Design and Analysis of a High-Performance Packet Multiplexer for Multiservice Networks with Delay Guarantees *

Jörg Liebeherr *Dallas E. Wrege*

Department of Computer Science
University of Virginia
Charlottesville, VA 22903

Abstract

A major challenge for the design of multiservice networks with quality of service guarantees is an efficient implementation of a bounded delay service, that is, a service that guarantees maximum end-to-end delays for every packet from a single traffic stream. A crucial component of a bounded delay service is the packet multiplexing technique employed at network switches that must keep the variable statistical multiplexing delays below a predetermined threshold. To achieve a high utilization of network resources, the multiplexing technique must be sufficiently sophisticated to support a variable set of delay bounds for a large number of traffic streams. On the other hand, since multiplexing of packets is to be performed at the data rate of the network links, the complexity of the multiplexer should be strictly limited. A novel packet multiplexing technique, called *Rotating Priority Queues* (RPQ), is presented which exploits the tradeoff between efficiency, i.e., the ability to support many connections with delay bounds, and low complexity. The operations required by the RPQ multiplexer are similar to those of the simple, but inefficient, *Static Priority* (SP) multiplexer. The overhead of RPQ, as compared to SP, consists of a periodic rearrangement (*rotation*) of the priority queues. It is shown that queue rotations can be implemented by updating a set of pointers. The efficiency of RPQ can be made arbitrarily close to the highly efficient, yet complex, *Earliest Deadline First* (EDF) multiplexer. Exact expressions for the worst case delays in an RPQ multiplexer are derived and compared to expressions for an EDF multiplexer.

Key Words: Real-time networks, Bounded Delay Service, Multiplexing, Quality of Service, Packet Scheduling, Admission Control.

*This work is supported in part by the National Science Foundation under Grant No. NCR-9309224.

1 Introduction

Recent advances in fiber optic and computer hardware technology have dramatically increased the switching and transmission capacity of communication systems and have enabled the design of packet switching multiservice networks which support transmission of data, voice, and video traffic. Due to the stringent service requirements of voice and video transmissions, multiservice packet switching networks must provide guarantees on throughput, delay, delay jitter, and error rate.

In network architectures that provide service guarantees [3, 8, 12], the relationship between network clients and the network provider can be defined in terms of a *traffic contract* [23]. When requesting a connection with service guarantees, the client submits a specification of its traffic together with a set of desired guarantees. The network performs *admission control tests* to verify that the requested service can be given without violating any previously given guarantees. Once a connection is established, the network provider commits to support the service guarantees until the connection is released. To ensure that a network client does not exceed the specified traffic, the network monitors the client's traffic and prevents excessive traffic from entering the network (*traffic policing*).

A major challenge in the design of multiservice networks is the implementation of a *bounded delay service*, that is, a communication service with deterministically bounded delays for all packets from a single connection. A rigorous approach to a bounded delay service must consider all delay types that a packet may incur, including fixed processing and propagation delays, and variable statistical multiplexing delays at network switches. Since fixed delays result from physical or technological constraints, the implementation of a bounded delay service is centered around the design of appropriate packet multiplexers 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 of the connections, a large number of packet multiplexers can provide bounds on delays [7]; however, most multiplexers will result in an inefficient use of network resources. The performance of a packet multiplexer in providing bounded delay services can be determined by the degree to which it satisfies the following requirements [26]:

- *Efficiency:* An efficient use of network resources such as link bandwidth can only be achieved if the packet multiplexers can support bounded delays for a large number of connections.

- *Flexibility:* A packet multiplexer must be sufficiently flexible to satisfy a diverse set of delay requirements. A FIFO multiplexer, which can only support one delay bound for all connections, is an example of a packet multiplexer with insufficient flexibility.

- *Complexity:* Since multiplexing of packets must be performed at the speed of the transmission link, the complexity of the packet multiplexer must be kept minimal. If the operations at the

multiplexer consume more time than the actual transmission of a packet, transmission links will be left idle most of the time.

- *Analyzability:* The admission control functions which determine whether a new connection may result in delay bound violations of requested or existing connections must have available analytical *schedulability conditions* for the multiplexers, that is, expressions which determine if the maximum delay of any packet may exceed its delay bound. If exact schedulability conditions are not available, the admission control tests will unnecessarily limit the number of connections in the network, thus, reducing the efficiency of the multiplexer.

Note that a single packet multiplexer cannot simultaneously optimize all of the above criteria. In particular, high efficiency and low complexity are contradictory design goals. Thus, each multiplexing technique presents a tradeoff in satisfying the above requirements. In this study, we propose a new multiplexing technique, referred to as *Rotating Priority Queues* (RPQ), that can satisfy all of the above requirements to a very high degree. RPQ can be considered as a hybrid of the well-known Earliest Deadline First (EDF) and Static Priority (SP) packet multiplexers, both of which have been considered for bounded delay services [8, 26].

EDF multiplexers which always select the packet with the shortest time until a delay bound violation for transmission offer high efficiency and flexibility.[1] A disadvantage of EDF multiplexing is that packets in the multiplexer queue must be sorted according to their deadlines, hence, introducing a considerable degree of complexity. A Static Priority (SP) multiplexer supports a fixed number of priority levels for connections and maintains one FIFO queue for each priority level. The first packet in the highest-priority FIFO queue is selected for transmission. Due to the implementation with FIFO queues, the complexity of SP multiplexing is low. However, the efficiency achieved by SP multiplexing is significantly inferior to EDF multiplexing [15]. Also, since SP multiplexers 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) multiplexer combines the advantages of high efficiency of EDF multiplexers with the low complexity of SP multiplexers. The flexibility of RPQ in providing different delay bounds is the same as for SP. 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. We will show that by selecting the length of *rotation intervals* to be sufficiently small, RPQ can approximate the efficiency of EDF arbitrarily closely.

[1] Preemptive EDF scheduling is optimal in respect to efficiency. However, since packet transmissions cannot be preempted, the result does not apply to packet multiplexers.

We present the exact schedulability conditions for RPQ multiplexers; hence, we can accurately provide the delay bounds obtained with RPQ multiplexing. By comparing the schedulability conditions of RPQ with those of EDF and SP multiplexers [15] we can precisely compare the efficiency of these multiplexers. We are able to show that RPQ approximates the efficiency of EDF to a very high degree even for large values of the rotation interval.

The remainder of this study is structured as follows. In Section 2 we discuss previous work on multiplexing techniques for networks with bounded delay services. In Section 3 we discuss a general traffic and multiplexer model. In Section 4 we present the schedulability conditions of EDF multiplexers as derived in [15]. In Section 5 we present the novel RPQ packet multiplexer, and prove its necessary and sufficient schedulability conditions. In Section 6 we present empirical examples to compare the efficiency of the EDF, SP, and RPQ multiplexing techniques. We conclude our study in Section 7.

2 Previous Work

The transition from classical data networks to multiservice networks with quality-of-service guarantees has emphasized the importance of sophisticated packet multiplexing. In recent years, a considerable number of multiplexing techniques has been developed that can provide provable delay bounds. Several studies employ probabilistic traffic models and derive bounds for the delay distribution obtained for various multiplexers [2, 11, 14, 17, 24, 25]. Here, we only consider research on multiplexers that attempts to provide deterministic delay bound guarantees.

Stop-and-Go Queueing [9] and Hierarchical Round Robin (HRR) [13] employ a framing strategy which assigns a fixed portion of link bandwidth in a fixed time frame to connections, similar to Time-Division Multiplexing. The schedulability conditions in these multiplexers can be easily obtained from the frame size. However, since bandwidth assignment to connections is based on the peak rate of traffic both multiplexers have a low efficiency. Additionally, since delay bounds are tied to the frame size, Stop-and-Go as well as HRR have limited flexibility for assigning different delay bounds to connections.

Fair Queueing (FQ) [6] employs a round-robin strategy for selecting packets for transmission. Since in FQ the maximum delay of a packet from a connection is proportional to the traffic generated by this connection, FQ has limited flexibility for assigning delay bounds to connections, in particular, for connections with low bandwidth but stringent delay requirements. The Weighted Fair Queueing (WFQ) multiplexer [6, 18] is an extension of FQ where the delay bound of a connection can be derived from weights that are assigned during connection establishment. Parekh has derived schedulability conditions for WFQ in [18]. A drawback of WFQ is that the weight that is assigned to a connection is dependent on the weights assigned to other connections. Therefore, to maximize the efficiency of an WFQ multiplexer, the weights of all connections must be recalculated each time a connection is established or released.

