Exact Admission Control for Networks with a Bounded Delay Service

Jörg Liebeherr (Member), Dallas E. Wrege (Student Member), and Domenico Ferrari (Fellow)

Abstract— **To support the requirements for the transmission of continuous media, such as audio and video, multiservice packet-switching networks must provide service guarantees to connections, including guarantees on throughput, network delays, and network delay variations. For the most demanding applications, the network must offer a service which provides deterministically bounded delay guarantees, referred to as 'bounded delay service'. The admission control functions in a network with a bounded delay service require 'schedulability conditions' that detect violations of delay guarantees in a network switch. In this paper, exact schedulability conditions are presented for three packet scheduling methods: Earliest-Deadline-First (EDF), Static-Priority (SP), and a novel scheduling method, referred to as Rotating-Priority-Queues (RPQ). By characterizing the worst-case traffic with general subadditive functions, the presented schedulability conditions can be applied to a large class of traffic models. Examples, which include actual MPEG video traces, are presented to demonstrate the trade-offs involved in selecting a packet scheduling method for a bounded delay service.**

1 Introduction

A major challenge in the design of multiservice networks with quality-of-service is the implementation of a *bounded delay service*, that is, a communication service with deterministically bounded delays for all packets from a single connection [6]. A rigorous approach to a bounded delay service must consider all delay types that a packet may incur, including fixed processing and propagation delays as well as variable statistical multiplexing delays at network switches. Since fixed delays result from physical or technological constraints, the design of a bounded delay service focuses on finding appropriate packet scheduling techniques which determine the variable delays at the network switches.

In the presence of admission control and policing, which limit the number of connections and the traffic on the connections, a large number of packet scheduling techniques can provide bounds on delays [1, 5, 14]. However, many packet schedulers will result in an inefficient use of network resources. The performance of a packet scheduling method in providing bounded delay services can be determined by the degree to which it satisfies the following requirements [18, 25]:

- *Efficiency:* To efficiently utilize network resources, the packet switches should be able to support bounded delays for a large number of connections.
- *Flexibility:* A packet switch must be sufficiently flexible to satisfy a diverse set of delay requirements.
- *Complexity:* Since packet scheduling must be performed at the speed of the transmission links, the complexity of the scheduling operations must be kept small. If the time to schedule a packet exceeds its transmission time, the network links will be left idle most of the time.
- *Analyzability:* The admission control functions, which determine whether the network can accept a new connection without causing delay bound violations, require that analytical *schedulability conditions* be available for the packet schedulers. The schedulability conditions verify that the maximum delay of a packet does not exceed the delay bound of its connection. If exact schedulability conditions are not available, the admission control functions will unnecessarily limit the number of connections in the network.

Note that a single packet scheduling method cannot simultaneously optimize all of the above criteria. In particular, high efficiency and low complexity appear to be contradictory design goals. Recently many new scheduling methods were proposed for use in networks with a bounded delay service [1, 8, 11, 12, 15, 18, 26, 27, 28, 29]. Each of these scheduling methods presents a particular tradeoff in satisfying the above requirements.

In this paper, we investigate the trade-offs involved in scheduling for three different scheduling methods. We consider two traditional packet scheduling methods, Earliest-Deadline-First (EDF) and Static-Priority (SP), both of which have been considered for implementations of bounded delay services [6, 25]. The third scheduling method, called Rotating-Priority-Queues (RPQ), is a novel scheduling method for bounded delay services. For each scheduling method we present and prove exact schedulability conditions. With the exact conditions we can precisely evaluate the efficiency of the scheduling methods. By characterizing the worst-case traffic of a network connection in terms of general subadditive functions, the derived schedulability conditions are applicable to most traffic models used in the literature, e.g., [3, 6, 8, 23, 13, 18].

An EDF scheduler, which always selects the packet with the shortest deadline for transmission, is an optimal scheduler for a bounded delay service in the sense that it can support the delay bounds for any set of connections that can be supported by some other packet scheduling method [17]. A disadvantage of EDF scheduling is that queued

J. Liebeherr and D. E. Wrege are with the Department of Computer Science, University of Virginia, Charlottesville, VA 22903. Their research is supported in part by the National Science Foundation under Grant No. NCR-9309224. Domenico Ferrari is with the Università Cattolica, Piacenza, Italy.

packets must be sorted according to their deadlines, requiring a search operation whenever a new packet arrives at the scheduler.

An SP scheduler supports a fixed number of priority levels for connections. It maintains one FIFO queue for each priority level, and it always selects for transmission the first packet from the nonempty FIFO queue with the highest priority. Due to the implementation with FIFO queues, the complexity of SP scheduling is low. However, as we will show in our numerical examples, the number of connections with delay bound constraints that can be supported with an SP scheduler is significantly less as compared to an EDF scheduler. In addition, since SP schedulers can enforce only one delay bound at each priority level, the flexibility in providing variable delay bounds is limited by the number of priority levels.

The new Rotating Priority Queues (RPQ) scheduler approximates EDF scheduling without requiring queued packets to be sorted. RPQ is implemented with a set of ordered FIFO queues, similar to SP. Different from SP, the order of the FIFO queues is modified (*"rotated"*) after fixed so-called *rotation intervals*. As a result, the priority level of each FIFO queue is increased at the end of each rotation interval. Since queue rotations can be implemented without actually moving any packets, the additional complexity of RPQ as compared to SP is low. The number of FIFO queues needed for RPQ is inversely proportional to the length of the rotation interval. We show that by decreasing the length of the *rotation intervals* and appropriately increasing the number of FIFO queues, RPQ schedulers approximate EDF schedulers arbitrarily closely. Our numerical examples indicate that even with relatively large rotation intervals, thus, a small number of FIFO queues, an RPQ scheduler can closely approximate the efficiency of an EDF scheduler.

The remainder of this paper is structured as follows. In Section 2 we discuss the components needed for a network with a bounded delay service. In Sections 3 and 4, respectively, we give the necessary and sufficient schedulability conditions for EDF and SP packet schedulers. In Section 5 we present the novel RPQ packet multiplexer and prove its necessary and sufficient schedulability conditions. In Section 6 we present an empirical evaluation of the EDF, SP, and RPQ scheduling methods. The conclusions of this paper are given in Section 7.

2 Components of a Network with Bounded Delay Services

We consider connection-oriented packet-switching networks where packets from a connection traverse the network on a fixed path of switches and links. For each outgoing link of a network switch there is a packet scheduler which selects the order of packet transmission. In such a network, the number of connections with a bounded delay service that can be supported is majorly determined by the *traffic characterization* used to describe the worst-case traffic of a connection, the *packet scheduling method* at the switches, and the *accuracy of the schedulability conditions* used for the admission control tests. In the following we present our assumptions on these components and introduce necessary notation and terminology.

2.1 Traffic Characterization

Let $\mathcal{N}$ denote the set of connections with traffic arrivals to a packet scheduler. Let $a_j(t)$ denote the traffic on a connection $j \in \mathcal{N}$ that arrives at the scheduler at time t. Traffic on connection j consists of packets with maximum transmission time s_j^{max} and minimum transmission time s_j^{min}. We use $A_j[t, t+\tau] = \int_t^{t+\tau} a_j(t)dt$ to denote the *traffic arrivals* from connection j in time interval[1] $[t, t+\tau]$. The maximum traffic arrival from connection $j \in \mathcal{N}$ to the packet scheduler is assumed to be bounded by a right-continuous subadditive *traffic constraint function* A_j^*, such that for all times $t > 0$ and for all $\tau \geq 0$ we have [2, 3]:

$$A_j[t, t+\tau] \leq A_j^*(\tau) \tag{1}$$

where $A_j^*(t) = 0$ for all $t < 0$ and $A_j^*(t) \geq 0$ for $t \geq 0$.

We assume that the network has two mechanisms to enforce that traffic on a connection j entering a scheduler conforms to the given traffic constraint function A_j^*. The first such mechanism is a *traffic policer* at the entrance of the network which rejects traffic from connection j if it does not comply to A_j^*. The other mechanism is a *rate controller* which temporarily buffers packets to ensure that traffic from connection j entering the scheduler queue conforms to A_j^*. [2]

Traffic constraint functions are derived from deterministic traffic models which (a) characterize the worst-case traffic from a connection by a small set of parameters, and (b) enable simple traffic policing and rate controlling mechanisms. For example, the (σ, ρ)-model [3] describes the worst-case traffic on a connection j by a burst parameter σ_j and a rate parameter ρ_j, and can be policed by a leaky bucket mechanism [22]. The traffic constraint function for the (σ, ρ)-model is

$$A_j^*(t) = \sigma_j + \rho_j t$$

As another example, in the (x_{min}, x_{ave}, I, s)-model [6], $x_{min,j}$ specifies the minimum time interval between any two packets from a connection, $x_{ave,j}$ denotes a lower bound on the average interarrival time of packets averaged over a time interval I_j, and s_j^{max} is the maximum packet transmission time. Here, the traffic constraint function for a connection j is given by:

$$A_j^*(t) = \left\lfloor \frac{t}{I_j} \right\rfloor \frac{I_j\, s_j^{max}}{x_{ave,j}} + \\ + \min\left\{ \left\lceil \left(\frac{t}{I_j} - \left\lfloor \frac{t}{I_j} \right\rfloor \right) \frac{I_j}{x_{min,j}} \right\rceil, \frac{I_j}{x_{ave,j}} \right\} s_j^{max}$$

[1] We use [a,b] to denote the set of all x with $a \leq x \leq b$, (a,b] to denote the set of all x with $a < x \leq b$, [a,b) to denote the set of all x with $a \leq x < b$, and (a,b) to denote the set of all x with $a < x < b$.

[2] In an alternative approach for implementing a bounded delay service, changes to the worst-case traffic arrivals due to statistical multiplexing at the switches are accounted for through modifications to the traffic constraint function [4, 24].

2.2 Packet Scheduling and Schedulability Conditions

The packet scheduler at an outgoing link of a switch selects packets for transmission according to a given scheduling discipline. For example, a FIFO scheduler transmits packets in the order of their arrival. We assume that packet transmissions cannot be preempted. As a result, the only time instants when the scheduler selects a packet for transmission are (a) upon completion of a packet transmission if additional packets are waiting for transmission, and (b) upon arrival of a packet from the rate-controller at an empty scheduler. Throughout the paper we assume that the transmission rate of all schedulers is equal to one.

We use $W(t)$ to denote the *workload* (or backlog) of traffic in the scheduler at time $t \geq 0$. $W(t)$ includes all queued packets and the packet that is in transmission at time t. By assuming $W(t) = 0$ if $t < 0$, the workload in the scheduler at time $t \geq 0$ due to a set of connections $\mathcal{N}$ with arrivals $\{A_j\}_{j \in \mathcal{N}}$ is given by [21]:

$$W(t) = \sup_{0 \leq u \leq t} \left\{ \sum_{j \in \mathcal{N}} A_j[u,t] - (t-u) \right\} \tag{2}$$

We denote by $W(t^-)$ the workload at time t excluding the arrivals at time t, that is, $W(t^-) = \lim_{h \to 0} W(t-h)$. A *busy period* of a packet scheduler is a time interval where the scheduler queue is nonempty. Thus, a time interval $[t_1, t_2]$ is a busy period if $W(t_1^-) = 0$, $W(t_2^-) = 0$, and $W(t) > 0$ for all $t_1 \leq t < t_2$. We say that a packet scheduler is *stable* if all its busy periods are finite. Stability of a packet scheduler implies that the delays in the scheduler queue are finite. The condition for stability of a work-conserving packet schedulers is given by [3]:

$$\lim_{t \to \infty} \frac{\sum_{j=1}^{N} A_j^*(t)}{t} < 1 \tag{3}$$

To perform admission control tests for a packet scheduler, one needs to know the conditions that must hold at the scheduler such that delay bound violations do not occur. These conditions are referred to as *schedulability conditions*. Let d_j be the delay bound of connection j, that is, the maximum tolerable delay at the scheduler, including queueing and transmission delay, for any packet from connection j. [3] Then, schedulability is formally defined as follows:

Definition *Given a packet scheduler with scheduling method Σ, and given a set $\mathcal{N}$ of connections where each connection $j \in \mathcal{N}$ is characterized by a tuple $\{A_j^*, d_j\}$. The set of connections is said to be $\underline{\Sigma-\text{schedulable}}$ if for all $t > 0$ and for all arrivals $\{A_j\}_{j \in \mathcal{N}}$ that satisfy equation (1) no packet exceeds its delay bound d_j.*
The set is said to be $\underline{\text{schedulable}}$ if it is $\Sigma-$schedulable for some scheduler Σ.

