a higher bandwidth to the connection. Such solutions cannot efficiently support low delay/low bandwidth connections.

- Unlike work-conserving disciplines which require the reservation of more buffer space at the downstream switches traversed by a connection, rate-controlled disciplines also have the advantage of requiring evenly distributed buffer space at each switch to prevent packet loss.

We have shown that rate-controlled service disciplines provide a general framework within which most of the existing non-work-conserving disciplines such as Jitter-EDD [26], Stop-and-Go [13] and Hierarchical Round Robin [14] can be naturally expressed. One discipline in this class, called Rate-Controlled Static Priority (RCSP), is particularly suitable for providing performance guarantees in high-speed networks. It achieves both flexibility in the allocation of bandwidth and delay bounds to different connections, as well as simplicity of implementation.

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] 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.

[3] H. J. Chao. Architecture design for regulating and scheduling user's traffic in ATM networks. In *Proceedings of ACM SIGCOMM'92*, pages 77–87, Baltimore, Maryland, August 1992.

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

[5] R.L. Cruz. A calculus for network delay, part II : Network analysis. *IEEE Transaction of Information Theory*, 37(1):121–141, 1991.

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

[7] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Journal of Internetworking Research and Experience*, pages 3–26, October 1990. Also in Proceedings of ACM SIGCOMM'89, pp 3-12.

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

[9] D. Ferrari. Real-time communication in an internetwork. *Journal of High Speed Networks*, 1(1):79–103, 1992.

[10] D. Ferrari, A. Banerjea, and H. Zhang. Network support for multimedia: a discussion of the Tenet approach. Technical Report TR-92-072, International Computer Science Institute, Berkeley, California, October 1992. Also to appear in *Computer Networks and ISDN Systems*.

[11] 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.

[12] S. Floyd and V. Jacobson. The synchronization of periodic routing messages. In *Proceedings of ACM SIGCOMM'93*, pages 33–44, San Francisco, CA, September 1993.

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

[14] C.R. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks. In *IEEE Global Telecommunications Conference*, pages 300.3.1 – 300.3.9, San Diego, California, December 1990.

[15] D.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] D.E. Knuth. *The Art of Computer Programming. Volume 3: Sorting and searching.* Addison-Wesley, 1975.

[18] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *ACM SigMetrics'92*, 1992.

[19] 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.

[20] C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of ACM*, 20(1):46–61, January 1973.

[21] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transaction on Communication*, 36(7):834–844, July 1988.

[22] A.K.J. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control - the single node case. In *Proceedings of the INFOCOM'92*, 1992.

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

[24] John Stankovic and Krithi Ramamritham. *Hard Real-Time Systems.* IEEE Computer Society Press, 1988.

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

[26] 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.

[27] R. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.

[28] H. Zhang. Service disciplines for integrated services packet-switching networks. PhD Dissertation. UCB/CSD-94-788, University of California at Berkeley, November 1993.

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

[30] H. Zhang and D. Ferrari. Improving utilization for deterministic service in multimedia communication. In *1994 International Conference on Multimedia Computing and Systems*, Boston, MA, May 1994.

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

[32] H. Zhang and E. Knightly. Providing end-to-end statistical performance guarantees with interval dependent stochastic models. In *ACM Sigmetrics'94*, Nashville, TN, May 1994.

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

[34] Q. Zheng and K. Shin. On the ability of establishing real-time channels in point-to-point packet-switching networks, March 1994. to appear in *IEEE Transactions on Communications*.