Delay-EDD [8] is a multiple class version of the Earliest Deadline First (EDF) technique which transmits packets in the order of decreasing deadlines. Jitter-EDD [22] extends Delay-EDD by a holding mechanism for packets and can also provide bounds on network delay variations, i.e., the delay jitter. Ferrari and Verma show sufficient schedulability conditions for EDF multiplexers in [8]. Necessary and sufficient conditions for EDF are derived in [29] for a particular policing algorithm, and in [15] for general policing methods.

The Rate-Controlled Static-Priority multiplexer (RCSP) [26, 27] is based on fixed priorities for each connection. Static priority multiplexers can be implemented with low complexity, however, their efficiency is significantly lower than that of EDF multiplexers [15]. Zhang proves sufficient schedulability conditions for static priority multiplexers in [26, 27]. Using a fluid flow traffic approximation, Cruz has shown necessary and sufficient conditions [4]. For a general class of policing functions, necessary and sufficient conditions are proven in [15].

Several multiplexing techniques can provide throughput guarantees, e.g., Fair Queueing [6], Virtual Clock [28], FIFO+ [3]; however, since schedulability conditions are not available for these multiplexers, delay bound guarantees cannot be directly obtained.

3 Packet Multiplexers for Bounded Delay Services

We consider connection-oriented packet-switching networks with arbitrary topologies. Packets from a particular connection traverse the network on a fixed path of switches and links. At each network switch there is one packet multiplexer for each outgoing link. In the following, we only consider a single multiplexer at an arbitrary network switch. Our results can be applied to routes which include multiple multiplexers by either considering the distortion of the packet stream at each multiplexer as in [5, 18], or by providing a holding mechanisms at each switch as shown in [22, 26].

Next we provide a description of packet multiplexers for networks with a bounded delay service. In Subsection 3.1 we present a general traffic characterization of packet arrivals from a rate-controlled traffic source at a multiplexer. The general characterization allows us to formally express traffic for a large class of policing functions. In Subsection 3.2 we discuss the properties of the multiplexers considered in this study. In Subsection 3.3 we formally define schedulability conditions and admission control tests for general multiplexers.

3.1 Traffic Arrivals

A packet multiplexer which determines the order of packet transmission experiences variable-length packet arrivals from a set $\mathcal{N}$ of connections, with $\mathcal{N} = \{1, 2, \ldots, |\mathcal{N}|\}$. Packet arrivals are assumed to be instantaneous, that is, a packet arrival is considered complete when the last bit of the packet is received.

We use a function A_j to describe the *(actual) traffic arrivals* from connection j, where $A_j[t, t+\tau]$

provides the actual arrivals from connection j in time interval $[t, t+\tau]$. [2] The measure for traffic is the transmission time at the multiplexer.

The traffic arrivals from a connection $j \in \mathcal{N}$ are characterized by a *rate-controlling function* A_j^* and by s_j, the maximum transmission time of any packet from connection j. The rate-controlling function A_j^* is used to describe the maximum traffic arrivals from connection j. The relation between actual and maximum traffic is such that for all times $t > 0$ and for all $\tau \geq 0$, A_j is bounded by A_j^* in the following way: [1, 4]

$$A_j[t, t+\tau] \leq A_j^*[0, \tau] \tag{1}$$

If equation (1) holds, we say that A_j is *rate-controlled by* A_j^*, and we write $A_j \prec A_j^*$. In the following we will use $A_j^*(t)$ as short-hand notation for $A_j^*[0, t]$, and we set $A_j^*(t) = 0$ and $A_j(t) = 0$ for all $t < 0$.

The maximum traffic arrival from a connection j is observed if packets arrive according to the rate-controlling functions, i.e., $A_j = A_j^*$. In this case, due to equation (1), for all t_1 and t_2 we obtain:

$$A_j^*(t_1 + t_2) \leq A_j^*(t_1) + A_j^*(t_2) \tag{2}$$

We assume that the network has policing mechanisms that can enforce the rate-controlling function A_j^* for each connection. In Appendix A we present rate-controlling functions for several policing methods considered in the literature.

3.2 Packet Transmissions

A multiplexer can only transmit one packet at a time. If multiple packets reside at the multiplexer, all but the packet that is transmitted are kept in a queue at the multiplexer. The multiplexer implements a set of rules to select a packet for transmission, referred to as the *scheduling discipline*. We refer to a Σ-multiplexer as a packet multiplexer which implements scheduling discipline Σ. For example, a *FIFO*-multiplexer always selects packets for transmission in the order of their arrival. We only consider *work-conserving* packet multiplexers, that is, multiplexers are never idle if there are packets in the multiplexer queue. We assume that the transmission of a packet cannot be preempted. Thus, the only instants when a multiplexer selects a packet for transmission are (a) after the completion of a packet transmission if the multiplexer queue is non-empty, or (b) after a packet arrival at an empty multiplexer. A packet is considered transmitted if the last bit of the packet is transmitted.

We use a function $W(t)$ to describe the *workload* (or backlog) of traffic that resides in the multiplexer at time $t > 0$ (including the packet in transmission). By setting $W(t) = 0$ for $t < 0$, the workload in the multiplexer at time $t \geq 0$ due to a set $\mathcal{N}$ of connections with arrivals $\{A_j\}_{j \in \mathcal{N}}$.

[2] We use [a,b] to denote the set of all x with $a \leq x \leq b$.

is given by [20]:

$$W(t) = \sup_{0 \le u \le t} \left\{ \sum_{j \in \mathcal{N}} A_j[u,t] - (t-u) \right\} \tag{3}$$

Since packet arrivals are instantaneous, the workload $W(t)$ is a right-continuous function in t. We denote by $W(t^-)$ the workload at time t excluding the arrivals at time t, that is, $W(t^-) = \lim_{h \to 0} W(t-h)$. Note that, for all time instants t at which packets arrive at the multiplexer, we have that $W(t^-) \neq W(t)$, and $W(t^-) = W(t)$ otherwise.

3.3 Schedulability Conditions

Each connection j with traffic to the multiplexer has a *delay bound* d_j that indicates the maximum tolerable delay of any packet from connection j in the multiplexer.[3] A packet from connection j that arrives at the multiplexer at time t is assigned a *deadline* $t + d_j$. If a packet that arrives at time t is not transmitted by its deadline, then a *deadline violation* occurs.

Given a multiplexer, we say that a set $\mathcal{N}$ of rate-controlled connections is *schedulable* if no deadline violation occurs for all feasible arrival functions $\{A_j\}_{j \in \mathcal{N}}$ which conform to equation (1). Schedulability is formally defined as follows:

Definition 1 *Given a Σ-multiplexer and a set $\mathcal{N}$ of connections where each connection $j \in \mathcal{N}$ is characterized by (A_j^*, s_j, d_j), the set of connections is said to be $\underline{\Sigma\text{-schedulable}}$ if for all $t > 0$ and for all arrival functions $\{A_j\}_{j \in \mathcal{N}}$, with $A_j \prec A_j^*$, no deadline violation occurs for any connection.*

The conditions which determine if a set of connections is Σ-*schedulable* are referred to as *schedulability conditions*. With the knowledge of the schedulability conditions we can determine the maximum delay experienced by a packet. Moreover, schedulability conditions are required for admission control tests in bounded delay services. Admission control can be formally defined as follows:

Definition 2 *Given a set $\mathcal{N}$ of Σ-schedulable connections that are characterized by (A_j^*, s_j, d_j) for $j \in \mathcal{N}$, a new connection k with (A_k^*, s_k, d_k) is said to be $\underline{\text{admissible}}$ if the set of connections $\mathcal{N} \cup \{k\}$ is also Σ-schedulable.*

From Definition 2 we see that the efficiency of a bounded delay service is largely determined by the choice of the schedulability conditions. An overly pessimistic schedulability condition will cause rejection of new connections even though admitting the connection may not result in deadline violations.