[3] For multi-hop routes an algorithm is needed to partition the end-to-end delay constraints of a connection into segments [6].

In this paper we restrict ourselves to the best possible, that is necessary and sufficient, schedulability conditions. We present schedulability conditions for the Earliest-Deadline-First scheduler in Section 3, the Static-Priority scheduler in Section 4, and the novel Rotating-Priority-Queues scheduler in Section 5.

3 Earliest-Deadline-First Packet Schedulers

An Earliest-Deadline-First (EDF) scheduler assigns each arriving packet a *deadline*, computed as the sum of the arrival time at the scheduler and the delay bound d_j. The EDF scheduling algorithm always selects the packet with the earliest deadline for transmission. Since the scheduler queue of an EDF scheduler must be sorted according to deadlines, each packet arrival involves a search operation to find the correct position of the newly arrived packet in the scheduler queue. Ferrari and Verma presented sufficient schedulability conditions for EDF scheduling for a bounded delay service in [6]. Using a traffic model with periodic traffic arrivals, Zheng and Shin [30] derived necessary and sufficient schedulability conditions. Georgiadis, Guerin, and Parekh [7] proved necessary and sufficient conditions for a traffic characterization that complies to the (σ, ρ)-traffic model. For the (σ, ρ)-traffic model and with the assumption that the maximum packet transmission time is identical for all connections, the authors of [7] proved that EDF scheduling is optimal with respect to schedulability.

Next we present the general necessary and sufficient condition for schedulability in an EDF scheduler. The condition holds for all subadditive traffic constraint functions that bound the traffic on a connection in the sense of equation (1). The schedulability conditions are given as follows:

Theorem 1 *A set $\mathcal{N}$ of connections that is given by $\{A_j^*, d_j\}_{j \in \mathcal{N}}$ and $d_i \leq d_j$ whenever $i < j$ is EDF-schedulable if and only if for all $t \geq d_1$:*

$$t \geq \sum_{j \in \mathcal{N}} A_j^*(t - d_j) + \max_{k, d_k > t} s_k^{max} \tag{4}$$

where $\max_{k, d_k > t} s_k^{max} \equiv 0$ *for* $t > \max_{k \in \mathcal{N}} d_k$.

Recall that s_j^{max} denotes the maximum packet transmission time of a packet from connection j. Informally, the condition states that a deadline violation occurs at time t if the maximum traffic arrivals with a deadline before or at time t, i.e., $\sum_{j \in \mathcal{N}} A_j^*(t - d_j)$, exceeds t, the time that the scheduler has available for the transmission of this traffic. The term $\max_{k, d_k > t} s_k^{max}$ accounts for the fact that packet transmissions cannot be preempted. In Appendix A we give a complete proof of Theorem 1 which is a generalization of the proofs in [30] and [7]. In fact, our proof applies the same arguments as used in the proof of [30]. The steps of the proof carry over readily by replacing the specific traffic constraint function used in [30] by a general subadditive constraint function A^*.

The proof of Theorem 1 in Appendix A can be used to show optimality of EDF schedulers, in the sense that any schedulable set of connections is EDF-schedulable. This is stated in the following corollary:

Corollary 1 *Any schedulable set $\mathcal{N}$ of connections is EDF-schedulable.*

Proof: Note that the proof of necessity of Theorem 1 in Appendix A is done without assuming a specific scheduling method. Thus, we can conclude that the schedulability conditions of Theorem 1 in equation (4) are necessary schedulability conditions for all packet schedulers. Since the conditions in equation (4) are also sufficient for EDF schedulers, the claim follows. □

Example: For some traffic models, it is possible to obtain a closed-form expression for the schedulability conditions in Theorem 1. Note that closed-form schedulability conditions are attractive due to their low computational overhead. We assume that connections are ordered so that $i < j$ whenever $d_i < d_j$. Then, if traffic characterizations comply to the (σ, ρ)-model, we can rewrite the condition in equation (4) as:

$$\begin{cases} t \geq \sum_{i=1}^{j} \sigma_i + \rho_i (t - d_i) + \max_{k>j} s_k^{max} & \text{for } d_j \leq t < d_{j+1}, 1 \leq j < |\mathcal{N}| \\ t \geq \sum_{i=1}^{|\mathcal{N}|} \sigma_i + \rho_i (t - d_i) & \text{for } t \geq d_{|\mathcal{N}|} \end{cases} \quad (5)$$

As long as the stability condition in equation (3) is satisfied, i.e., $\sum_{j=1}^{|\mathcal{N}|} \rho_j < 1$, we obtain:

$$d_j \geq \frac{\sigma_j + \sum_{i=1}^{j-1} (\sigma_i - \rho_i d_i) + \max_{k>j} s_k^{max}}{1 - \sum_{i=1}^{j-1} \rho_i} \qquad \text{for all } j \in \mathcal{N} \quad (6)$$

4 Static-Priority Packet Schedulers

A Static-Priority (SP) scheduler assigns each connection to one priority level and always selects the highest-priority packet with the earliest arrival time for transmission. SP is a relatively simple scheduling method since it can be implemented with a fixed number of FIFO queues, one for each priority level. For the (σ, ρ)-traffic model, Cruz [3] derived necessary and sufficient schedulability conditions for SP schedulers. In fact, the conditions given in [3] are necessary and sufficient for all concave traffic constraint functions. For the (non-concave) $(x_{min,j}, x_{ave,j}, I_j, s_j^{max})$-model, Zhang and Ferrari [26] and Zhang [25] derived several sufficient schedulability conditions.

We consider an SP scheduler with P priority levels. Each connection is assigned a priority p with $1 \leq p \leq P$, and packets arriving on a connection are inserted into a FIFO queue associated with the priority of the connection. At the beginning of a busy period, or after completing the transmission of a packet, the SP scheduler always selects the first packet in the nonempty FIFO queue with the highest priority for transmission. We use $\mathcal{C}_p$ to denote the set of connections with priority p, and a lower priority index indicates a higher priority. All connections in $\mathcal{C}_p$ have the same delay bound d_p, with $d_p < d_q$ for $p < q$. Thus, the priority of a connection is high if its delay bound is short. We use s_p^{max} to denote the maximum transmission time of packets from a priority-p connection, i.e., $s_p^{max} = \max_{j \in \mathcal{C}_p} s_j^{max}$, and $s_p^{min} \geq 0$ to denote the minimum packet transmission time for a priority-p connection.

With these definitions, we formulate the necessary and sufficient schedulability conditions for SP schedulers as follows.

Theorem 2 *A set $\mathcal{N}$ of connections that is given by $\{A_j^*, d_j\}_{j \in \mathcal{N}}$ is SP-schedulable if and only if for all priorities p and for all $t \geq 0$ there exists a τ with $\tau \leq d_p - s_p^{min}$ such that:* [4]

$$t + \tau \geq \sum_{j \in \mathcal{C}_p} A_j^*(t) + \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*((t+\tau)^-) - s_p^{min} + \max_{r>p} s_r^{max} \quad (7)$$

A complete proof of Theorem 2 is presented in Appendix B. To arrive at an informal intuitive interpretation of the schedulability conditions, let us view time t as the arrival time of a (tagged) priority-p packet with transmission time s_p^{min} and time $t + \tau$ as the time when the tagged packet is scheduled for transmission. Then, equation (7) gives the condition that must hold when the tagged packet is scheduled for transmission. If the condition is satisfied at or before time $t + d_p - s_p^{min}$, then the tagged packet will leave the scheduler without a deadline violation. The terms in equation (7) are interpreted as follows: $\sum_{j \in \mathcal{C}_p} A_j^*(t) - s_p^{min}$ is the maximum priority-p traffic that can arrive before or together with the tagged packet; $\sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*((t+\tau)^-)$ is the maximum high-priority traffic that is transmitted before the tagged packet is scheduled for transmission; $\max_{r>p} s_r^{max}$ reflects that a (low-priority) packet in transmission cannot be preempted.

We note that testing exact schedulability for SP schedulers requires significantly more effort than for EDF schedulers. First, condition (7) must be tested for each priority level. Second, for a fixed priority p and fixed value of t, condition (7) must possibly be tested for the entire range of values of τ.

An equivalent formulation of Theorem 2 can be given in terms of the maximum delay of a priority-p packet in the

[4] t^- denotes the time immediately prior to time t.

scheduler, denoted by D_p^{max}:

$$D_p^{max} = s_p^{min} + \max_{t\geq 0} \min \left\{ \tau \mid t+\tau \geq \sum_{j\in\mathcal{C}_p} A_j^*(t) + \right.$$
$$\left. + \sum_{q=1}^{p-1} \sum_{j\in\mathcal{C}_q} A_j^*((t+\tau)^-) - s_p^{min} + \max_{r>p} s_r^{max}, \tau \geq 0 \right\} \quad (8)$$

This notation is similar to the one used by Cruz in [3] for the proof of schedulability conditions that are necessary and sufficient for concave traffic constraint functions. By taking into account that packets have a minimum size and relaxing the assumption that the traffic constraint function is derived from the (σ,ρ)-model, then the schedulability conditions derived by Cruz in [3] can be expressed as follows:

$$D_p^{max} \leq s_p^{min} + \max_{t\geq 0} \max \left\{ \tau \mid t+\tau \leq \sum_{j\in\mathcal{C}_p} A_j^*(t) + \right.$$
$$\left. + \sum_{q=1}^{p-1} \sum_{j\in\mathcal{C}_q} A_j^*((t+\tau)^-) - s_p^{min} + \max_{r>p} s_r^{max}, \tau \geq 0 \right\} \quad (9)$$

The difference between (8) and (9) may appear subtle, and in fact, both expression are identical if traffic constraints functions are concave. However, the difference between (8) and (9) can be significant for traffic models with non-concave traffic constraint functions.

Example: For the (σ,ρ)-model, the conditions of Theorem 2 can be much simplified. Assuming that there is only one connection p in each priority set $\mathcal{C}_p$ and assuming no restriction on the minimum packet size, i.e., $s_j^{min} = 0$, we can rewrite the condition in (7) as:

$$t\left(1 - \sum_{q=1}^{p} \rho_q\right) + \tau\left(1 - \sum_{q=1}^{p-1} \rho_q\right) \geq \sum_{q=1}^{p} \sigma_q + \max_{r>p} s_r^{max} \quad \text{for all } p = 1,2,\ldots,P \quad (10)$$

Clearly, for fixed τ the condition is satisfied for all $t \geq 0$ if it is satisfied for $t = 0$. Thus, for $\sum_{q=1}^{P} \rho_q < 1$, the connections are SP-schedulable if d_p is set to:

$$d_p \geq \frac{\sum_{q=1}^{p} \sigma_q + \max_{r>p} s_r^{max}}{1 - \sum_{q=1}^{p-1} \rho_q} \qquad \text{for all } p = 1,2,\ldots,P \quad (11)$$

Since the (σ,ρ)-model has a concave traffic constraint function, equations (8) and (9) coincide and the above conditions are equivalent to the condition given in [3].

5 Rotating-Priority-Queues Schedulers

In this section we present an approximation of the EDF scheduling method, referred to as Rotating-Priority-Queues (RPQ) scheduler. The advantage of RPQ over EDF is that the transmission queue of an RPQ scheduler queue need not be sorted. Rather, similar to an SP scheduler, RPQ can be implemented with a fixed number of FIFO queues.

Approximations of EDF scheduling with a set of ordered FIFO queues have been considered before [16, 19], but not in the context of a bounded delay service. The Head-of-Line with Priority Jumps (HOL-PJ) scheduler proposed by Lim and Kobza [16] assigns each FIFO queue a range of laxity values, where the laxity of a packet is the remaining time until a deadline violation. Timers are used to detect when a packet violates the laxity range of its FIFO queue. If a violation occurs for a packet, it is moved to the FIFO queue with the correct laxity range. A disadvantage of this method is that it requires a separate timer for each FIFO queue to detect violations of the laxity range. In a different approach, presented by Peha [19], the movement of queued packets is avoided by periodically rearranging the order of the FIFO queues. However, the implementation suggested in [19] cannot guarantee the absence of deadline violations and therefore is not applicable to a bounded delay service. Finally, the *calendar queue* implementation of the *Hierarchical Round Robin* discipline proposed by Kalmanek, Kanakia, and Keshav [12], rearranges queues after fixed-time intervals, however, without trying to approximate Earliest-Deadline-First scheduling and without considering deadline constraints. The RPQ scheduler presented here is similar to Peha's approach [19] in that RPQ approximates EDF by reordering FIFO queues after fixed time intervals without moving queued packets. However, the RPQ scheduler can guarantee that no packet exceeds a given delay bound.

