A Versatile Packet Multiplexer for Quality-of-Service Networks*

Jörg Liebeherr *Dallas E. Wrege*
Department of Computer Science
University of Virginia
Charlottesville, VA 22903

Abstract

A novel packet multiplexing technique, called Rotating-Priority-Queues (RPQ), is presented which exploits the tradeoff between high 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 presented and compared to expressions for an EDF multiplexer.

1 Introduction

A major challenge in the design of multiservice networks that support transmission of data, voice, and video 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 tests* which limit the number of connections and *policing mechanisms* which monitor the traffic on each connection, a large number of packet multiplexers can provide bounds on delays [4, 10, 14, 16]; 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:

- *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. For example, a FIFO multiplexer can support only one delay bound for all connections and thus has 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 require 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 and reduce 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 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 [4, 16].

EDF multiplexers which always select the packet with the shortest time until a delay bound violation for transmission are optimal with respect to efficiency

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

and flexibility [5]. A disadvantage of EDF multiplexing is that packets in the multiplexer queue must be sorted according to their deadlines, introducing a considerable degree of complexity. Delay-EDD [4] is a multi-class version of EDF that supports connections with both deterministic and probabilistic delay guarantees. Jitter-EDD [13] extends Delay-EDD by a holding mechanism for packets and can also provide bounds on network delay variations, i.e., delay jitter. Ferrari and Verma show sufficient schedulability conditions for EDF multiplexers in [4]. Necessary and sufficient conditions for EDF are derived in [5, 6, 17] for specific policing mechanisms, and in [8] for general policing methods.

SP multiplexers support 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 [8]. 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. Zhang proves sufficient schedulability conditions for SP multiplexers in [16]. Using a fluid flow traffic approximation, Cruz has shown necessary and sufficient conditions [2]. For a general class of policing mechanisms, necessary and sufficient conditions are proven in [8].

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 close to SP. RPQ is implemented with a set of ordered FIFO queues, similar to SP. Different from SP, the ordering 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 the rotation intervals sufficiently small, RPQ can approximate the efficiency of EDF arbitrarily closely.

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 [8] 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 §2 we discuss a general traffic and multiplexer model. In §3 we present the schedulability conditions of EDF multiplexers as derived in [8]. In §4 we present the novel RPQ packet multiplexer and its necessary and sufficient schedulability conditions. In §5 we compare the efficiency of the EDF, SP, and RPQ multiplexing techniques with empirical examples. We conclude our study in §6.

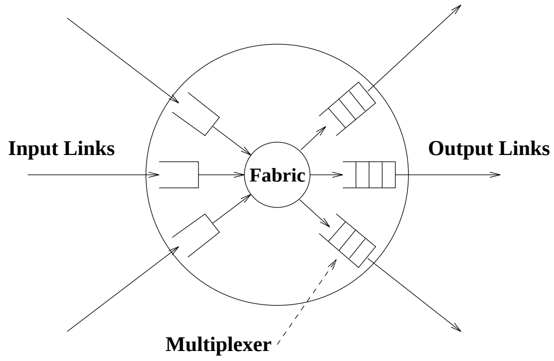


Figure 1: Network switch components.

2 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. Figure 1 depicts the components of a single network switch. At each switch there is one packet multiplexer for every 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 [3], or by providing a holding mechanisms at each switch as shown in [15].

Next we provide a description of packet multiplexers for networks with a bounded delay service. In §2.1 we present a general traffic characterization for packet arrivals at a multiplexer. In §2.2 we discuss the properties of packet multiplexers and formally define both schedulability conditions and admission control tests.

2.1 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 $[t, t+\tau]$. 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 *traffic constraint function* A_j^* and by s_j, the maximum transmission time of any packet from connection j. The traffic constraint function A_j^*

is used to describe the maximum traffic arrivals from connection j in any time interval. 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, 2]:

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

If equation (1) holds, we say that A_j is *constrained* by A_j^*, and we write $A_j \prec A_j^*$. ($A_j^*(t) = 0$ and $A_j(t) = 0$ for all $t < 0$.)

The maximum traffic arrivals from a connection j is observed if packets arrive according to the traffic constraint functions, i.e., $A_j = A_j^*$.

The specification of constrained traffic given in equation (1) is very general. Our only assumption on the packet arrivals is the existence of a traffic constraint 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.

2.2 Packet Transmissions and Schedulability Conditions

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*. 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.

Each connection j with traffic to the multiplexer has a *delay bound* d_j that indicates the maximum tolerable delay (including queueing and transmission delays) of any packet from connection j in the multiplexer. 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 constrained connections is *schedulable* if no deadline violation occurs for all feasible arrival functions $\{A_j\}_{j \in \mathcal{N}}$ which conform to equation (1).