4 Earliest-Deadline-First Multiplexers

An EDF multiplexer maintains a single queue of untransmitted packets, and the queue is sorted in increasing order of packet deadlines. The EDF multiplexer always selects the packet in the

[3] The delay includes queueing and transmission delays.

first position of the queue, that is, the packet with the lowest deadline, for transmission. The transmission of a packet is not interrupted by the arrival of a packet with a lower deadline. Since the scheduler queue of an EDF multiplexer must be sorted according to the deadlines, each packet arrival involves a search operation to find the correct position of the newly arrived packet in the scheduler queue.

Earliest-Deadline-First (EDF) is known to be a highly efficient multiplexing technique, and, hence, is attractive for use in a network with bounded delay services. However, the complexity of EDF multiplexing which requires the maintenance of a sorted multiplexing queue may prevent the use of EDF in networks that operate at high data rates.

Here, we will briefly review a recent result that presents exact schedulability conditions for EDF multiplexers [15]. In Section 5, we will use the schedulability condition of EDF to show that RPQ can approximate EDF arbitrarily closely.

In the next theorem we present tight schedulability conditions for an EDF multiplexer with the general set of rate-controlled arrival functions defined in Subsection 3.1. A proof of Theorem 1 is presented in [15]. In Section 5, we will use the schedulability conditions of EDF to show that RPQ can approximate EDF arbitrarily closely. We assume that the connections are ordered so that $i < j$ whenever $d_i < d_j$.

Theorem 1 *A set $\mathcal{N}$ of connections where each connection $j \in \mathcal{N}$ is characterized by (A_j^*, s_j, d_j), is EDF-schedulable for all $A_j \prec A_j^*$ if and only if for all $t \geq 0$*

$$t \geq \sum_{j \in \mathcal{N}} A_j^*(t - d_j) \tag{4}$$

and for all t with $d_1 \leq t < d_{|\mathcal{N}|}$:

$$t \geq \sum_{j \in \mathcal{N}} A_j^*(t - d_j) + \max_{d_k > t} s_k \tag{5}$$

Recall from Subsection 3.1 that s_k is the maximum transmission time for any packet from connection k. The first condition (equation (4)) is the schedulability condition for a preemptive EDF multiplexer, and the second condition (equation (5)) considers that packet transmissions cannot be preempted.

The RPQ multiplexer presented in the next section approximates EDF multiplexing with a set of ordered FIFO queues which are rearranged (*"rotated"*) after fixed time intervals. Thus, RPQ multiplexers do not have the complexity of EDF multiplexers, but can support a bounded delay service with efficiency close to that of an EDF multiplexer.

5 The Rotating-Priority-Queues (RPQ) Multiplexer

The Rotating-Priority-Queues (RPQ) multiplexer attempts to exploit the tradeoff presented by the simplicity of a Static-Priority (SP) multiplexer and the efficiency of an EDF multiplexer. Similar to the SP multiplexer, RPQ is implemented with a fixed number of FIFO queues. However, packet arrivals from the same connection are inserted into different FIFO queues depending on the arrival instant of the packet. We will show that the efficiency of the RPQ multiplexer can be arbitrarily close to the efficiency of an EDF multiplexer.

Approximations of EDF multiplexers with a set of ordered FIFO queues have been considered before [16, 19]; however, not in the context of bounded delay services. The Head-of-Line with Priority Jumps (HOL-PJ) multiplexer proposed by Lim and Kobza [16] assigns each FIFO queue a range of laxity values.[4] 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. In another approach [19], the movement of queued packets is avoided by periodically rearranging the order of the FIFO queues. However, the suggested implementation of this approach cannot guarantee the absence of deadline violations and therefore is not applicable in an implementation of a bounded delay service.

Similar to the approach suggested in [19], RPQ multiplexing approximates EDF by reordering FIFO queues after fixed time intervals without moving queued packets. However, RPQ multiplexing can guarantee that no packet exceeds a given delay bound. We discuss the operations of an RPQ multiplexer in the next subsection. Then we derive an expression for the workload in an RPQ multiplexer that is served before an arbitrary packet. This expression is used to develop the necessary and sufficient schedulability conditions.

5.1 Description of the RPQ Multiplexer

The connections with traffic to the RPQ multiplexer are partitioned into P disjoint priority sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P$. Packets from connections in the same priority set $\mathcal{C}_p$ have a common delay bound $d_p = \rho_p \Delta$, where $\Delta > 0$ denotes the length of the so-called *rotation interval*, and ρ is a positive integer with $\rho_p < \rho_q$ if $p < q$ and $\rho_1 > 0$. Thus, all delay bounds are multiples of the rotation interval. Traffic that arrives at the multiplexer from a connection j is limited by a rate-controlling function A_j^*.

The RPQ multiplexer maintains $\rho_P + 1$ ordered FIFO queues. At all times, FIFO queues are tagged with an integral index σ where $0 \leq \sigma \leq \rho_P$; however, the tagging of FIFO queues is modified at the end of each rotation interval. We refer to the FIFO queue that is tagged with index σ as the σ-queue. Upon arrival of a packet from a connection j with $j \in \mathcal{C}_p$, the packet is inserted into the current ρ_p-queue. Since $\rho_p > 0$ for all priorities, no packet arrival is inserted into the current

[4] The laxity of a packet stored in a queue is the remaining time until a deadline violation.

0-queue. The RPQ multiplexer always selects a packet from the non-empty σ-queue with the lowest index σ. 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 multiplexer rearranges the tagging of the FIFO queues. For each $\sigma \geq 1$, the current σ-queue will be relabeled as $(\sigma - 1)$-queue, and the current 0-queue becomes the new ρ_P-queue. Thus, the FIFO queues can be thought of as having performed a *"rotation"*. Queue rotations are performed independent of the presence of packets in the FIFO queues, that is, queues are rotated even if the RPQ multiplexer is empty. We assume that queue rotations are 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.

Next we illustrate the operations of the RPQ multiplexer in a simple example with three priority sets. The delay bounds for connections are given by Δ, 2Δ, and 3Δ for connections from priority sets 1, 2, and 3, respectively. As shown in Figure 1, the RPQ multiplexer for three priorities has four FIFO queues: one for each priority set, and one for the current 0-queue. Figure 1(a) shows an empty multiplexer at time 0^-. [5] 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 multiplexer through the circle shown in Figure 1(a).

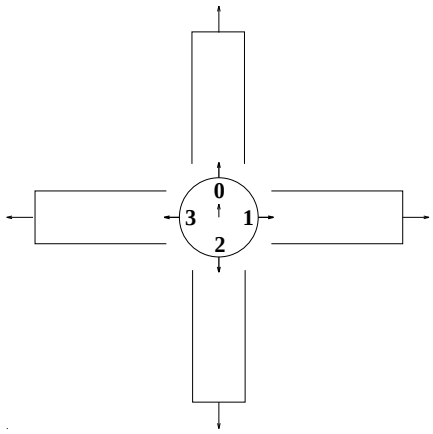
Assuming that packets start to arrive at time 0, Figure 1(b) shows a feasible snapshot of the FIFO queues at some time $0 \leq t < \Delta$. Here, we assume that the figure depicts a scenario at the end of the first rotation interval, at time Δ^-. In Figure 1(b), packets are shown as dark boxes and are labeled with their priority index. Since the 0-queue is empty, the packets in the 1-queue have highest priority.

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 following rotation interval.

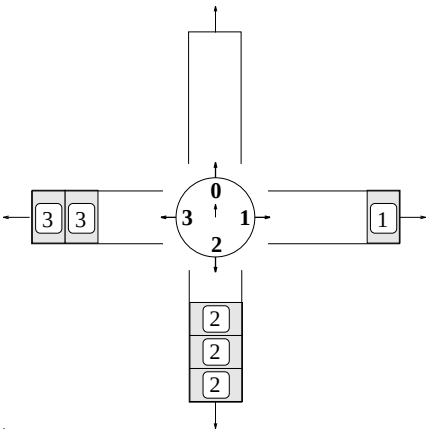
Figure 1(d) depicts a feasible scenario in the second rotation interval. For the sake of the presentation we assume that the scenario depicts the multiplexer at time $2\Delta^-$. Note that priority-p packets that arrived at the current p-queue may find packets from priority $(p+1)$ at the head of the queue.