In Subsection 5.1 we give a description of the RPQ scheduling method. Then we discuss an example to illustrate the operations of RPQ in Subsection 5.2. Finally, in Subsection 5.3, we derive an expression for the workload in an RPQ scheduler that is served before an arbitrary packet and use this expression to develop the necessary and sufficient schedulability conditions.

5.1 The RPQ Scheduler

The connections with traffic to the RPQ scheduler are partitioned into P disjoint priority sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P$ and connections in the same set have identical delay bounds. An RPQ scheduler uses a system parameter $\Delta > 0$, referred to as the *rotation interval*. All delay bounds supported by the RPQ scheduler are multiples of the rotation interval, that is, $d_p = n_p\Delta$ for connections from priority set $\mathcal{C}_p$ where $n_p < n_q$ if $p < q$ and $n_1 > 0$.

The RPQ scheduler maintains $n_P + 1$ ordered FIFO queues, and each FIFO queue is tagged with an integer index n where $0 \leq n \leq n_P$. The tagging of the queues is modified at the end of each rotation interval Δ. We refer to the FIFO queue that is tagged with index n as the n-queue. If a packet from a priority-p connection j arrives to the scheduler, it is inserted into the current n_p-queue. Since $n_p > 0$ for all priorities, no packet arrival is inserted into the current 0-queue. The RPQ scheduler always selects a packet from the non-empty queue tagged with the lowest index. Hence, packets in the 0-queue have the highest priority.

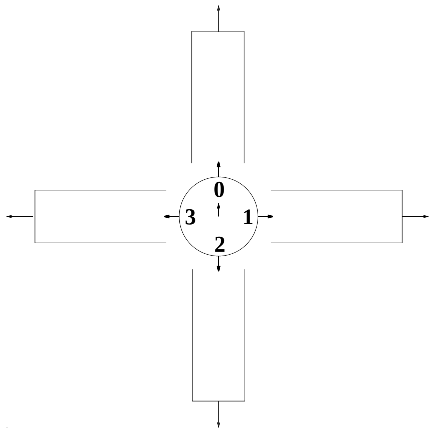
After every Δ time units, i.e., at the end of a rotation interval, the scheduler rearranges the taggings of the FIFO queues. For each $n \geq 1$, the current n-queue is relabeled as $(n-1)$-queue, and the current 0-queue becomes the new n_P-queue. Thus, the FIFO queues can be thought of as having performed a *"rotation"*. We assume that the queue rotation is performed independent of the presence of packets in the FIFO queues, that is, queues are rotated even if the RPQ scheduler is empty. We also assume that the queue rotation is performed instantaneously. If a packet arrival occurs at the time instant of a queue rotation, we assume that the queue rotation is performed before the packet arrives.

5.2 Illustration of RPQ Scheduling

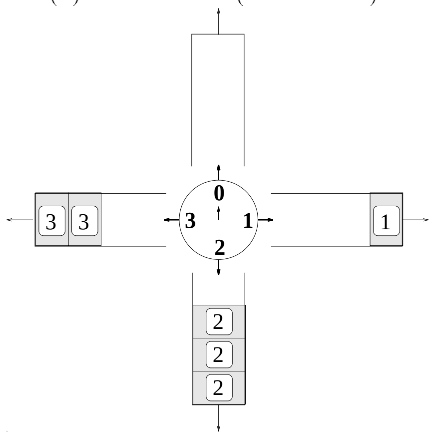
Next we illustrate the operations of the RPQ scheduler in a simple example with three priorities and delay bounds $d_p = p\Delta$ for $p = 1, 2, 3$. As shown in Figure 1, the RPQ scheduler for three priorities has four FIFO queues: one for each priority, and one for the current 0-queue. Figure 1(a) shows an empty scheduler at time 0^-. The tagging of FIFO queues is indicated by the labels in the circle shown in Figure 1(a). Here, the top queue is the current 0-queue, and proceeding clockwise, the other queues are tagged as 1-queue, 2-queue, and 3-queue, respectively. Arriving priority-p packets are thought to enter the RPQ scheduler through the circle shown in Figure 1(a).

Assuming that packets start to arrive at time 0, Figure 1(b) shows a feasible snapshot of the FIFO queues at the end of the first rotation interval, that is, at time $t = \Delta^-$. In Figure 1(b), packets are shown as dark boxes and are labeled with their priority index. Note that the arrived packets have been added to the queue with the same priority label as the packet. The scheduler always selects a packet for transmission from the nonempty queue with the highest-priority label. Since the 0-queue is empty, the packets in the 1-queue have highest priority. Here, we assume that the figure depicts a scenario at the end of the first rotation interval, at time Δ^-.

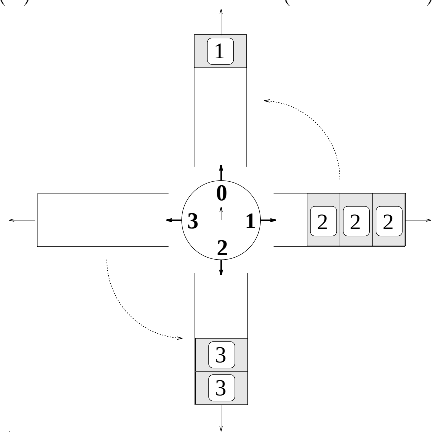
In Figure 1(c) we show the new tagging of the FIFO queues after the first queue rotation at time Δ. The rearrangement of FIFO queues and priority labeling is indicated as a counterclockwise rotation of the queues in Figure 1(c). Since the (former) 1-queue now becomes the new 0-queue, no packets will arrive to this queue during the next rotation interval. Figure 1(d) depicts a feasible scenario in the second rotation interval, shown at time $2\Delta^-$. Note that due to the previous queue rotation, new arriving packets from priority p are now queued behind priority-$(p+1)$ packets. In Figure 1(e) we show the result of the second queue rotation at time 2Δ.



(a) Initial state (at time 0^-).



(b) Before first rotation (at time Δ^-).

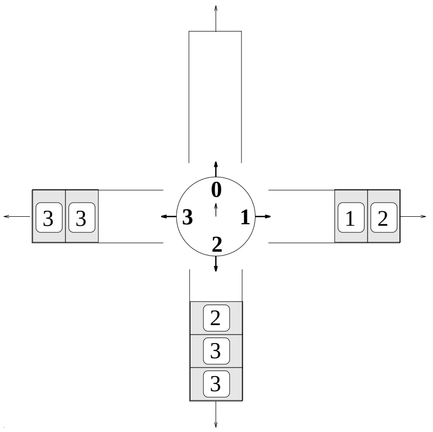


(c) After first rotation (at time Δ).

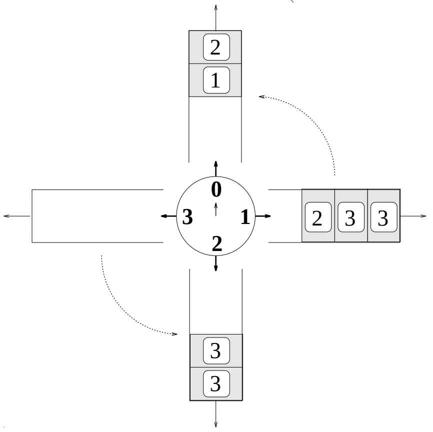
Figure 1: Example of RPQ scheduling.

Note that in order to perform the rotation, we require that the 0-queue is empty at time $2\Delta^-$, the end of the second rotation interval. However, since the delay bounds are set to Δ, 2Δ and 3Δ for priorities 1, 2, and, 3, a nonempty 0-queue at the end of a rotation interval implies a deadline violation for some packet. Thus, if we can guarantee that the delay requirements of all packets are met, we can ensure that the 0-queue is empty at the end of each rotation interval.

Since the queue rotation in RPQ can be implemented by merely updating a set of pointers, the additional complexity of RPQ scheduling as compared to SP scheduling is low if the rotation interval is selected large. By selecting



(d) Before second rotation (at time $2\Delta^-$).



(e) After second rotation (at time 2Δ).

Figure 1: Example of RPQ scheduling.

$\Delta = \infty$, i.e., queues are never rotated, an RPQ scheduler is equivalent to an SP scheduler. On the other hand, by reducing the length of the rotation interval, the RPQ scheduler can approximate an EDF scheduler arbitrarily closely. However, for small values of Δ, the number of FIFO queues needed by the RPQ scheduler will grow large. Therefore, for the practical use of RPQ, one needs to investigate how well RPQ approximates EDF with a small number of FIFO queues. The examples presented in Section 6 indicate that even for relatively large values of Δ, an RPQ scheduler supports a number of connections with deterministically bounded delays similar to an EDF scheduler.

5.3 Schedulability Conditions of RPQ Schedulers

In this subsection, we give the necessary and sufficient schedulability conditions for an RPQ scheduler. Before we state the conditions, we will discuss the traffic workload that is transmitted before an arbitrary packet is completely transmitted by the RPQ scheduler. This will help obtain an intuitive understanding of the schedulability conditions.

In Figure 2 we show the arrivals of packets, indicated as arrows, at an RPQ scheduler over a period of five rotation intervals. The figure depicts, from top to bottom, packet arrivals at the FIFO queues from connections with priorities $p+2$, $p+1$, p, $p-1$, and $p-2$. For the moment, we assume that $n_p = p$, that is, the delay bounds are given by $d_p = p\Delta$ for connections in priority set $\mathcal{C}_p$. The boundaries of the rotation intervals of length Δ are indicated in Figure 2 as dashed vertical lines.

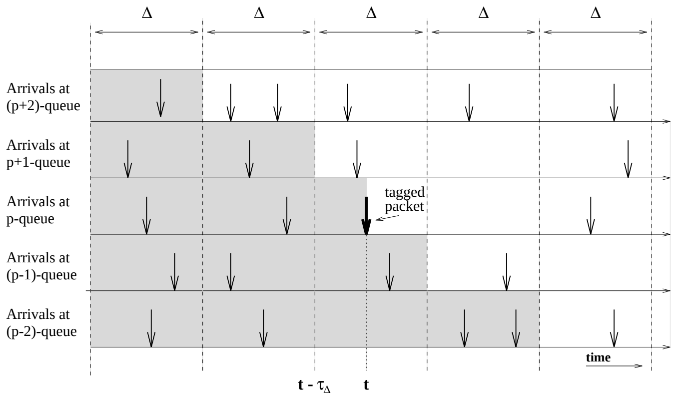


Figure 2: Workload served before a tagged packet in an RPQ scheduler.

Consider the tagged packet from priority p that arrives at the RPQ scheduler at time t as indicated in Figure 2. The packet arrives in a rotation interval that started at time $t - \tau_\Delta$. Thus, queue rotations are performed at times:

$$\{(t - \tau_\Delta) + j\,\Delta \mid j \text{ an integer}\} \qquad (12)$$

The shaded areas in Figure 2 indicate the time intervals during which packet arrivals from a given priority are transmitted before the tagged priority-p packet with arrival time t. Since packets from connections in the same priority set are served in FIFO order, all arrivals from priority p that occur before time t are served before the tagged packet. Packets from lower priority sets ($q > p$) that are transmitted before the tagged packet are those packets that at time t reside in a n_q-queue with $n_q \le n_p$. For priority $(p+1)$, this includes all packet arrivals until time $t - \tau_\Delta$, the end of the last rotation interval that ends before time t, and for priority $(p+2)$, all arrivals until time $t - \tau_\Delta - \Delta$, the end of the last rotation interval that starts before time t.

For priority $p-1$, the maximum number of packets that is transmitted before the tagged packet is limited to arrivals before $t - \tau_\Delta + \Delta$, the end of the current (at time t) rotation interval. At time $t - \tau_\Delta + \Delta$, the priority-p queue to which the tagged packet has arrived is relabeled as the $(n_p - 1)$-queue. Thus, all priority-$(p-1)$ packets that arrive after the end of the current rotation interval will be queued behind the tagged packet. Likewise, the packets of priority-$(p-2)$ served before the tagged packet are limited to those packets that arrive before $t - \tau_\Delta + 2\Delta$, the end of the first rotation interval that begins after time t.