The conditions which determine if a set of connections is *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: A new connection k is said to be *admissible* if the set of connections $\mathcal{N} \cup \{k\}$ is also schedulable. 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.

3 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 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 multiplexer queue of an EDF multiplexer must be sorted according to the packet deadlines, each packet arrival involves a search operation to find the correct position of the newly arrived packet in the multiplexer queue.

In the next theorem we present tight schedulability conditions for an EDF multiplexer with the general set of constrained arrival functions defined in §2.1. A proof of Theorem 1 is presented in [8]. In §4, 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 $d_i < d_j$, and we use s_k to denote the maximum transmission time for any packet from connection k.

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 $t \geq d_1$:*

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

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 they can support a bounded delay service with efficiency close to that of an EDF multiplexer.

4 The Rotating-Priority-Queues Multiplexer

The Rotating-Priority-Queues (RPQ) multiplexer attempts to approximate the EDF multiplexer without maintaining a sorted queue. Similar to the Static-Priority (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 RPQ multiplexer can achieve a utilization of network resources that is arbitrarily close to that of an EDF multiplexer.

Approximations of EDF multiplexers with a set of ordered FIFO queues have been considered before [9, 11]; however, not in the context of bounded delay services. The Head-of-Line with Priority Jumps (HOL-PJ) multiplexer proposed by Lim and Kobza [9] assigns each FIFO queue a range of laxity values, where the laxity of a packet stored in a queue 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. In another approach [11], 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 [11], 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. We then present necessary and sufficient schedulability conditions for an RPQ multiplexer.

4.1 Description of the RPQ Multiplexer

The RPQ multiplexer depends upon a so-called *rotation interval* Δ, where $\Delta > 0$. The choice of Δ impacts the operations of the RPQ multiplexer and determines the delay bounds that it supports. All delay bounds in RPQ are required to be multiples of the rotation interval Δ, and thus Δ determines the granularity with which RPQ approximates EDF.

Connections with traffic to the RPQ multiplexer are partitioned into P disjoint priority sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P$, where the delay bound of each priority set is a multiple of the fixed rotation interval Δ. Packets from a connection in priority set $\mathcal{C}_p$ have delay bound $d_p = \rho_p \Delta$, where ρ is a positive integer with $\rho_p < \rho_q$ if $p < q$ and $\rho_1 > 0$. Traffic that arrives at the multiplexer from a connection j is limited by a traffic constraint function A_j^*.

The RPQ multiplexer maintains $\rho_P + 1$ ordered FIFO queues. At all times, each FIFO queue is tagged with an integral index σ where $0 \leq \sigma \leq \rho_P$; however, the taggings of the FIFO queues are 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 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 2, the RPQ multiplexer for three priorities has four FIFO queues: one for each priority set, and one for the current 0-queue. Arriving priority-p packets are thought to enter the RPQ multiplexer through the circle as shown in Figure 2(a). The taggings of the four FIFO queues are indicated by the labels in the circle. 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.

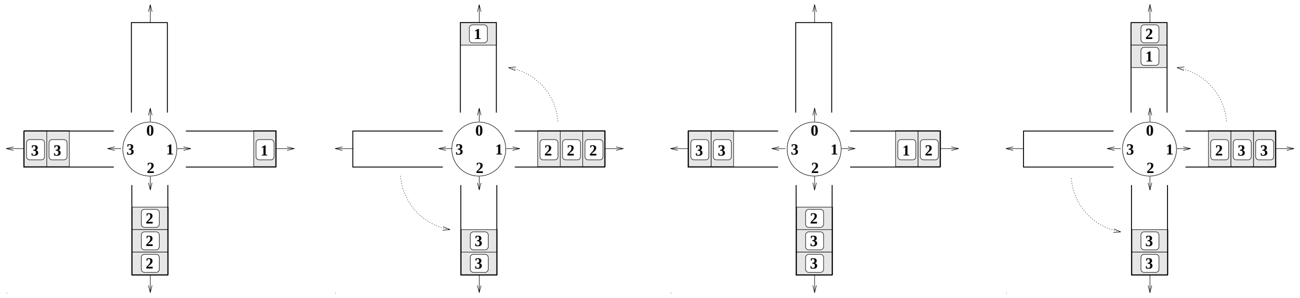
Assuming that packets start to arrive at time 0, Figure 2(a) shows a feasible snapshot of the FIFO queues at some time $0 \leq t < \Delta$. Here, we assume that the figure depicts a scenario at the end of the first rotation interval, at time Δ^-. (t^- denotes the time immediately prior to time t.) In Figure 2(a), 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 2(b) 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 2(b). Since the (former) 1-queue now becomes the 0-queue, no packets will arrive to this queue during the following rotation interval.