In Figure 1(e) we show the result of the second queue rotation at time 2Δ. 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, by having the delay bounds 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

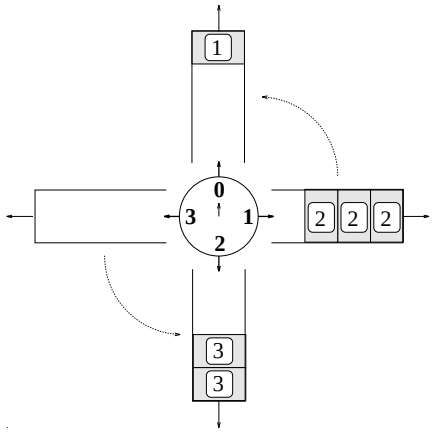
[5] t^- denotes the time immediately prior to time t.



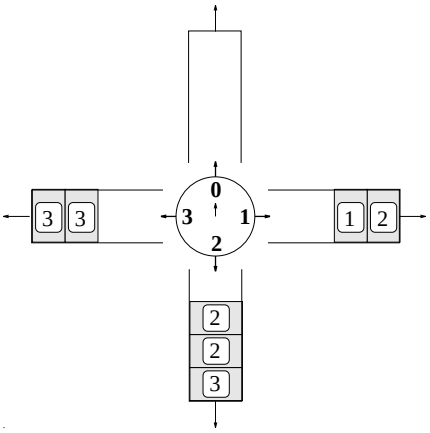
(a) RPQ multiplexer at time 0.



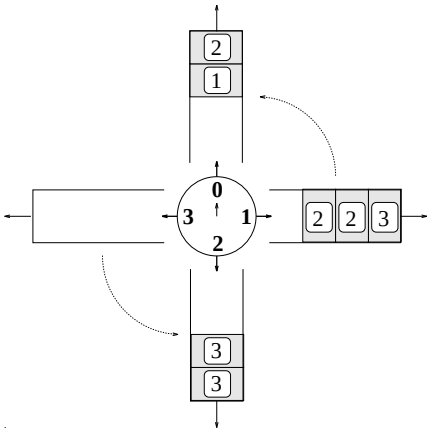
(b) RPQ multiplexer in time interval $[0, \Delta)$.



(c) RPQ multiplexer at time Δ.



(d) RPQ multiplexer in time interval $[\Delta, 2\Delta)$.



(e) RPQ multiplexer at time 2Δ.

Figure 1: Example of RPQ multiplexing.

that the 0-queue is empty at the end of each rotation interval.

From the example it becomes obvious that the queue rotation can be implemented by simply updating a set of pointers that indicate the position of each σ-queue. Thus, the additional complexity of RPQ multiplexing as compared to a Static-Priority (SP) multiplexer is low if the rotation interval is selected large. By selecting $\Delta = \infty$, i.e., queues are never rotated, an RPQ multiplexer is equivalent to an SP multiplexer.

We will show that by selecting the length of *rotation intervals* sufficiently small, the RPQ multiplexer closely approximates the efficiency of an EDF multiplexer. However, for small values of Δ, the number of FIFO queues needed by the RPQ multiplexer will grow. In Section 6 we show that, even for large values of Δ, the efficiency of an RPQ multiplexer is similar or even identical to the efficiency of an EDF multiplexer.

Note that RPQ multiplexing distinguishes itself from most multiplexing techniques in that knowledge of the schedulability conditions is required for a correct operation. Recall from the discussion of the example that we demand the 0-queue to be empty at the end of each rotation interval. By choosing the delay bound for connections from priority p to be equal to $\rho_p\Delta$, a packet that resides in the 0-queue at the end of a rotation interval must have a deadline violation. Thus, the requirements to have an empty 0-queue at the end of each rotation interval is a necessary condition for schedulability in an RPQ multiplexer.

5.2 Transmitted Workload before an Arbitrary Packet

In the following we derive tight, that is, necessary and sufficient, conditions for the schedulability of an RPQ multiplexer. Before we state the conditions in Subsection 5.3, we will derive an expression for the traffic workload that is transmitted before an arbitrary packet. The expressions help to obtain an intuitive understanding of the schedulability conditions.

In Figure 2 we show the arrivals of packets, indicated as arrows, at an RPQ multiplexer 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 example, we assume that $\rho_p = p$, 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.

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

$$\{(t - \tau_\Delta) + j\,\Delta \mid j \text{ an integer}\} \tag{6}$$

The shaded areas in Figure 2 indicate the time intervals for each priority during which a packet arrival from a given priority is transmitted before the tagged priority-p packet with arrival time t.

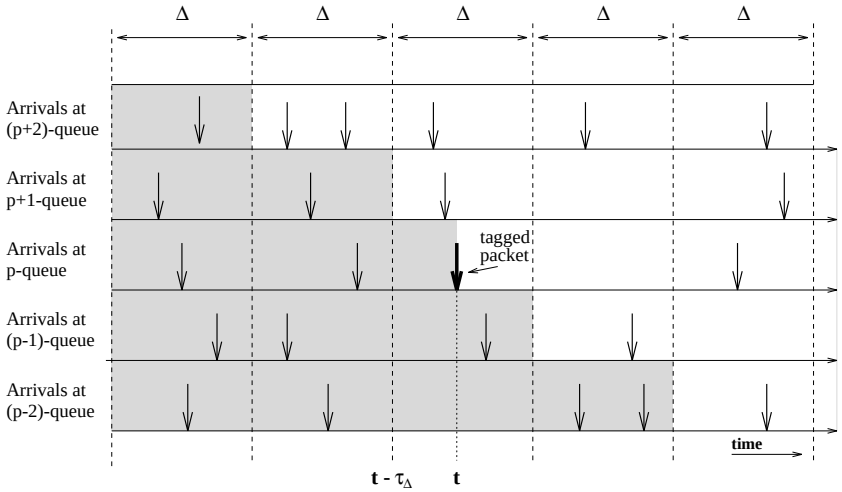


Figure 2: Workload served before a tagged packet in an RPQ multiplexer.

Since packets from connections in the same priority set are served in FIFO order, all arrivals from priority p that arrive before time t are served before the tagged packet. Packets from lower priority sets that are transmitted before the tagged packet are those packets that at time t reside in a ρ_q-queue with $\rho_q \leq \rho_p$. 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 $(\rho_p - 1)$-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 number of priority-$(p-2)$ packets that is served before the tagged packet is limited to packets that arrive before $t - \tau_\Delta + 2\Delta$, the end of the first rotation interval that begins after time t.

Let us ignore for the moment that the transmission of a packet cannot be interrupted, that is, assume that the RPQ multiplexer is preemptive. Further assume that the tagged packet has a transmission time of s_k^*, with $s_k^* \leq s_k$, and is completely transmitted at time $t + \delta$. Then we can generalize the example shown in Figure 2 and obtain time intervals for each priority q so that priority-q packets that arrive in the respective time interval are transmitted before the packet from connection k, with $k \in \mathcal{C}_p$, with arrival time t. The intervals for each priority set are as follows:

$$\begin{cases} [0, t - \tau_\Delta + (\rho_p - \rho_q + 1)\Delta] & \text{for all } q < p \\ [0, t] & \text{for } q = p \\ [0, \min\{t + \delta, t - \tau_\Delta + (\rho_p - \rho_q)\Delta\}] & \text{for all } q > p \end{cases} \tag{7}$$

Note that the given intervals are maximal if the arrival time t of the tagged packet occurs imme-

diately after a queue rotation, i.e., if $\tau_\Delta = 0$.