Next we relax the assumption $n_p = p$ and obtain time intervals for each priority q during which arrivals of this priority are transmitted before the packet from connection $k \in \mathcal{C}_p$ with arrival time t and departure time $t + \delta$. The intervals are as follows:

$$\begin{cases} [0, t - \tau_\Delta + d_p - d_q + \Delta] & \text{for all } q > p \\ [0, t] & \text{for } q = p \\ [0, \min\{t + \delta, t - \tau_\Delta + d_p - d_q\}] & \text{for all } q > p \end{cases} \tag{13}$$

Note that the given intervals are maximal if the arrival time t of the tagged packet occurs immediately after a queue rotation, i.e., if $\tau_\Delta = 0$. Also note that the discussion so far has ignored that the transmission of a packet cannot be interrupted. The effects of nonpreemptiveness of packet transmissions on the workload served before a packet are addressed in Appendix C.

The following theorem states the necessary and sufficient schedulability conditions for an RPQ scheduler. As in Section 4, we use s_p^{max} to denote the maximum transmission time of packets from a priority-p connection, i.e., $s_p^{max} = \max_{j \in \mathcal{C}_p} s_j^{max}$.

Theorem 3 *A set $\mathcal{N}$ of connections that is characterized by $\{A_j^*, d_j\}_{j \in \mathcal{N}}$ with $d_j = n_p \Delta$ for $j \in \mathcal{C}_p$ is schedulable on an RPQ scheduler with rotation interval Δ if and only if for all $t \geq d_1$:*

$$t \geq \sum_{j \in \mathcal{C}_1} A_j^*(t - d_1) + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t + \Delta - d_q) + \max_{r, d_r > t + \Delta} s_r^{max} \tag{14}$$

where $\max_{r, d_r > t} s_r^{max} \equiv 0$ for $t > d_P - \Delta$.

A complete proof of Theorem 3 is presented in Appendix C. Note that the schedulability conditions in Theorem 3 are similar to the conditions for EDF schedulers in Theorem 1 when the RPQ scheduler has $P = |\mathcal{N}|$ priority levels with $|\mathcal{C}_p| = 1$ for all p, that is, there is only one connection per priority level. In this case, we can replace equation (14) by:

$$t \geq A_1^*(t - d_1) + \sum_{j=2}^{|\mathcal{N}|} A_j^*(t + \Delta - d_j) + \max_{j, d_j > t + \Delta} s_j^{max} \tag{15}$$

The last equation shows that the condition in Theorem 3 converges to the one in Theorem 1 if we reduce the length of the rotation interval Δ. This observation is manifested in the following corollary.

Corollary 2 *Any EDF-schedulable set $\mathcal{N}$ of connections can be made RPQ-schedulable by appropriately reducing Δ.*

Example: We show how the schedulability conditions of RPQ simplify when using the (σ, ρ)-traffic model. We again assume that each priority set $\mathcal{C}_p$ contains only one connection p. Rewriting the conditions in equation (14) and inserting the traffic constraint function for the (σ, ρ)-model we obtain:

$$\begin{cases} t \geq \sigma_1 + \rho_1 (t - d_1) + \max_{q > 1} s_q^{max} \\ \qquad\qquad \text{for } d_1 \leq t < d_2 - \Delta \\ t \geq \sum_{q=1}^{p} \sigma_q + \rho_1 (t - d_1) + \sum_{q=2}^{p} \rho_q (t + \Delta - d_q) + \max_{q > p} s_q^{max} \\ \qquad\qquad \text{for } d_p - \Delta \leq t < d_{p+1} - \Delta, 2 \leq p < P \\ t \geq \sum_{q=1}^{P} \sigma_q + \rho_1 (t - d_1) + \sum_{q=2}^{P} \rho_q (t + \Delta - d_q) \\ \qquad\qquad \text{for } t \geq d_P - \Delta \end{cases} \tag{16}$$

If the scheduler is stable, i.e., $\sum_{q=1}^{P} \rho_q < 1$, we obtain from these inequalities that

$$d_1 \geq \sigma_1 + \max_{q > 1} s_q^{max} \tag{17}$$

and

$$d_p \geq \frac{\sigma_p + \sum_{q=1}^{p-1} (\sigma_q - \rho_q d_q) + (1 - \rho_1)\Delta + \max_{q > p} s_q^{max}}{1 - \sum_{q=1}^{p-1} \rho_q} \quad \text{for all } p > 1 \tag{18}$$

6 Empirical Evaluation

In this section we apply the schedulability conditions derived so far to compare the number of connections with a bounded delay service that can be supported with EDF schedulers, SP schedulers, and RPQ schedulers. Particularly, we want to evaluate how well the RPQ scheduler approximates the EDF scheduling method for relatively large values of the rotation interval Δ.

We present three sets of examples. In the first example, we compare the schedulers using three connection groups that comply to the (σ, ρ)-traffic model. In the second example, we consider similar connection groups, but assume that the traffic model is a discrete version of the (σ, ρ)-traffic model. In the third set of examples, we compare the schedulers for traffic arrivals obtained from actual MPEG video traces.

Since we are mostly interested in comparing scheduling methods, both examples only consider a single network switch. For schedulers that have a rate-controlling mechanism (see Subsection 2.1), our schedulability conditions can be directly applied to multi-hop routes. We wish to emphasize that the consideration of a single switch reflects the focus of this paper on scheduling methods and is not due to technical limitations of our work.

6.1 Example 1

We investigate schedulability for a set of three connection groups at an ATM switch with a link rate of 155 Mbps which transmits 53-byte cells. By assuming that all traffic

Connection Group	Index j	Delay Bound d_j *(msec)*	Burst Size σ_j *(cells)*	Rate ρ_j *(Mbps)*
Low Delay	1	12	4,000	10 – 155
Medium Delay	2	24	2,000	10 – 155
High Delay	3	36	4,000	10 – 155

Table 1: Parameter set for Example 1.

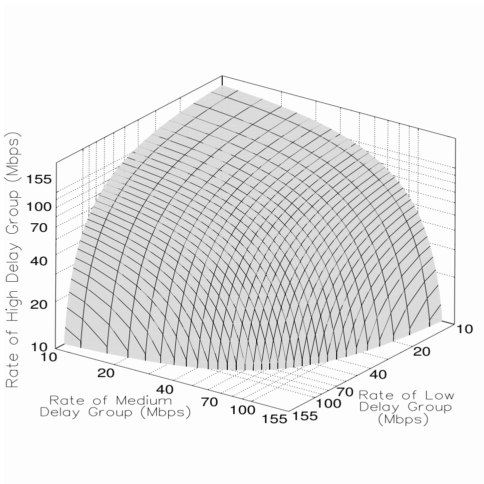


Figure 3: Schedulable region without delay constraints.

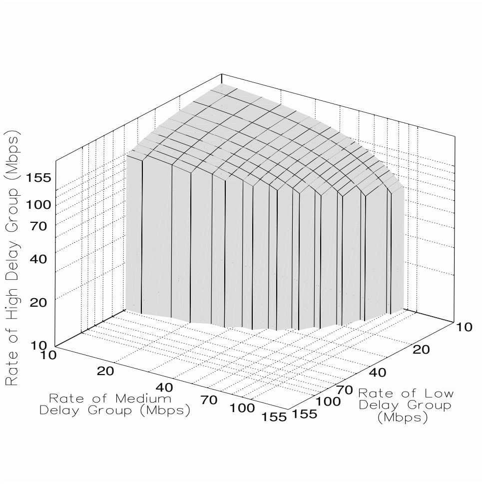


Figure 4: Schedulable region for EDF.

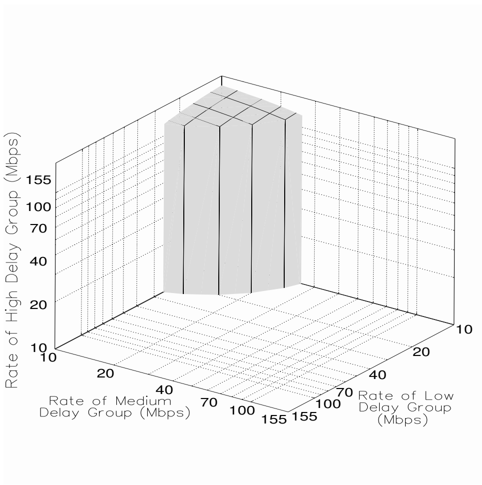


Figure 5: Schedulable region for SP.

at the switch belongs to one of three connection groups, we are able to give a graphical representation of the schedulability at a scheduler. We assume that the aggregated traffic of a connection group is characterized by the (σ, ρ)-model, that is, the traffic constraint function for traffic from connection group j is given by $A_j^*(t) = \sigma_j + \rho_j t$. The traffic parameters and the delay bounds of the connection groups are given in Table 1.

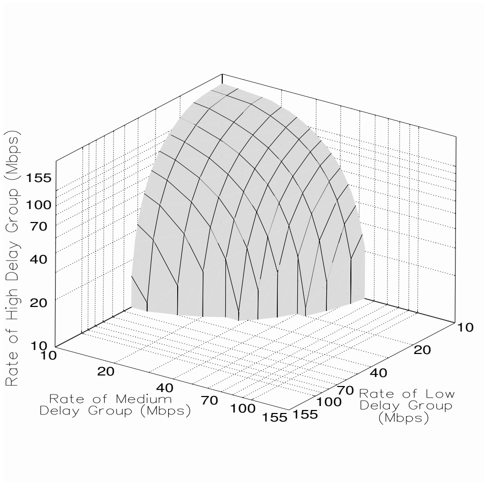
The results for the given parameter set are presented in Figures 3–6. By considering different transmission rates for the connection groups from Table 1, we graph the range of values for which the connection groups satisfy the schedulability conditions. The volume below the surface graphs depicts the schedulable region [10] of a scheduler, that is, the rate values ρ_i at which the schedulability conditions are satisfied. All values that are not schedulable lie in the region above the surface.

As a reference case for our example, we show in Figure 3 the schedulable region when packets do not have delay bounds, i.e., $d_1 = d_2 = d_3 = \infty$. Since in this case, the schedulability of the connection group is only bounded by the stability condition given in equation (3), the region in Figure 3 is an upper bound for any selection of delay bound parameters.

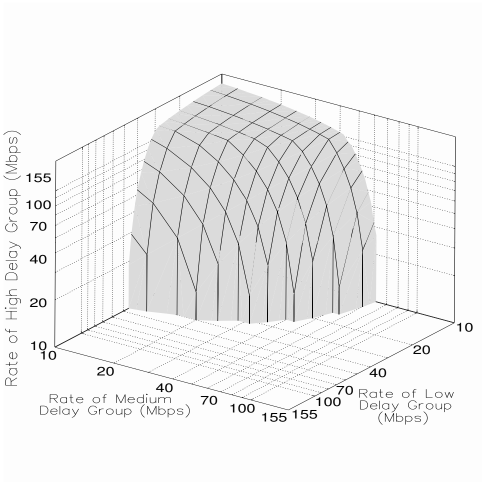
In Figures 4 and 5 we show the schedulable regions for the EDF scheduler and the SP scheduler. For our parameter set, the volume covered by the schedulable region of the EDF scheduler clearly contains that of the SP scheduler. Since the maximum packet transmission time, i.e., an ATM cell with 53 bytes, is identical for all connections, the schedulable region Figure 4 is maximal for any scheduler according to Corollary 1.

In Figures 6(a)–6(d) we show the schedulable regions for RPQ schedulers with rotation intervals set to $\Delta = 6, 4, 3, 2$ msec. With the given values for Δ, the number of FIFO queues required by RPQ are given by $36/\Delta + 1$, that is, the scheduler must provide between 7 and 19 FIFO queues. Two observations are noteworthy. First, Figure 6 shows that the volume covered by the graphs grows monotonically as the rotation interval is decreased and converges to the schedulable region of the EDF scheduler shown in Figure 4. For $\Delta = 3$ msec, the schedulable region of RPQ is almost identical to the schedulable region of EDF. However, for the above examples complete convergence is obtained only if $\Delta \leq 0.033$ msec.