To accurately describe the workload served before the tagged priority-p packet with arrival time t, we need to account for the effects of nonpreemption. We define time $t - \hat{\tau}$ to be the last time prior to t at which the multiplexer is not transmitting packets that will be transmitted before the tagged packet. For priorities $q \leq p$, these are all the arrivals in time interval $[0, t - \hat{\tau}]$; for priorities $q > p$, all the arrivals in the interval $[0, \min\{t - \hat{\tau}, (t - \tau_\Delta) + (\rho_p - \rho_q + 1)\Delta)$. Denoting by $W_j(t)$ the workload in the RPQ multiplexer from connection $j \in \mathcal{N}$, we can determine $\hat{\tau}$ by:

$$\hat{\tau} = \min\{z \mid \sum_{q=1}^{P} \sum_{j \in \mathcal{C}_q} W_j(\min\{t - z,\, (t - \tau_\Delta) + (\rho_p - \rho_q + 1)\Delta\}\,) = 0, z \geq 0\} \tag{8}$$

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

- 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 $(\rho_q - \rho_p)$th rotating interval that ends *before* time t, that is, in time interval $[t - \hat{\tau}, (t - \tau_\Delta) + \rho_p\Delta - \rho_q\Delta + \Delta]$,
- The workload due to packets from all connections $j \in \mathcal{C}_q$ $(q < p)$ that arrive before time $t + \tau$ *and* before the $(\rho_p - \rho_q)$th rotating interval that ends *after* time t, or, equivalently, arrivals in the time interval $[t - \hat{\tau}, \min\{t + \tau,\ (t - \tau_\Delta) + \rho_p\Delta - \rho_q\Delta\}]$.
- Due to nonpreemption, 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 $[t - \hat{\tau}, t + \tau]$. By choice of $\hat{\tau}$, the total amount of workload served in this interval is $\hat{\tau} + \tau$.

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

$$\begin{aligned} W^{p,t}(t + \tau) \;=\; & \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j[t - \hat{\tau}, \min\{t + \tau, (t - \tau_\Delta) + \rho_p\Delta - \rho_q\Delta\}] + \sum_{j \in \mathcal{C}_p} A_j[t - \hat{\tau}, t] + \\ & + \sum_{q=p+1}^{P} \sum_{j \in \mathcal{C}_q} A_j[t - \hat{\tau}, (t - \tau_\Delta) + \rho_p\Delta - \rho_q\Delta + \Delta] + R(t - \hat{\tau}) - (t + \tau) \end{aligned} \tag{9}$$

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

$$\delta = s_k^* + \min\{z \mid W^{p,t}(t+z) = s_k^*,\ z \geq 0\} \tag{10}$$

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

5.3 Schedulability Conditions for the RPQ Multiplexer

We now present the schedulability conditions for RPQ multiplexers in Theorem 2. The conditions apply to arbitrary sets of connections with rate-controlled arrivals as defined in Subsection 3.1. We use s_p to denote the maximum transmission time of packets from a priority-p connection, i.e., $s_p = \max_{j \in \mathcal{C}_p} s_j$.

Theorem 2 *Given a set $\mathcal{N}$ of connections where each connection $j \in \mathcal{C}_p$ is characterized by (A_j^*, s_j, d_p), and given an RPQ multiplexer with rotation interval Δ such that, for each priority p, we have $d_p = \rho_p \Delta$. The set of connections is RPQ-schedulable for all $A_j \prec A_j^*$ if and only if for all $t \geq 0$:*

$$t \geq \sum_{j \in \mathcal{C}_1} A_j^*(t-d_1) + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t+\Delta-d_q) \tag{11}$$

and for all t with $d_1 \leq t < d_P - \Delta$:

$$t \geq \sum_{j \in \mathcal{C}_1} A_j^*(t-d_1) + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t+\Delta-d_q) + \max_{d_u > t+\Delta} s_u \tag{12}$$

The first condition in equation (11) is the schedulability condition for an preemptive RPQ multiplexer, and the second condition in equation (12) accounts for the fact that the multiplexer is nonpreemptive.

In the following corollary, we state that an RPQ multiplexer can be made to approximate the efficiency of an EDF multiplexer arbitrarily closely by appropriately selecting the length of the rotation interval Δ.

Corollary 1 *Given a set $\mathcal{N}$ of connections where each connection $j \in \mathcal{N}$ is characterized by (A_j^*, s_j, d_j) that is EDF-schedulable for all $A_j \prec A_j^*$, there exists a rotation interval Δ such that the connections are RPQ-schedulable.*

Corollary 1 is directly obtained by inspection of the conditions in Theorem 2 as $\Delta \to 0$.

Next we prove the correctness of Theorem 2. The proof is largely based on the previously derived expression for the workload served before a packet in equation (9). We first show the sufficiency of the conditions in equations (11) and (12). Following is the proof of necessity.

(a) Proof of Sufficiency

Assume that an arbitrary packet from connection k with $k \in \mathcal{C}_p$ arrives at the multiplexer at time t. We will show that conditions (11) and (12) guarantee that the packet does not have a deadline violation, i.e., that there exists a τ with $0 \leq \tau \leq d_p$ such that $W^{p,t}(t+\tau) = 0$, where $W^{p,t}(t+\tau)$ is given in equation (9).

Let $t - \tau_\Delta$ denote the rotation time immediately preceding t and let $t - \hat{\tau}$ be the last time that the multiplexer is not transmitting a packet that will be transmitted before the tagged packet from connection k, as obtained in equation (8). From the property of the rate-controlling functions A_j^* in equation (1), we state the following inequalities:

$$\sum_{j \in \mathcal{C}_p} A_j[t-\hat{\tau}, t] \leq \sum_{j \in \mathcal{C}_p} A_j^*(\hat{\tau}) \tag{13}$$

$$\begin{aligned} \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j[t-\hat{\tau}, (t-\tau_\Delta) + \rho_p\Delta - \rho_q\Delta] &\leq \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} - \tau_\Delta + \rho_p\Delta - \rho_q\Delta) \\ &\leq \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} + \rho_p\Delta - \rho_q\Delta) \end{aligned} \tag{14}$$

$$\begin{aligned} \sum_{q=p+1}^{P} \sum_{j \in \mathcal{C}_q} A_j[t-\hat{\tau}, (t-\tau_\Delta) + \Delta + \rho_p\Delta - \rho_q\Delta] &\leq \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*((\hat{\tau} - \tau_\Delta) + \Delta + \rho_p\Delta - \rho_q\Delta) \\ &\leq \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} + \Delta + \rho_p\Delta - \rho_q\Delta) \end{aligned} \tag{15}$$

Consider the workload served before our tagged packet at time $t+\rho_p\Delta$. We obtain from equation (9):

$$\begin{aligned} W^{p,t}(t+\rho_p\Delta) &= \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j[t-\hat{\tau}, (t-\tau_\Delta) + \rho_p\Delta - \rho_q\Delta] + \sum_{j \in \mathcal{C}_p} A_j[t-\hat{\tau}, t] + \\ &\quad + \sum_{q=p+1}^{P} \sum_{j \in \mathcal{C}_q} A_j[t-\hat{\tau}, (t-\tau_\Delta) + \Delta + \rho_p\Delta - \rho_q\Delta] + \\ &\quad + R(t-\hat{\tau}) - (\hat{\tau} + \rho_p\Delta) \end{aligned} \tag{16}$$

$$\begin{aligned} &\leq \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} + \rho_p\Delta - \rho_q\Delta) + \sum_{j \in \mathcal{C}_p} A_j^*(\hat{\tau}) + \\ &\quad + \sum_{q=p+1}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(\hat{\tau} + \Delta + \rho_p\Delta - \rho_q\Delta) + R(t-\hat{\tau}) - (\hat{\tau} + \rho_p\Delta) \end{aligned} \tag{17}$$

Observe that inequality (17) follows directly from equation (16) combined with equations (13), (14), and (15). Since the highest priority set (lowest index) with incoming traffic is $\mathcal{C}_1$, we can relax

and rewrite equation (17) as follows:

$$W^{p,t}(t+\rho_p\Delta) \leq \sum_{j\in\mathcal{C}_1} A_j^*(\hat{\tau}+\rho_p\Delta-\rho_1\Delta)+\sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(\hat{\tau}+\Delta+\rho_p\Delta-\rho_q\Delta)+R(t-\hat{\tau})-(\hat{\tau}+\rho_p\Delta) \quad (18)$$

We now consider the possible effects of nonpreemption. We turn to analyze whether or not there is any workload in the multiplexer at time $t-\hat{\tau}$.

Case 1: $W(t-\hat{\tau})=0$
Thus, there is no work in the multiplexer at time $t-\hat{\tau}$, so it must be the case that $R(t-\hat{\tau})=0$. We can combine equation (11) and equation (18), bounding the workload at time $t+\rho_p\Delta$. More formally,

$$W^{p,t}(t+\rho_p\Delta) \leq 0 \quad (19)$$

Thus, there exists a $\tau \leq t+\rho_p\Delta$ such that $W^{p,t}(t+\tau)\leq 0$, and the tagged packet will not violate its deadline.