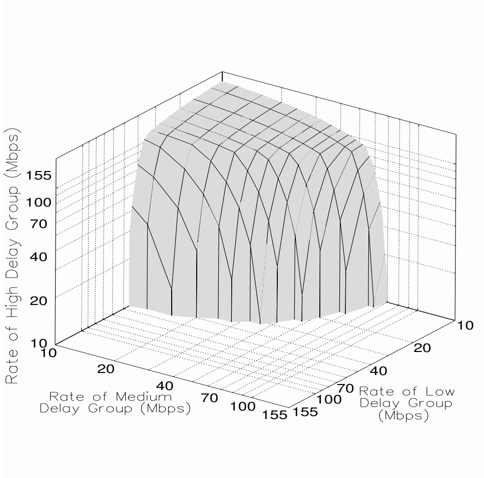
Second, by comparing the graphs of RPQ with that of the SP scheduler in Figure 5, we see that for $\Delta \leq 4$ msec, the schedulable region of RPQ is always above the graph of SP. For $\Delta > 4$ msec, some parameter sets are schedulable with SP but not with RPQ, and vice versa. This illustrates



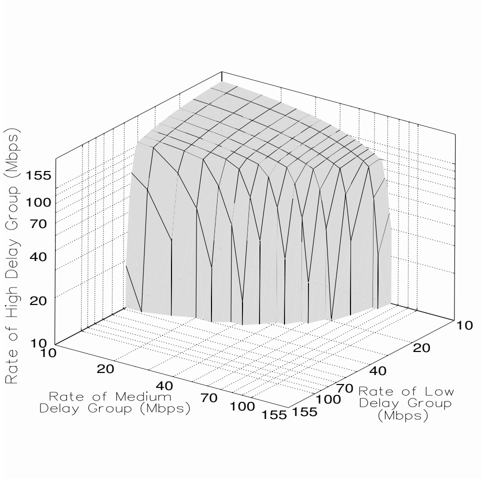
(a) RPQ Scheduler ($\Delta = 6$ msec).



(b) RPQ Scheduler ($\Delta = 4$ msec).



(c) RPQ Scheduler ($\Delta = 3$ msec).



(d) RPQ Scheduler ($\Delta = 2$ msec).

Figure 6: Schedulable regions for RPQ.

that SP is equivalent to RPQ for $\Delta = \infty$, but that the schedulable regions of the two schedulers are different for finite values of Δ.

6.2 Example 2

In the previous example, RPQ approximated an EDF scheduler well even when the rotation interval was selected large. Next we present an example where RPQ is close to EDF only for a relatively short rotation interval, thus, a large number of FIFO queues. We perform three modifications to the parameter set of Example 1, all of which have a negative impact on the effectiveness of RPQ as an approximation of the EDF scheduling method. First, the delay bounds relative to the transmission rate are reduced; second, the packet transmission times are increased, and third, the traffic models used to characterize the worst-case traffic are derived from a discrete-time model.

The traffic model in this example is a discrete-time version of the leaky bucket traffic policing mechanism [22] which characterizes the worst-case traffic on a connection group j by a parameter set (T_j, b_j, s_j^{max}), where T_j denotes the shortest period of packet arrivals, b_j denotes the maximum burst size, and s_j^{max} denotes the maximum packet size. With this traffic model, the traffic constraint function

	Group Index j	Delay Bound d_j *(msec)*	Packet Size s_j^{max} *(Bytes)*	Burst Size b_j *(packets)*	Period T_j *(msec)*	Maximum Average Rate *(Mbps)*
Connection Group						
Low Delay	1	2	1250	8	0.2–10	$\approx$ 1–50
Medium Delay	2	4	1250	9	0.2–10	$\approx$ 1–50
High Delay	3	8	1250	2	0.2–10	$\approx$ 1–50

Table 2: Parameter set for Example 2.

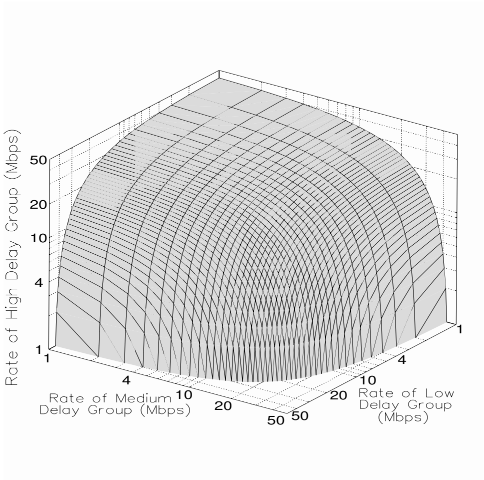


Figure 7: Schedulable region without delay constraints.

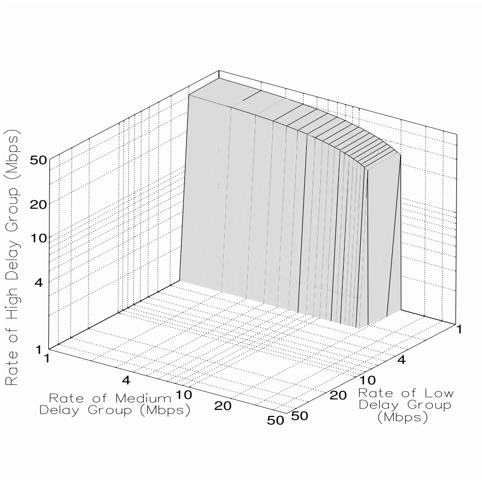


Figure 9: Schedulable region of SP.

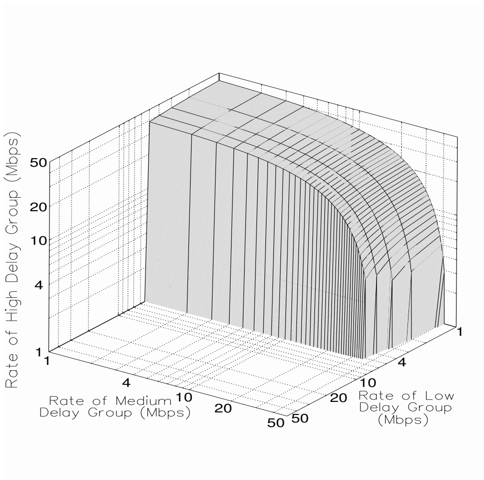


Figure 8: Schedulable region for EDF.

$A_j^*(t)$ for connection group j is given by:

$$A_j^*(t) = b_j s_j^{max} + \left\lfloor \frac{t}{T_j} \right\rfloor s_j^{max} \qquad (19)$$

We assume that the data rate of the scheduler is set to 50 Mbps. The parameter sets for the connection groups are shown in Table 1. As in Example 1, we have three connection groups. The delay bounds of packets are given by $d_1 = 2$ msec for the low delay group, $d_2 = 4$ msec for the medium delay group, and $d_3 = 8$ msec for the high delay group. For all connection groups, we assume constant packet sizes set to 1250 Bytes, and burst sizes that are 2–9 packets per connection group. The periods of the connection groups are selected such that the maximum average data rate varies between 1 and 50 Mbps for the each delay group.

To evaluate the effects of deadlines in our parameter set, we show in Figure 7 the schedulable region if packets do not have deadlines, i.e., when $d_1 = d_2 = d_3 = \infty$. In Figures 8 and 9 we illustrate the schedulable region for an SP scheduler and an EDF scheduler, respectively. We see that EDF admits much more traffic than SP for our parameter set.

In Figures 10(a)–10(d) we show the graphs obtained for RPQ multiplexers with rotation intervals set at values ranging from $\Delta = 1$ msec to $\Delta = 0.05$ msec. Here, the number of FIFO queues required by the RPQ multiplexer is given by $8/\Delta + 1$. Therefore the number of FIFO queues required by the RPQ schedulers is given by 9 queues in Figure 10(a), 17 queues in Figure 10(b), 41 queues in Figure 10(c), and 161 queues in Figure 10(d). In Figure 10(a) we see that for $\Delta = 1$, that is, 9 FIFO queues, the schedulable region of the RPQ scheduler is clearly worse than that of the SP scheduler shown in Figure 9. By decreasing the rotation interval to $\Delta = 0.5$ msec and $\Delta = 0.2$ msec, shown in Figures 10(b) and 10(c), RPQ improves upon SP. (The schedulable region of SP is completely contained within that of RPQ for $\Delta \leq 0.4$ msec). By comparing Figure 8 with Figures 10(c)–10(d) we can see that the schedulable region of RPQ approximates that of EDF well only for $\Delta = 0.05$ msec (161 queues).

For certain parameter sets, RPQ is identical to EDF only for $\Delta = 0$. Let us construct such a scenario for a scheduler with two types of connections. The packet transmission times are assumed to be constant with $s_1^{max} = s_2^{max} = 1$ msec, and the traffic constraint function for a connection from type $j = 1, 2$ is given by:

$$A_j^*(t) = \left\lfloor \frac{t}{T_j} \right\rfloor + 1 \tag{20}$$

Assume that the periods of the connections are given by $T_1 = T_2 = 20\ msec$, and the delay bounds are given by $d_1 = 10$ msec and $d_2 = 20$ msec. For this constructed example, an EDF and an SP scheduler admit an identical number of connections. Let us denote by N_1 the number of admitted connections from type 1, and by N_2 the number of admitted connections from type 2. Assuming that there is at least one connection from either type, i.e., $N_1 > 0$ and $N_2 > 0$, we obtain for both EDF and SP schedulers that connections are admissible if $N_1 < 10$ and $N_1 + N_2 \leq 20$. With the same parameters, it can be verified from Theorem 3 that the connections are RPQ schedulable with a rotation interval Δ only for $N_1 < 10$ and $N_1 + N_2 + \lceil \Delta \rceil \leq 20$. Consequently, RPQ admits the same number of connections as SP and EDF only if $\Delta = 0$.

6.3 Example 3

In this example, we take actual traces of MPEG-compressed video and empirically evaluate how the EDF, SP, and RPQ schedulers are capable of supporting MPEG compressed video transmissions with deterministically bounded delays. As in Example 1, we assume a single ATM multiplexer operating at 155 Mbps. All traffic sources are obtained from MPEG-compressed sequences of video frames, where each video frame is fragmented into 53-byte ATM cells with a payload of 48 bytes. The cells of a frame are spaced evenly over the time period between two consecutive frames.

We consider two video sequences, the first trace ("*News*") depicts a TV News show, the second trace ("*Settop*") is obtained from a desktop video camera and contains a recording of a 'talking head'. Both traces are taken from a publicly available library of MPEG traces [20]. The traces were encoded with the UC Berkeley MPEG-I software encoder [9] using the following parameter settings [20]: the encoder input is 384×288 pixels per frame, the frame pattern is *IBBPBBPBBPBB* and each sequence consists of about 40,000 video frames, corresponding to approximately 30 minutes of video. We refer to [20] for a discussion of the complete parameter sets and statistical analyses of the traces.

Previous studies on MPEG transmissions in networks with bounded delay services [13, 23] have pointed out that simple traffic models such as the (σ, ρ)-model are not adequate for expressing the burstiness and temporal correlations of MPEG-encoded video. Since we are primarily

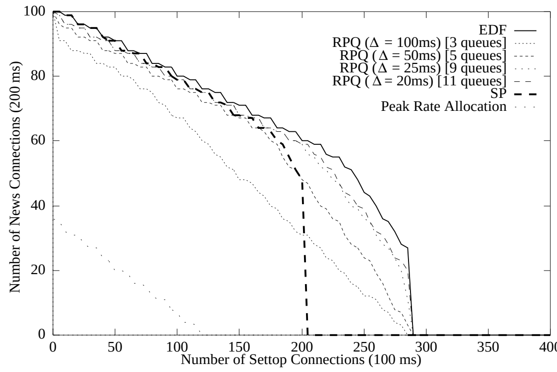


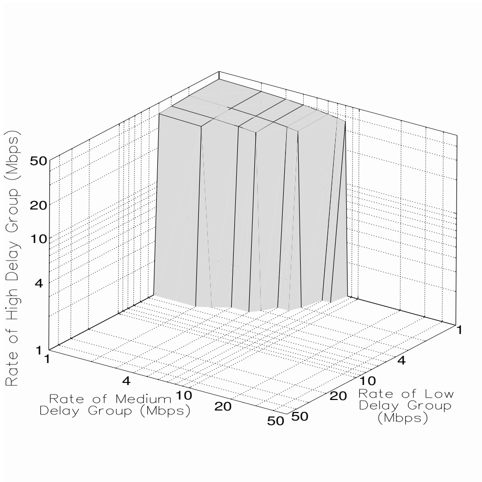
Figure 11: Schedulability of schedulers with MPEG traces.

interested in a comparison of scheduling methods, we do not wish to skew the results by the selection of a particular traffic model. Therefore, we will use a traffic characterization that is independent of any parameterized traffic model. We denote the traffic that is generated by an MPEG source in time interval $[t, t+\tau]$ by $A[t, t+\tau]$, and we take the following traffic constraint function $\mathcal{E}$ for the MPEG traffic source [2, 23]:

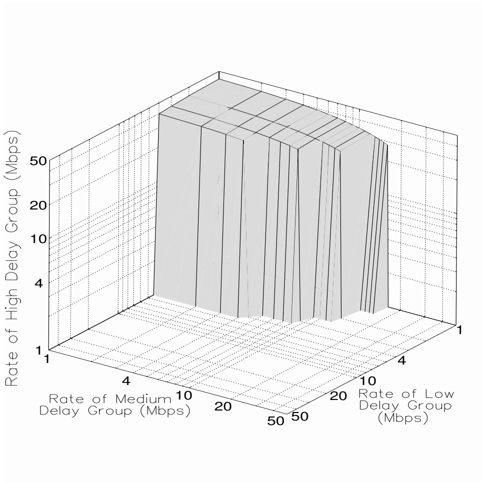
$$\mathcal{E}(\tau) := \max_t A[t, t+\tau]$$

It can be easily shown that $\mathcal{E}$, referred to as the *empirical envelope*, is the best time-invariant and subadditive bound for the given traffic arrival A, in the sense that all traffic constraint functions A^* for A satisfy $\mathcal{E}(t) \leq A^*(t)$ [2, 23]. It is obvious, that the empirical envelope is not a practical traffic constraint function since policing the envelope requires knowledge of all frame sizes in the entire video sequence. Nonetheless, by selecting the best possible traffic characterization we obtain an upper bound for schedulability as compared to any other traffic model.

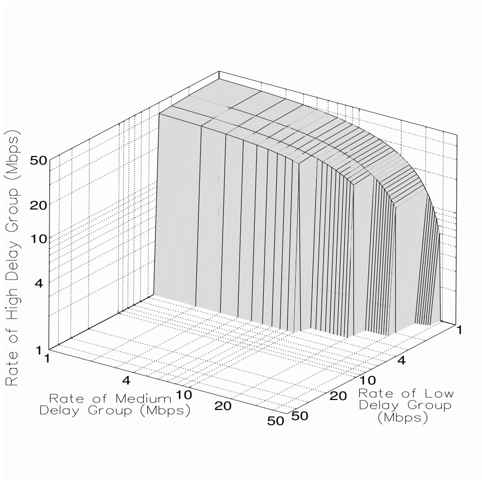
We use the schedulability conditions from Theorems 1–3 to determine the maximum number of *News* and *Settop* connections that can be simultaneously supported on a 155 Mbps link, when the delay bounds are set to $d_{Settop} = 100$ msec for all *Settop* connections and $d_{News} = 200$ for all *News* connections. The results are shown in Figure 11 where we depict the results for an EDF scheduler, an SP scheduler and RPQ schedulers with different rotation intervals. We have included a plot that shows the maximum number of supported connections if admission control is based on a peak rate allocation, where the peak rate is determined from the transmission time of the largest *I-frame* in the sequence [23]. For the RPQ scheduler, the rotation intervals are set to $\Delta = 100, 50, 25$ and 20 msec, requiring, respectively, $3, 5, 9$, and 11 FIFO queues. Figure 11 shows that the difference between the schedulers is small if the number of *News* connections is large, and more noticeable for large numbers of *Settop* connections. Similar to the observations made in Example 1, the connection sets that are RPQ-schedulable approximate the EDF-schedulable sets well even for relatively large rotation intervals. The plots for RPQ and EDF become identical for $\Delta \leq 100\ \mu s$.



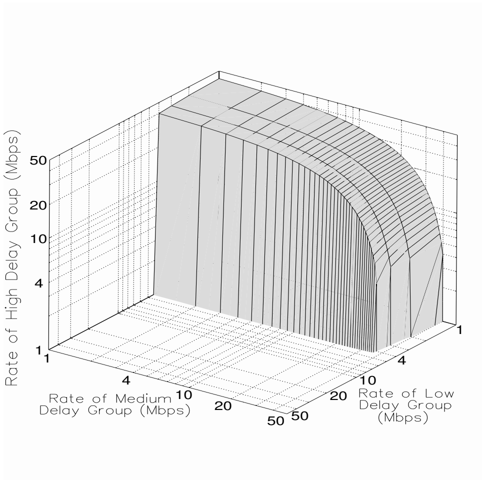
(a) RPQ Scheduler ($\Delta = 1$ msec).



(b) RPQ Scheduler ($\Delta = 0.5$ msec).



(c) RPQ Scheduler ($\Delta = 0.2$ msec).



(d) RPQ Scheduler ($\Delta = 0.05$ msec).

Figure 10: Schedulable region for RPQ.

7 Conclusions

We have studied packet schedulers and their exact schedulability conditions for switches in connection-oriented packet-switching networks with a bounded delay service, that is, a service with deterministically bounded network delays for all packets on a connection. The schedulability conditions of a packet scheduler verify that all packets meet their delay requirements and are an essential component of the connection admission control tests. We presented and proved necessary and sufficient schedulability conditions for three packet schedulers: Earliest-Deadline-First (EDF), Static-Priority (SP), and Rotating-Priority-Queues (RPQ). The Rotating-Priority-Queues (RPQ) packet scheduler is a new scheduling method which approximates EDF scheduling and can be implemented with FIFO queues, similar to SP. The FIFO queues of an RPQ scheduler are *'rotated'* after fixed time intervals. We showed that by properly decreasing the time between queue rotations and increasing the number of available FIFO queues, the efficiency of RPQ scheduling closely approximates the efficiency of an EDF scheduler. In a set of examples we demonstrated that the selection of the packet scheduler has a large impact on the number of connections that can be supported in a network with a bounded delay service. We showed that the RPQ scheduling method usually approximates EDF well even when the number of FIFO queues of the RPQ scheduler is small.

8 Acknowledgments

We would like to thank an anonymous reviewer who pointed out a problem in an earlier version of Theorem 2.

A Proof of Theorem 1

The proof of Theorem 1 proceeds in three steps. First we derive an expression for the traffic that is transmitted before an arbitrary packet. Using this expression, we will show sufficiency and necessity of the conditions in (4) in the second and the third step, respectively.

(a) Workload served before an arbitrary packet

We will derive the workload transmitted before a tagged packet from connection $k \in \mathcal{N}$ that arrives at the EDF scheduler at time t and is completely transmitted at time $t+\delta$. We assume that time 0 indicates the start of a busy period, that is, the scheduler is empty before time 0. Let $A_j^{\leq x}[t, t+\tau]$ denote the traffic arrival from connection j in time interval $[t, t+\tau]$ with deadlines less than or equal to x. We use $W^{\leq x}(y)$ to denote the workload in the scheduler at time y due to packets with deadlines less than or equal to time x, and $W^{k,t}(t+\tau)$ $(0 \leq \tau \leq \delta)$ to denote the workload in the scheduler at time $t+\tau$ that must be served before the tagged packet from connection k with arrival time t can depart. Note that $W^{k,t}(t+\tau)$ includes the tagged packet.

Let $t - \hat{\tau}$ $(\hat{\tau} \geq 0)$ be the last time before t when the scheduler does not contain traffic with a deadline less than or equal to the deadline of the tagged packet. Since the scheduler is empty before time 0, the time $t - \hat{\tau}$ is guaranteed to exist. Expressed in terms of $W^{\leq x}(y)$, $\hat{\tau}$ is given by:

$$\hat{\tau} = \min\{z \mid W^{\leq t+d_k}(t-z) = 0, z \geq 0\} \tag{21}$$

Hence, in time interval $[t-\hat{\tau}, t+\delta)$, the scheduler always contains work with a deadline less than or equal to $t+\delta$. With $\hat{\tau}$ we can determine $W^{k,t}(t+\tau)$, the workload that is transmitted before the tagged packet. $W^{k,t}(t+\tau)$ is composed of:

- The remaining transmission time of the packet that is in transmission at time $t-\hat{\tau}$, denoted by $R(t-\hat{\tau})$. With equation (21), this packet has a deadline greater than $t+d_k$.
- $\sum_{j\in\mathcal{N}} A_j^{\leq t+d_k}[t-\hat{\tau}, t+\tau]$, that is, all arrivals in time interval $[t-\hat{\tau}, t+\tau]$ with deadlines less than or equal to $t+d_k$. Note that $A_k^{\leq t+d_k}[t-\hat{\tau}, t+\tau]$ includes the tagged packet.
- The length of time interval $[t-\hat{\tau}, t+\tau]$.

From equation (21) we obtain that in time interval $[t-\hat{\tau}+R(t-\hat{\tau}), t+\tau]$, the EDF scheduler only transmits traffic with a deadline less than or equal to $t+d_k$. Therefore, we obtain the following expression for $W^{k,t}(t+\tau)$ with $0 \leq \tau \leq \delta$:

$$W^{k,t}(t+\tau) = \sum_{j\in\mathcal{N}} A_j^{\leq t+d_k}[t-\hat{\tau}, t+\tau] + R(t-\hat{\tau}) - (\hat{\tau}+\tau) \tag{22}$$

Since all traffic from a connection j that arrives after time $t+d_k-d_j$ has a deadline greater than $t+d_k$, we can rewrite (22) as:

$$W^{k,t}(t+\tau) = \sum_{j\in\mathcal{N}} A_j[t-\hat{\tau}, \min\{t+\tau, t+d_k-d_j\}] + \\ +R(t-\hat{\tau}) - (\hat{\tau}+\tau) \tag{23}$$

(b) Proof of Sufficiency

Consider the tagged packet from connection k that arrives at time t. The packet does not have a deadline violation if there exists a τ $(0 \leq \tau \leq d_k)$ such that:

$$W^{k,t}(t+\tau) = 0 \tag{24}$$

where $W^{k,t}(t+\tau)$ is as given in equation (23). For $\tau = d_k$ we obtain from equation (23):

$$W^{k,t}(t+d_k) =$$

$$\sum_{j\in\mathcal{N}} A_j[t-\hat{\tau}, t+d_k-d_j] + R(t-\hat{\tau}) - (\hat{\tau}+d_k) \tag{25}$$

$$\leq \sum_{j\in\mathcal{N}} A_j^*(\hat{\tau}+d_k-d_j) + R(t-\hat{\tau}) - (\hat{\tau}+d_k) \tag{26}$$

Equation (26) follows from equation (25) with the property of A_j^* from equation (1).

$R(t-\hat{\tau})$, the remaining transmission time of any packet in transmission at time $t-\hat{\tau}$, can be bounded as follows. Since such a packet has a deadline greater than $t+d_k$ by choice of $\hat{\tau}$, this packet is associated with some connection j with delay bound:

$$d_j > \hat{\tau} + d_k \tag{27}$$

Assuming that the maximum length of such a packet is s_j^{max}, we obtain the following inequality from equation (27):

$$R(t-\hat{\tau}) \leq \max_{j, d_j > \hat{\tau}+d_k} s_j^{max} \tag{28}$$

Combining equation (28) with equation (26) yields the following inequality:

$$W^{k,t}(t+d_k) \leq \sum_{j\in\mathcal{N}} A_j^*(\hat{\tau}+d_k-d_j) + \max_{j, d_j > \hat{\tau}+d_k} s_j^{max} - (\hat{\tau}+d_k) \tag{29}$$

With equation (4), we have:

$$W^{k,t}(t+d_k) \leq 0 \tag{30}$$

Hence, there exists a $\tau \leq t+d_k$ such that $W^{k,t}(t+\tau) = 0$.

(c) Proof of Necessity

Suppose that inequality (4) is violated at some time $t \geq 0$, that is:

$$t < \sum_{j\in\mathcal{N}} A_j^*(t-d_j) + \max_{j, d_j > t} s_j^{max} \tag{31}$$

Let us first assume that $t \leq \max_{j\in\mathcal{N}} d_j$. Consider a scenario where the scheduler is empty before time 0^-, and at time 0 a packet from connection i with $s_i^{max} = \max_{j,d_j>t} s_j^{max}$ arrives with a transmission time of s_i^{max}. Now, starting at time 0, packets from connections j with $d_j \le t$ arrive according to A_j^*. Since the scheduler is nonpreemptive, the packet from connection i will be transmitted before any packet from a connection j with $d_j \le t$. Now we look at the workload in the scheduler at time t with a deadline less than or equal to t. The traffic from connection j that arrives in time interval $[0,t]$ with a deadline at or before time t is given by $A_j^{\le t}[0,t] = A_j^*(t-d_j)$. The maximum time period in the interval $[0,t]$ during which the scheduler is transmitting packets with a deadline $\le t$ is given by $t - \max_{j,d_j>t} s_j^{max}$. Note that the actual time period may be shorter since the scheduler may be idle in the interval $[0,t]$. Therefore, $W^{\le t}(t)$ is given by:

$$W^{\le t}(t) \ge \sum_{d_j \le t} A_j^*(t-d_j) - t + \max_{j,d_j>t} s_j^{max} \qquad (32)$$

$$= \sum_{j \in \mathcal{N}} A_j^*(t-d_j) - t + \max_{j,d_j>t} s_j^{max} \qquad (33)$$

Note that equation (33) follows from $A_j^*(x) = 0$ if $x < 0$. With the assumption from (31) we have $W^{\le t}(t) > 0$, that is, at time t the scheduler contains traffic with a deadline less than or equal to t. Thus, there must be a packet in the scheduler at time t with a deadline violation.