Case 2: $W(t-\hat{\tau})>0$
Thus, the multiplexer is transmitting a packet at time $t-\hat{\tau}$ from some connection $l\in\mathcal{C}_u$. By selecting $\hat{\tau}$ as in equation (8), the delay bound of such a packet must exceed $\hat{\tau}+\rho_p\Delta+\Delta$. Since the maximum delay bound of all packets is given by $\rho_P\Delta$, observe that:

$$\hat{\tau}+\rho_p\Delta < \rho_P\Delta-\Delta \quad (20)$$

It follows that:

$$R(t-\hat{\tau}) \leq \max_{d_u>\hat{\tau}+d_p+\Delta} s_u \quad (21)$$

Combining equation (21) with our expression for the workload served before the tagged packet from equation (18) we obtain:

$$W^{p,t}(t+\rho_p\Delta) \leq \sum_{j\in\mathcal{C}_1} A_j^*(\hat{\tau}+\rho_p\Delta-\rho_1\Delta)+\sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(\hat{\tau}+\Delta+\rho_p\Delta-\rho_q\Delta)+ \\ + \max_{d_u>\hat{\tau}+\rho_p\Delta+\Delta} s_u-(\hat{\tau}+\rho_p\Delta) \quad (22)$$

Since $\hat{\tau}+\rho_p\Delta<\rho_P\Delta-\Delta$ and $\rho_p\geq\rho_1$, we can combine equation (12) and equation (22) to obtain

$$W^{p,t}(t+\rho_p\Delta)\leq 0 \quad (23)$$

Thus, our tagged packet will meet its deadline. □

(b) Proof of Necessity

We will construct a feasible pattern of packet arrivals at the RPQ multiplexer that will have a packet with a deadline violation if either equation (11) or equation (12) is violated.

Our proof will take advantage of the following observation. Assume that the RPQ multiplexer is empty at time 0^-, the workload served before a packet of the highest priority set $\mathcal{C}_1$, i.e., $W^{1,t}$, is bounded for all time $t > 0$ as follows:

$$W^{1,t}(t+\tau) \geq \sum_{j\in\mathcal{C}_1} A_j^*(t) + \sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(t - \tau_\Delta + \rho_1\Delta - \rho_q\Delta + \Delta) + R(0) - (t+\tau). \tag{24}$$

Since $W^{1,t}(t+\tau)$ is strictly decreasing over the interval $[t, t+\tau]$, a deadline violation will occur for some packet, if, for any time t,

$$W^{1,t}(t+\rho_1\Delta) > 0 \tag{25}$$

We will now show that if either equation (11) or equation (12) is violated, a deadline violation will occur due to equation (25).

First assume that the condition in (11) is violated at some time $\hat{t} > \rho_P\Delta - \Delta$, that is:

$$\hat{t} < \sum_{j\in\mathcal{C}_1} A_j^*(\hat{t} - \rho_1\Delta) + \sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(\hat{t} + \Delta - \rho_q\Delta) \tag{26}$$

Assume without loss of generality that time $\hat{t}$ will occur immediately after a priority rotation, and thus τ_Δ is small. Now, consider a scenario in which the multiplexer is empty at time 0^-, and starting at time 0 all connections j submit packets to the network according to A_j^*, with one exception: the last packet submitted to the network from a priority-1 connection before time $\hat{t} - \rho_1\Delta$ is submitted at time $\hat{t} - \rho_1\Delta$. In other words, the last packet arrival from connection $k \in \mathcal{C}_1$ before $\hat{t} - \rho_1\Delta - z$ is delayed until $\hat{t} - \rho_1\Delta - z$ where

$$z = \min\{y \mid A_k^*((\hat{t} - \rho_1\Delta - y)^-) < A_k^*(\hat{t} - \rho_1\Delta), y \geq 0\} \tag{27}$$

In the following, we refer to the delayed packet as the 'tagged packet.' Note that we can delay the submission of this packet as described above without violating the rate-controlling function A_k^*. With equation (24) and submissions at rate A^*, we find the total workload that must be served before our tagged packet from connection $k \in \mathcal{C}_1$:

$$W^{1,\hat{t}-\rho_1\Delta}(\hat{t}) \geq \sum_{j\in\mathcal{C}_1} A_j^*(\hat{t} - \rho_1\Delta) + \sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(t - \rho_q\Delta + \Delta) + R(0) - \hat{t} \tag{28}$$

Combining equation (28) with our assumption (26), we see that $W^{1,\hat{t}-\rho_1\Delta}(\hat{t}) > 0$. Thus, the tagged packet from connection k has a deadline violation by equation (25).

Next we assume that the condition in (12) is violated at some time $\hat{t}$ with $\rho_1\Delta \leq \hat{t} \leq \rho_P\Delta - \Delta$, that is:

$$\hat{t} < \sum_{j\in\mathcal{C}_1} A_j^*(\hat{t} - d_1) + \sum_{q=2}^{P}\sum_{j\in\mathcal{C}_q} A_j^*(\hat{t} + \Delta - d_q) + \max_{k\in\mathcal{C}_u, d_u > \hat{t}+\Delta} s_k \tag{29}$$

As before, assume that $\hat{t}$ occurs immediately after a queue rotation, and thus τ_Δ is small. Consider a similar scenario in which the multiplexer is empty before time 0^-, and at time 0^- a packet from connection k arrives with a transmission time of s_k, where k is a connection with maximal packet size among all connections that do not appear in the summation in equation (24), i.e., $s_k = \max_{d_u > \hat{t}+\Delta} s_u$. Also assume that, at time 0, all connections j, $j \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_{\lfloor \hat{t}/\Delta \rfloor + 1}$ submit packets to the multiplexer according to A_j^* with one exception: the last (tagged) packet submitted to the network from a priority-1 connection before time $\hat{t} - \rho_1\Delta$ from some connection $k \in \mathcal{C}_1$ is submitted at time $\hat{t} - \rho_1\Delta$.

With equation (9), the workload that is served before the tagged packet from connection $k \in \mathcal{C}_1$ at time $\hat{t} - \rho_1\Delta$ is given by:

$$W^{1,\hat{t}-\rho_1\Delta}(\hat{t}) = \sum_{j \in \mathcal{C}_1} A_j^*(\hat{t} - \rho_1\Delta) + \sum_{q=2}^{P} \sum_{j \in \mathcal{C}_q} A_j^*(t + \Delta - \rho_q\Delta) + \max_{k \in \mathcal{C}_u, d_u > \hat{t}+\Delta} s_k - \hat{t} \qquad (30)$$

Combining equation (30) with our assumption (29), we know that $W^{1,\hat{t}-\rho_1\Delta}(\hat{t}) > 0$. Thus, our tagged packet from connection l has a deadline violation by equation (25). □

6 Efficiency Comparison

In Section 5 we provided the necessary and sufficient schedulability conditions for the new RPQ packet multiplexer. However, the conditions alone provide little insight into the performance of RPQ multiplexing. Here, we present an empirical efficiency comparison of RPQ multiplexers with EDF and SP multiplexers. By varying the rotation interval Δ of the RPQ multiplexer we show that the efficiency of the RPQ multiplexer effectively approximates the efficiency of an EDF multiplexer. For the efficiency comparison, we use necessary and sufficient schedulability conditions for all considered multiplexers. The conditions are obtained from Theorem 1 for EDF multiplexing, from [15] for SP multiplexing, and from Theorem 2 for RPQ multiplexing.

For the sake of the presentation, we show the efficiency comparison for groups of connections rather than for individual connections. Thus, by selecting a small number number of only three connection groups, we can graphically illustrate the efficiency obtained by the respective multiplexers.

To describe the maximum traffic that can arrive to a multiplexer from connection group j we employ a simple traffic model that is defined by the parameter set (T_j, b_j, s_j). The traffic model is based on a variation of the leaky bucket traffic policing mechanism [21] and operates as follows. For each connection group j there exists a counter with maximum value b_j. Each time the connection group sends a packet to the multiplexer, the counter is decremented by one. Packets cannot be sent to the multiplexer if the counter is zero. The counter is incremented by one after each T_j time units if its value is less than b_j, and not incremented otherwise. We refer to T_j and b_j as the *period* and the *burst size* of the connection group, respectively, and s_j denotes the maximum transmission

	Group Index j	Delay Bound d_j	Max. Transmission Time per Packet s_j	Burst Size b_j	Period T_j
Low Delay Group	1	2 ms	200 μs	8 packets	0.5 – 2 ms
Medium Delay Group	2	4 ms	200 μs	9 packets	0.3 – 2 ms
High Delay Group	3	8 ms	200 μs	8 packets	2.5 – 10 ms

Table 1: Parameter Set for RPQ Multiplexer with 50 Mbps Transmission Rate.

time of a packet. With this traffic model, the rate-controlling function $A_j^*(t)$ for connection group j is given by:

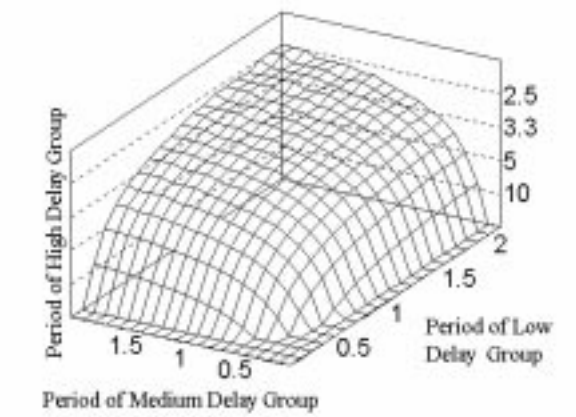
$$A_j^*(t) = b_j s_j + \left\lfloor \frac{t}{T_j} \right\rfloor s_j \tag{31}$$

We consider multiplexers that operate at 50 Mbps. The parameter sets for the connection groups are shown in Table 1. We have three connection groups referred to as low delay group, medium delay group, and high delay group. The delay bounds of packets are given by $d_1 = 2$ ms for the low delay group, $d_2 = 4$ ms for the medium delay group, and $d_3 = 8$ ms for the high delay group. For all connection groups, the maximum transmission time of a packet is set to 200 μs ($\approx$ 1250 Bytes), and the burst sizes are 8–9 packets per connection group. The periods of the connection groups are such that the maximum average data rate varies between 4–16 Mbps for the low delay group, 4–26 Mbps for the medium delay group, and 0.8–3.2 Mbps for the high delay group.

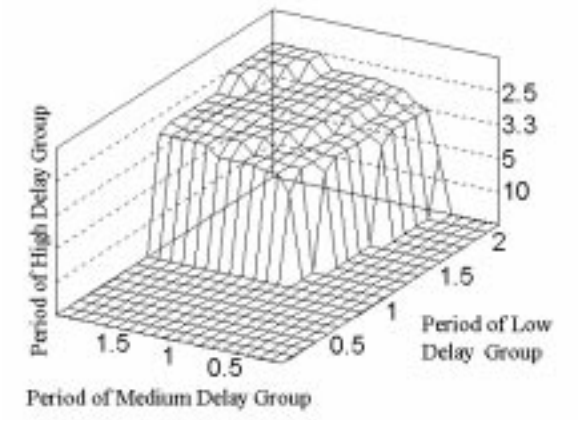
The results of the efficiency comparison for the given parameter set are graphically illustrated in Figures 3 and 4. Each graph shows a region of schedulability for a particular multiplexer when the periods of the connection groups are varied. The graphs, referred to as *schedulability graphs*, are interpreted as follows. The volume below the surface in each graph depicts the period values at which the connection groups are schedulable in the sense of Definition 2, i.e., no deadline violation occurs for any feasible traffic arrival sequence $\{A_j\}_{j=1,2,3}$ that conforms to the rate-controlling functions $\{A_j^*\}_{j=1,2,3}$ in equation (31) with $A_j \prec A_j^*$. The volume above the surface depicts parameter sets that are not schedulable in the worst case.

With the schedulability graphs we can directly compare the efficiency of two multiplexers Σ_1 and Σ_1 as follows. If the surface of a Σ_1 multiplexer completely covers the surface obtained for a Σ_2 multiplexer, then the Σ_1 multiplexer has a higher efficiency than the Σ_2 one.

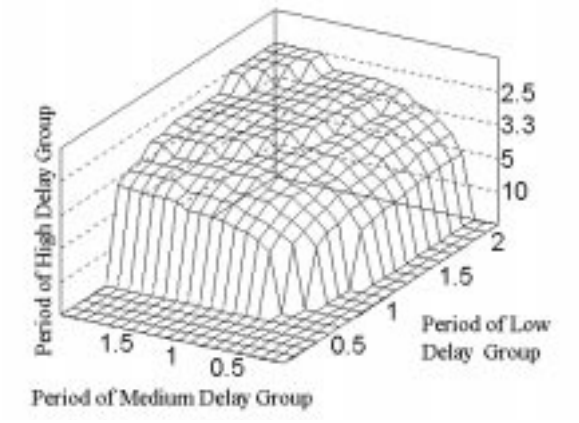
To evaluate the effects of deadlines in our parameter set, we show in Figure 3(a) the schedulability graph if packets do not have deadlines, i.e., when the delay bounds are set to $d_1 = d_2 = d_3 = \infty$. Since in this case the schedulability of the connection group is only bounded by the transmission



(a) Maximum Utilization.

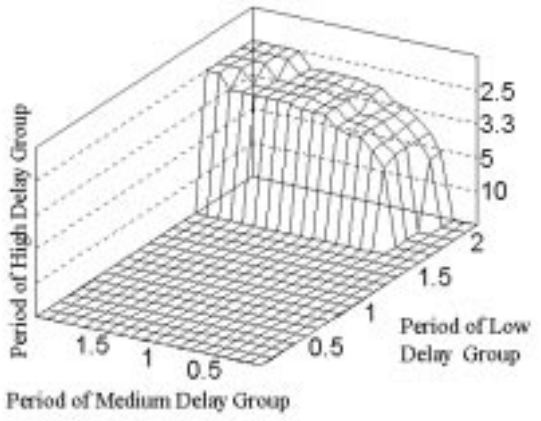


(b) SP Multiplexer.

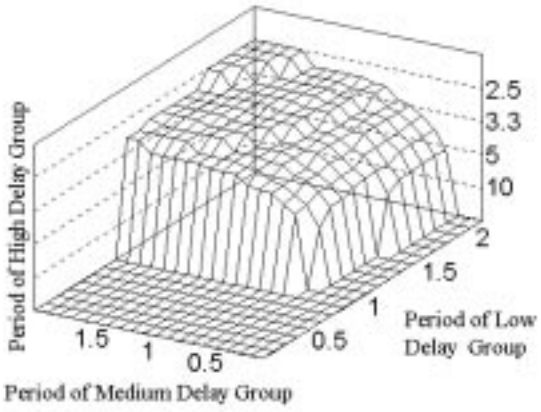


(c) EDF Multiplexer.

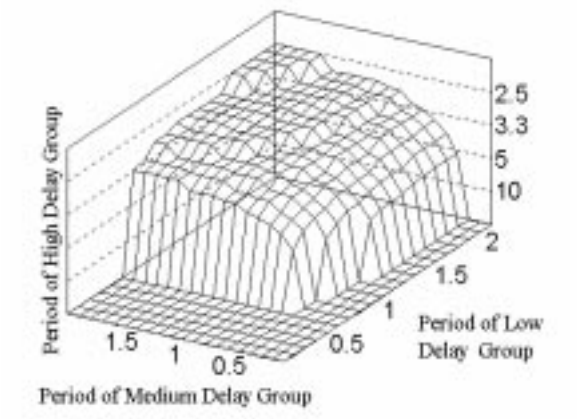
Figure 3: Schedulability Graphs (time values expressed in milliseconds).



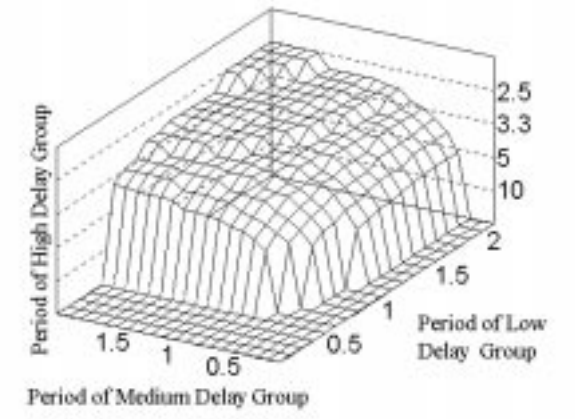
(a) RPQ Multiplexer ($\Delta = 0.5$ ms).