If $t > \max_{j\in\mathcal{N}} d_j$ we create a similar scenario. The scheduler is empty before time 0^-, and starting at time 0, packets from all connections $j \in \mathcal{N}$ arrive according to A_j^*. Then the workload in the scheduler at time t with a deadline less than or equal to t is given by $\sum_{j\in\mathcal{N}} A_j^*(t-d_j)$. Since the maximum workload that can be transmitted in the interval $[0,t]$ is given by t, and since, by assumption, $\max_{k,d_k>t} s_k^{max} \equiv 0$ for $t > \max_{j\in\mathcal{N}} d_j$ we obtain equations (32) and (33). As before, (31) implies $W^{\le t}(t) > 0$, yielding a deadline violation for some packet. □

B Proof of Theorem 2

Similar to the proof of Theorem 1, we proceed in three steps. First we obtain an expression for $W^{p,t}(t+\tau)$, the workload in the scheduler at time $t+\tau$ that must be served before a priority-p packet with arrival time t can leave the scheduler. As in the proof of Theorem 1, $W^{p,t}(t+\tau)$ includes the workload introduced by the tagged packet. Then we prove sufficiency and necessity of the schedulability condition.

(a) Workload served before an arbitrary packet

Suppose that a (*tagged*) packet from connection $k \in \mathcal{C}_p$ arrives at the scheduler at time t with a transmission time of s where $s_p^{min} \le s \le s_p^{max}$, and that it leaves the scheduler at time $t+\delta$. Due to non-preemption, the packet starts transmission at time $t+\delta-s$. The arrival time t falls into a *priority-p busy period*, that is, a time period where the scheduler contains packets with priority $\le p$. The priority-p busy period started at time $t-\hat{\tau}$, i.e.,

$$\hat{\tau} = \min\{z \mid \sum_{q=1}^{p} \sum_{j\in\mathcal{C}_q} W_j(t-z) = 0, z \ge 0\} \qquad (34)$$

where $W_j(t)$ is the workload from connection j in the scheduler at time t. Denoting by $W^{p,t}(t+\tau)$ $(0 \le \tau \le \delta)$ the workload in the SP scheduler at time $t+\tau$ that is served *before* the departure time of the tagged packet, $W^{p,t}(t+\tau)$ is determined for $t \le t+\tau \le t+\delta$ by:

- $R(t-\hat{\tau})$, the remaining transmission time of a priority-r packet with $r > p$ that is in transmission at time $t-\hat{\tau}$.
- Traffic from priority-p connections that arrives in time interval $[t-\hat{\tau}, t]$, i.e., before or with the arrival of the tagged packet. This traffic is given by $A_j[t-\hat{\tau},t]$ for $j \in \mathcal{C}_p$.
- Traffic from higher priority connections that arrives in time interval $[t-\hat{\tau}, t+\tau)$, given by $A_j[t-\hat{\tau}, t+\tau)$ for for $j \in \mathcal{C}_q$ and $q < p$.
- The traffic that is transmitted in time interval $[t-\hat{\tau}, t+\tau]$.

Formally, $W^{p,t}(t+\tau)$ is given for all $0 \le \tau \le \delta$ by:

$$W^{p,t}(t+\tau) = \sum_{j\in\mathcal{C}_p} A_j[t-\hat{\tau},t] + \sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j[t-\hat{\tau},t+\tau) + R(t-\hat{\tau}) - (\hat{\tau}+\tau) \qquad (35)$$

Since the transmission of the tagged priority-p packet begins at time $t+\delta-s$ and higher priority packets cannot preempt the transmission of the tagged packet after $t+\delta-s$, at $t+\delta-s$ the workload that should be transmitted before the departure time of the tagged packet is equal to the transmission time of the tagged packet. Therefore, the departure time $t+\delta$ is determined by:

$$\delta = s + \min\{z \mid W^{p,t}(t+z) = s, z \ge 0\} \qquad (36)$$

(b) Proof of Sufficiency

We will show that, for an arbitrary packet from connection $k \in \mathcal{C}_p$ with transmission time s with $s_p^{min} \le s \le s_p^{max}$ that arrives at time t, condition (7) guarantees that the packet will depart before $t + d_p$.

From the definition of A_j^* in (1) we have:

$$\sum_{j\in\mathcal{C}_p} A_j[t-\hat{\tau},t] \le \sum_{j\in\mathcal{C}_p} A_j^*(\hat{\tau}) \qquad (37)$$

$$\sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j[t-\hat{\tau},t+\tau) \le \sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j^*((\hat{\tau}+\tau)^-) \qquad (38)$$

Since the remaining non-preemptable transmission time of priority-r traffic $(r > p)$ at time $t-\hat{\tau}$ is maximal if a low priority packet with maximum transmission time begins transmission at $(t-\hat{\tau})^-$, we obtain:

$$R(t-\hat{\tau}) \le \max_{r>p} s_r^{max} \qquad (39)$$

With equations (37)–(39), we can give the following bound for $W^{p,t}(t+\tau)$ in equation (35):

$$W^{p,t}(t+\tau) \leq \sum_{j\in\mathcal{C}_p} A_j^*(\hat{\tau}) + \\ + \sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j^*((\hat{\tau}+\tau)^-) + \max_{r>p} s_r^{max} - (\hat{\tau}+\tau) \tag{40}$$

With condition (7) there exists a $\tau' \leq d_p - s_p^{min}$ such that:

$$W^{p,t}(t+\tau') \leq s_p^{min} \tag{41}$$

Since $W^{p,t}(t+\tau')$ includes the transmission time s of the tagged packet and since $s \geq s_p^{min}$, the tagged packet is either in transmission at time $t+\tau'$, or it is completely transmitted before time $t+\tau'$. In either case, the packet will depart before $t+d_p$.

(c) Proof of Necessity

Let us assume that the condition in equation (7) does not hold, that is, there exists a priority p and a time t such that for all $0 \leq t \leq t+d_p - s_p^{min}$ within a priority-p busy period such that, for all $0 \leq \tau \leq d_p - s_p^{min}$:

$$t+\tau < \sum_{j\in\mathcal{C}_p} A_j^*(t) + \sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j^*((t+\tau)^-) - s_p^{min} + \max_{r>p} s_r^{max} \tag{42}$$

Now assume a scenario where the SP scheduler is empty before time 0^-, and at time 0^- traffic from connection $i \in \mathcal{C}_r$ with $s_i^{max} = \max_{r>p} s_r^{max}$ arrives. Suppose that, starting at time 0, all connections j with priorities p or higher transmit the maximum traffic permitted by their traffic constraint functions A_j^*, with one exception: we delay the arrival of a packet from connection $k \in \mathcal{C}_p$ with transmission time s_p^{min} that would arrive before time t until time t.[5]

If the delayed packet from connection $k \in \mathcal{C}_p$ with arrival time t has not started transmission at time $t+\tau$, then the traffic that arrives in time interval $[0^-, t+\tau]$ and that should have been transmitted when the tagged packet departs consists at least of:

- $s_i^{max} = \max_{r>p} s_r^{max}$, the transmission time of traffic that arrived at time 0^-,
- $A_j^*(t)$ with $j \in \mathcal{C}_p$, the traffic from priority p that arrived in time interval $[0, t]$ including the tagged packet with arrival time t,
- $A_j^*((t+\tau)^-)$ with $j \in \mathcal{C}_q$ and $q < p$, the high priority traffic which arrives in time interval $[0, t+\tau)$.

Note that the in time interval $[0^-, t+\tau]$, the SP-scheduler may be idle or transmit low-priority traffic from a priority $q > p$. We assume the best case, that is, in $[0^-, t+\tau]$ the SP scheduler is always transmitting traffic with priority $\leq p$. Hence, we obtain the following lower bound for $W^{p,t}(t+\tau)$, the workload that is transmitted before the delayed packet is completely transmitted:

$$W^{p,t}(t+\tau) \geq \sum_{j\in\mathcal{C}_p} A_j^*(t) + \\ + \sum_{q=1}^{p-1}\sum_{j\in\mathcal{C}_q} A_j^*((t+\tau)^-) + \max_{r>p} s_r^{max} - (t+\tau) \tag{43}$$

With our assumption in (42) we obtain that $W^{p,t}(t) > s_p^{min}$ at in the entire time interval $[t, t+d_p - s_p^{min}]$. Therefore, the delay δ of the packet as calculated from equation (36) yields $\delta > d_p$, and a deadline violation occurs at time $t+d_p$. $\Box$

C Proof of Theorem 3

As before, we first derive an expression for $W^{p,t}(t+\tau)$, the workload in the scheduler at time $t+\tau$ that must be served before a packet from priority p that arrived at time t departs from the scheduler. Following, we prove the sufficiency and necessity of the theorem.

(a) Workload served before an arbitrary packet

The following derivation continues our discussion from Subsection 5.3, where we derived the time intervals during which arrivals from any priority level are served before a tagged packet from priority p with arrival time t. Recall that the derivations in Subsection 5.3 ignored that packet transmissions cannot be preempted. To accurately describe the workload served before the tagged packet we still need to account for the effects of non-preemption of packets. We define time $t-\hat{\tau}$ to be the last time prior to t at which the scheduler does not hold packets that are to be transmitted before the tagged packet. For priorities $q \leq p$, these are all the arrivals in time interval $[0, t-\hat{\tau}]$; and for priorities $q > p$, according to Subsection 5.3 all arrivals in the interval $[0, \min\{t-\hat{\tau}, (t-\tau_\Delta)+(n_p-n_q+1)\Delta)$. Denoting by $W_j(t)$ the workload in the RPQ scheduler from connection $j \in \mathcal{N}$, we can determine $\hat{\tau}$ by:

$$\hat{\tau} = \min\{z \mid z \geq 0, \\ \sum_{q=1}^{P}\sum_{j\in\mathcal{C}_q} W_j(\min\{t-z,\ (t-\tau_\Delta)+(n_p-n_q+1)\Delta\}) = 0\} \tag{44}$$

Thus, the workload that is transmitted by the (non-preemptive) RPQ scheduler in time interval $[t-\hat{\tau}, t+\delta]$ is limited to the packets arriving after time $t-\hat{\tau}$ *plus* the remaining transmission time of a packet that is in transmission at time $t-\hat{\tau}$.

Denoting by $W^{p,t}(t+\tau)$ the workload in the scheduler at time $t+\tau$ $(0 \leq \tau \leq \delta)$ that will be transmitted before the tagged priority-p packet that arrives at time t (including the tagged packet), $W^{p,t}(t+\tau)$ is determined by:

[5] Note that we assume that such a construction is feasible. However, we do not believe that the assumption is very restrictive. For example, with fixed-sized packets, $s_p^{max} = s_p^{min}$, the construction is obtained by merely delaying the last packet. If the maximum packet size is at least twice the minimum packet size, $s_p^{min} < s_p^{max}/2$, the construction is simply done by splitting the last packet of size s_p^{max} that arrives before time t into two packets, one with size $s_p^{max} - s_p^{min}$ and one with size s_p^{min}, and by delaying the arrival of the second packet.

- The workload due to packets from connections $j \in \mathcal{C}_p$ that arrive before or together with the tagged packet, that is, in time interval $[t-\hat{\tau}, t]$.
- The workload due to packets from all connections $j \in \mathcal{C}_q$ $(q > p)$ that arrive in the busy period before the end of the $(n_q - n_p)$th rotating interval that ends *before* time t, that is, in time interval $[t-\hat{\tau}, (t-\tau_\Delta) + (n_p - n_q + 1)\Delta]$.
- The workload due to packets from all connections $j \in \mathcal{C}_q$ $(q < p)$ that arrive before time $t+\tau$ *and* before the $(n_p - n_q)$th rotating interval that ends *after* time t, or, equivalently, arrivals in the time interval $[t-\hat{\tau}, \min\{t+\tau,\ (t-\tau_\Delta) + (n_p - n_q)\Delta\}]$.
- Due to non-preemption, the remaining transmission time of any low-priority packet that is in transmission at time $t-\hat{\tau}$, denoted by $R(t-\hat{\tau})$.
- The workload that has been transmitted in time interval $[t-\hat{\tau}, t+\tau]$.

Hence, for $0 \le \tau \le \delta$, $W^{p,t}(t+\tau)$ is given by:

$$
\begin{aligned}
&W^{p,t}(t+\tau) = \\
&\quad \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j[t-\hat{\tau}, \min\{t+\tau, (t-\tau_\Delta) + (n_p - n_q)\Delta\}] + \\
&\quad + \sum_{j \in \mathcal{C}_p} A_j[t-\hat{\tau}, t] + \\
&\quad + \sum_{q=p+1}^{P} \sum_{j \in \mathcal{C}_q} \quad A_j[t-\hat{\tau}, (t-\tau_\Delta) + (n_p - n_q + 1)\Delta] + \\
&\quad + R(t-\hat{\tau}) - (\hat{\tau} + \tau)
\end{aligned} \tag{45}
$$

Since the tagged priority-p packet leaves the switch at time $t+\delta$, the packet is scheduled for transmission by the RPQ scheduler at time $t+\delta-s$, where $s \le s_k^{max}$ is the transmission time of the packet. Thus, we can describe δ as follows:

$$
\begin{aligned}
\delta &= s + \min\{z \mid W^{p,t}(t+z) = s,\ z \ge 0\} && (46) \\
&\le \min\{z \mid W^{p,t}(t+z) = 0,\ z \ge 0\} && (47)
\end{aligned}
$$

Note that a deadline violation of the tagged packet occurs if and only if $\delta > n_p \Delta$.

(b) Proof of Sufficiency

We will show that condition (14) guarantees that the packet does not have a deadline violation, i.e., that there exists a $\delta \le n_p \Delta$ such that $W^{p,t}(t+\delta) = 0$, where $W^{p,t}(t+\delta)$ is given in equation (45).

Let $t - \tau_\Delta$ denote the rotation time immediately preceding t and let $t - \hat{\tau}$ be the last time that the scheduler does not contain a packet that will be transmitted before the tagged packet from connection k, as obtained in equation (44).

Consider the workload served before our tagged packet at time $t + n_p\Delta$. From equation (45) we can obtain $W^{p,t}(t+n_p\Delta)$. By taking advantage of the subadditivity of the traffic constraint functions A_j^*, and the fact that the highest priority set (lowest index) with incoming traffic is $\mathcal{C}_1$ we use equation (45) to provide the following upper bound for $W^{p,t}(t+n_p\Delta)$:

$$
\begin{aligned}
W^{p,t}(t+n_p\Delta) \quad \le \quad & \sum_{j \in \mathcal{C}_1} A_j^*(\hat{\tau} + (n_p - n_1)\Delta) + \\
& + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} + (n_p - n_q + 1)\Delta) + \\
& + R(t-\hat{\tau}) - (\hat{\tau} + n_p\Delta)
\end{aligned} \tag{48}
$$

We now consider $R(t-\hat{\tau})$, the remaining transmission time of a packet that is in transmission at time $t-\hat{\tau}$. By selecting $\hat{\tau}$ as in equation (44), the delay bound of such a packet must exceed $\hat{\tau} + n_p\Delta + \Delta$. It follows that $R(t-\hat{\tau}) \le \max_{u, d_u > \hat{\tau} + d_p + \Delta} s_u^{max}$, which we substitute in equation (48). With the condition of equation (14) in Theorem 3, the right-hand side of equation (48) cannot be positive. This is easily verified by substituting 't' in equation (14) with '$\hat{\tau} + n_p\Delta$' on the right-hand side in equation (48). Hence, we have $W^{p,t}(t+n_p\Delta) \le 0$ and with the argument used for equation (47), it is guaranteed that the tagged packet will meet its deadline.

(c) Proof of Necessity

Assume that the condition in (14) is violated at some time $t > n_1\Delta$, that is:

$$
t < \sum_{j \in \mathcal{C}_1} A_j^*(t - n_1\Delta) + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t + \Delta - n_q\Delta) + \max_{u, d_u > t+\Delta} s_u^{max} \tag{49}
$$

Assume without loss of generality that time t occurs immediately after a queue rotation, and thus the time since the last rotation, τ_Δ, is small. We will construct a feasible pattern of packet arrivals that results in a packet violation at time t. Consider a scenario in which the scheduler is empty before time 0^-, and at time 0^- a packet from connection $k \in \mathcal{C}_u$ arrives with $s_u^{max} = \max_{r, d_r > t+\Delta} s_r^{max}$. Also assume that starting at time 0 all connections j transmit at their maximum rate as permitted by their traffic constraint functions A_j^*, with one exception: the last packet submitted to the network from a priority-1 connection *before or at* time $t - n_1\Delta$ is submitted at exactly time $t - n_1\Delta$.

To derive the workload $W^{j,t}(t+\tau)$ as shown in equation (45) for the tagged packet, we need to consider that (a) the tagged packet is from priority 1, (b) each connection j sends according to A_j^* in the interval $[0, t]$, and (c) the scheduler can be idle in time interval $[\max_{r, d_r > t+\Delta} s_r^{max}, t]$. Therefore, we obtain that at time times $(t - n_1\Delta + \tau)$ with $\tau < n_1\Delta$ the following workload must be transmitted before the tagged packet can leave the scheduler:

$$
\begin{aligned}
W^{1, t-n_1\Delta}(t - n_1\Delta + \tau) \ge & \sum_{j \in \mathcal{C}_1} A_j^*(t - n_1\Delta) + \\
& + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t - \tau_\Delta + (n_1 - n_q + 1)\Delta) + \\
& + \max_{r, d_r > t+\Delta} s_r^{max} - (t + \tau)
\end{aligned} \tag{50}
$$

Inserting our assumption from (49) into equation (50) we see that $W^{1,t-n_1\Delta}(t) > 0$. Observing in equation (50) that $W^{1,t-n_1\Delta}$ is strictly decreasing in the time interval $[t - n_1\Delta, t]$, we have $W^{1,t-n_1\Delta}(t-n_1\Delta+\tau) > 0$ for all $0 \leq \tau \leq n_1\Delta$. Therefore, the tagged packet cannot be completely transmitted at any time in the interval $[t-n_1\Delta, t]$, resulting in a deadline violation for this packet. □

References

[1] C.M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne. Real-Time Communication in Packet-Switched Networks. *Proceddings of the IEEE*, 82(1):122–139, January 1994.

[2] C.-S. Chang. Stability, Queue Length and Delay of Deterministic and Stochastic Queueing Networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[3] R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[4] R. L. Cruz. A Calculus for Network Delay, Part II: Network Analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[5] D. Ferrari. Real-Time Communication in an Internetwork. *Journal of High-Speed Networks*, 1(1):79–103, 1992.

[6] D. Ferrari and D. C. 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.

[7] L. Georgiadis, R. Guerin, and A. Parekh. Optimal Multiplexing on a Single Link: Delay and Buffer Requirements. In *Proc. IEEE Infocom'94*, pages 524–532, June 1994.

[8] S. J. Golestani. A Framing Strategy for Congestion Management. *IEEE Journal on Selected Areas In Communications*, 9(7):1064–1077, September 1991.

[9] K. L. Gong. Berkeley MPEG-1 Video Encoder, User's Guide. Technical report, UC Berkeley, Computer Science Division, March 1994.

[10] J. M. Hyman, A. A. Lazar, and G. Pacifici. Real-Time Scheduling with Quality of Service Constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, September 1991.

[11] J. M. Hyman, A. A. Lazar, and G. Pacifici. Joint Scheduling and Admission Control for ATS-based Switching Nodes. In *Proc. ACM Sigcomm '92*, pages 223–234, August 1992.

[12] C. R. Kalmanek, H. Kanakia, and S. Keshav. Rate Controlled Servers for Very High-Speed Networks. In *Proc. IEEE Globecom '90*, pages 300.3.1–300.3.9, December 1990.

[13] E.W. Knightly and H. Zhang. Traffic Characterization and Switch Utilization Using a Deterministic Bounding Interval Dependent Traffic Model. In *Proc. IEEE Infocom'95*, pages 1137–1145, April 1995.

[14] J. F. Kurose. On Computing Per-Session Performance Bounds in High-Speed Multi-Hop Computer Networks. In *Proc. 1992 ACM Sigmetrics and Performance '92*, pages 128–139, June 1992.

[15] S. S. Lam and G. G. Xie. Burst Scheduling: Architecture and Algorithm for Switching Packet Video. In *Proc. IEEE Infocom'95*, pages 940–950, April 1995.

[16] Y. Lim and J. E. Kobza. Analysis of a Delay-Dependent Priority Discipline in an Integrated Multiclass Traffic Fast Packet Switch. *IEEE Transactions on Communications*, 38(5):659–665, May 1990.

[17] C. L. Liu and J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment. *Journal of the ACM*, 20(1):46–61, January 1973.

[18] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[19] J. M. Peha. *Scheduling and Dropping Algorithms to Support Integrated Services in Packet-Switched Networks*. PhD thesis, Stanford University, June 1991.

[20] O. Rose. Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems. Technical Report 101, Institute of Computer Science, Universiy of Wurzburg, Germany, February 1995.

[21] L. Takács. *Introduction to the Theory of Queues*. Oxford University Press, 1962.

[22] J. S. Turner. New Directions in Communications (or Which Way to the Information Age?). *IEEE Communications Magazine*, 25(8):8–15, October 1986.

[23] D.E. Wrege, E.W. Knightly, H. Zhang, and J. Liebeherr. Deterministic Delay Bounds for VBR Video in Packet-Switching Networks : Fundamental Limits and Tradeoffs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

[24] H. Zhang. Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines. *Computer Communications Journal*: Special Issue on System Support for Multimedia Computing, 18(10), October 1995.

[25] H. Zhang. *Service Disciplines for Packet-Switching Integrated-Services Networks*. PhD thesis, University of California - Berkeley, November 1993.

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

[27] H. Zhang and S. Keshav. Comparison of Rate-Based Service Disciplines. In *Proc. ACM Sigcomm '91*, pages 113–121, Zurich, Switzerland, September 1991.

[28] H. Zhang. Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. *Proceedings of the IEEE*, 83(10):1374–1399, 1995.

[29] L. Zhang. Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proc. ACM Sigcomm '90*, pages 19–29, September 1990.

[30] Q. Z. Zheng and K. G. Shin. On the Ability of Establishing Real-Time Channels in Point-to-Point Packet Switched Networks. *IEEE Transactions on Communications*, 42(3):1096–1105, March 1994.

Jörg Liebeherr (S'88 - M'92) received the Diplom-Informatiker degree from the University of Erlangen, Germany, in 1988 and the Ph.D. degree in Computer Science from the Georgia Institute of Technology in 1991.

From 1990 until 1991 he did his dissertation work partly at the IBM T. J. Watson Research Center. In 1992 he was a Postdoctoral Fellow in the Computer Science Division of the University of California, Berkeley. Since September 1992 he is an Assistant Professor in the Department of Computer Science at the University of Virginia. He received an NSF Research Initiation Award in 1993 and an NSF CAREER Award in 1996. His research interests are multimedia networks, broadband networks, real-time systems and performance evaluation. He is a member of IEEE and ACM (Sigcomm and Sigmetrics).

Dallas E. Wrege (S'93) received the B.A. degree in Applied Mathematics from Harvard University, Cambridge, in 1992.

He is currently a Research Assistant in the Multimedia Networks Group at the Department of Computer Science, University of Virginia, where he is working toward the Ph.D. degree. His main research interests are in the area of multimedia and quality-of-service networks.

Domenico Ferrari Domenico Ferrari earned a Dr.Ing. degree in Electronics Engineering at the Politecnico di Milano, Italy, in 1963. He became an Assistant Professor at the Politecnico in 1967. In 1970, he joined the faculty of the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, where he is now Professor Emeritus of Computer Science. From 1983 to 1987, he served as Chairman of the Computer Science Division and Associate Chairman of the Department. Since October 1995, he is the Director of the Center for Research on the Applications of Telematics to the Organization and Society (CRATOS) at the Università Cattolica, Piacenza, Italy. In 1989, he founded the Tenet Group, which has designed and built two real-time protocol suites, and experimented with them on several networking testbeds.

Dr. Ferrari has been, from its inception in April 1988 until June 1990, Deputy Director, and until 1995 Vice President, of the International Computer Science Institute (ICSI). During the 1988-95 period, he has also led ICSI's Networks Group. For his contributions to the evaluation and improvement of computer systems performance, he has been elevated to the grade of Fellow of the Institute of Electrical and Electronics Engineers in 1987. He also received the 1987 A.A. Michelson Award of the Computer Measurement Group for "outstanding contributions to computer metrics". He is one of the three Main Editors of the ACM/Springer-Verlag Journal on Multimedia Systems.