(b) RPQ Multiplexer ($\Delta = 0.4$ ms).



(c) RPQ Multiplexer ($\Delta = 0.2$ ms).



(d) RPQ Multiplexer ($\Delta = 0.05$ ms).

Figure 4: Schedulability Graphs for the RPQ Multiplexer (time values expressed in milliseconds).

speed of the multiplexer, the schedulability graph in Figure 3(a) has the largest surface of any multiplexer. In Figures 3(b) and 3(c) we illustrate the schedulability graph for the SP multiplexer[6] and the EDF multiplexer, respectively. We can clearly see that EDF is significantly more efficient than SP for our parameter set.

In Figures 4(a)–4(d) we show the graphs obtained for RPQ multiplexers with rotation intervals set at values from $\Delta = 0.5$ ms to $\Delta = 0.05$ ms. Here, the number of FIFO queues required by the RPQ multiplexer is given by $8/\Delta + 1$, where Δ is measured in milliseconds. In Figure 4(a) we see that, for $\Delta = 0.5$ ms, the efficiency of the RPQ multiplexer is below that of the SP scheduler shown in Figure 3(c). However, by decreasing the rotation interval by 0.1 ms to $\Delta = 0.4$ ms, we observe in Figure 4(b) that RPQ is superior to SP. If the rotation interval is further decreased, then the efficiency of RPQ quickly approaches the efficiency of EDF multiplexing. By comparing Figure 3(b) with Figures 4(c)–4(d) we can see that, for the chosen parameter set, the efficiency of RPQ as compared to that of EDF is almost identical for $\Delta = 0.2$ ms, and fully identical for $\Delta = 0.05$ ms.

7 Conclusions

The performance of a bounded delay service in a packet switching networks is largely determined by the selection of the packet multiplexer at the network switches which establishes the order of packet transmissions. All previously proposed multiplexing techniques either support only a limited number of connections with delay bound constraints, e.g., a Static Priority (SP), or require a complex implementation which may prevent their use in high-speed networks, e.g., Earliest Deadline First (EDF). We have proposed a novel multiplexing technique for bounded delay services, called Rotating Priority Queues (RPQ), which exploits the tradeoff between simple implementation and high efficiency. The RPQ multiplexer was shown to be implementable with a number of FIFO queues which are *'rotated'* after fixed time intervals. Since the queue rotations can be implemented by merely updating a set of pointers, the RPQ multiplexer does not incur significant computational overhead as compared to an SP multiplexer. We showed that, by properly decreasing the time between queue rotations, the efficiency of the RPQ multiplexer closely approximates the efficiency of an EDF multiplexer. We have presented necessary and sufficient schedulability conditions for the RPQ multiplexer. Knowledge of the schedulability conditions enables the detection of possible deadline violations of packets, and hence, is a requirement for admission control tests in networks that offer a bounded delay service. We used examples to compare the efficiency of the RPQ multiplexer with the efficiency of EDF and SP multiplexers. The examples illustrated that the RPQ multiplexer introduces a significant efficiency gain as compared to an SP multiplexer, and has

[6]For the SP multiplexer, the priorities are assigned so that the priority of a connection group is higher if the delay bound of the connection group is smaller.

a similar or identical efficiency as an EDF multiplexer even if the time between queue rotations is relatively long.

Acknowledgment

The authors are grateful to Domenico Ferrari for his many helpful comments and suggestions.

References

[1] C.-S. Chang. Stability, Queue Length and Delay, Part I: Deterministic Queueing Networks. Technical Report RC 17708, IBM Research Division, Yorktown Heights, February 1992.

[2] C.-S. Chang. Stability, Queue Length and Delay, Part II: Stochastic Queueing Networks. Technical Report RC 17709, IBM Research Division, Yorktown Heights, February 1992.

[3] D. D. Clark, S. Shenker, and L. Zhang. Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms. In *Proc. Sigcomm '92*, pages 14–26, August 1992.

[4] 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.

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

[6] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proc. Sigcomm '89*, pages 1–12, 1989.

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

[8] 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.

[9] S. J. Golestani. A Stop-and-Go Queueing Framework for Congestion Management. In *ACM Sigcomm '90*, pages 8–18, September 1990.

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

[11] R.-H. Hwang, J. F. Kurose, and D. Towsley. The Effect of Processing in High Speed Networks. In *Proc. Inforcom '92*, pages 160–169, April 1992.

[12] 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.

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

[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] J. Liebeherr, D. E. Wrege, and Domenico Ferrari. Exact Admission Control in Networks with Bounded Delay Services. Technical report, Department of Computer Science, Universiy of Virginia, July 1994.

[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] R. Nagarajan and J. F. Kurose. On Defining, Computing and Guaranteeing Quality-of-Service in High-Speed Networks. In *Proc. Inforcom '92*, pages 2016–2025, Florence, Italy, April 1992.

[18] A. K. J. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, February 1992.

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

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

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

[22] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing Delay Jitter Bounds in Packet Switching Networks. In *Proc. Tricomm '91*, Chapel Hill, North Carolina, April 1991.

[23] D. C. Verma. *Guaranteed Performance Communication in High Speed Networks*. PhD thesis, University of California - Berkeley, November 1991.

[24] O. Yaron and M. Sidi. Calculating Performance Bounds in Communication Networks. In *Proc. IEEE Infocom '93*, number CS-94-29, pages 539–546, April 1993.

[25] D. Yates, J.F. Kurose, D. Towsley, and M. G. Hluchyj. On Per-Session End-to-End Delay Distributions and the Call Admission Problem for Real-Time Applications with QOS Requirements. In *Proc. Sigcomm '93*, pages 13–19, September 1993.

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

[27] H. Zhang and D. Ferrari. Rate-Controlled Static-Priority Queueing. In *Proc. IEEE Infocom '93*, pages 227–236, April 1993.

[28] L. Zhang. *A New Architecture for Packet Switched Networks*. PhD thesis, Massachusetts Institute of Technology, July 1989.

[29] Q. Z. Zheng and K. G. Shin. On the Ability of Establishing Real-Time Channels in Point-to-Point Packet Switched Networks. to appear: IEEE Transactions on Communications.

A Traffic Characterizations

In this study, we have used a very general specification of rate-controlled traffic to a multiplexer (Section 3). Our only assumption on the packet arrivals is the existence of a rate-controlling function A_j^* which characterizes the worst case traffic of a connection j. To enforce equation (1) for all actual arrival functions A_j, policing mechanisms must be implemented at the boundary of the network or in the network switches. The choice of a particular rate-controlling function A_j^* depends on the complexity of the implemented policing functions. Here we present the rate-controlling functions A_j^* for different traffic characterizations considered in the literature.

1. The (ρ_j, σ_j)-model [4] describes traffic from a connection j in terms of a traffic rate ρ_j and a burstiness factor σ_j. Here, A_j^* is given by:

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

2. In the (σ_j, ρ_j, C_j)-model [4, 18] the traffic is characterized by a rate factor σ_j and a burst factor ρ_j which is additionally constrained by the maximum transmission rate of the network link C_j. In this case the rate-controlling function is given by:

$$A_j^*(t) = \min\{C_j\, t, \sigma_j + \rho_j t\}$$

Both the (ρ, σ)-model and the (σ, ρ, C)-model are continuous traffic characterizations, which ignore that data is transmitted as packets of nonzero length. Hence, these traffic characterizations can only approximately describe the observed worst case traffic in a network. More realistic discrete traffic models include a parameter s_j, to denote the maximum transmission time for a packet from a connection j.

3. The (r_j, T_j, s_j)-model [10] specifies the rate-controlling function by an average packet rate r_j averaged over a time period T_j. The expression for the rate-controlling function A_j^* is given as follows:

$$A_j^*(t) = \left\lceil \frac{t}{T_j} \right\rceil \frac{r_j\, T_j}{s_j}$$

4. In the $(x_{min,j}, x_{ave,j}, I_j, s_j)$-model [8], $x_{min,j}$ specifies the minimum time interval between any two packets from a connection, and $x_{ave,j}$ denotes the minimum average interarrival time of packets averaged over a time interval I_j. We obtain the following rate-controlling function:

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