Figure 2(c) depicts a feasible scenario in the second rotation interval, at time $2\Delta^-$. Note that priority-p packets arriving at the current p-queue may find packets from priority $(p+1)$ at the head of the queue.

In Figure 2(d) we show the result of the second queue rotation at time 2Δ. Note that in order to perform the rotation, we require the 0-queue to be 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 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 updat-



(a) Time interval $[0, \Delta)$. (b) Time Δ. (c) Time interval $[\Delta, 2\Delta)$. (d) Time 2Δ.

Figure 2: Example of RPQ multiplexing.

	Group Index j	Delay Bound d_j	Packet Size s_j	Burst Size b_j	Period T_j	Maximum Average Rate
Low Delay Group	1	2 ms	1250 Bytes	8 packets	0.5 – 2 ms	$\approx$ 4–16 Mbps
Medium Delay Group	2	4 ms	1250 Bytes	9 packets	0.3 – 2 ms	$\approx$ 4–26 Mbps
High Delay Group	3	8 ms	1250 Bytes	8 packets	2.5 – 10 ms	$\approx$ 0.8–3.2 Mbps

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

ing a set of pointers that indicate the position of each σ-queue. Thus, the additional complexity of RPQ multiplexing as compared to a 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 the rotation interval Δ 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.

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 of 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.

4.2 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 constrained arrivals as defined in §2.1. We use s_u to denote the maximum transmission time of any packet from a priority-u connection, i.e., $s_u = \max_{j \in \mathcal{C}_u} s_j$.

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 $t \geq d_1$:*

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

A proof of Theorem 2 is given in [7].

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

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.*

5 Numerical Examples

In §4 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 [8] 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 of only three connection groups, we can graphically illustrate the efficiency obtained by the respective multiplexers.

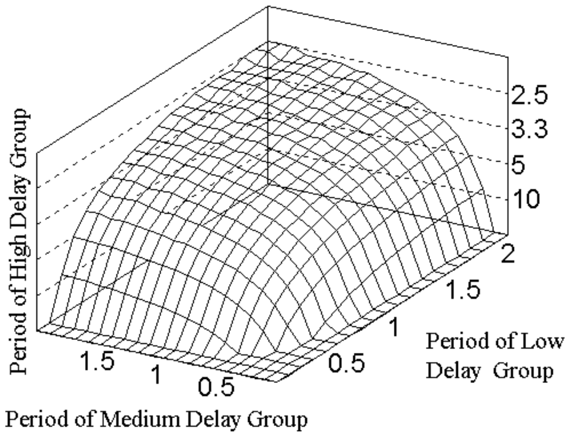
To describe the maximum traffic that can arrive to a multiplexer from connection group j we employ a simple traffic characterization that is defined by the parameter set (T_j, b_j, s_j). The traffic model is based on a variation of the leaky bucket traffic policing mechanism [12] and operates as follows. 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 it is 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 packet size. With this traffic model, the traffic constraint function $A_j^*(t)$ for connection group j is given by:

$$A_j^*(t) = b_j s_j + \left\lfloor \frac{t}{T_j} \right\rfloor s_j \qquad (2)$$

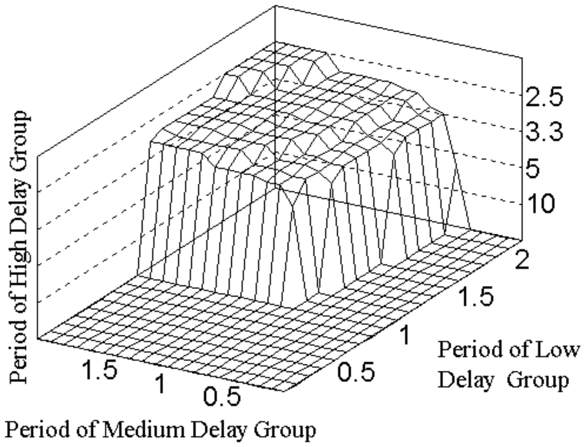
We consider a multiplexer that operates at 50 Mbps. The parameter sets for the connection groups are shown in Table 1. We have three connection groups referred to as low delay group, medium delay group, and high delay group. The delay bounds of packets are given by $d_1 = 2$ ms for the low delay group, $d_2 = 4$ ms for the medium delay group, and $d_3 = 8$ ms for the high delay group. For all connection groups, the maximum packet size is assumed to be 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; no deadline violation occurs for any feasible traffic arrival sequence $\{A_j\}_{j=1,2,3}$ that conforms to the traffic constraint functions $\{A_j^*\}_{j=1,2,3}$ in equation (2) with $A_j \prec A_j^*$. The volume above the surface depicts parameter sets that are not schedulable in the worst case.

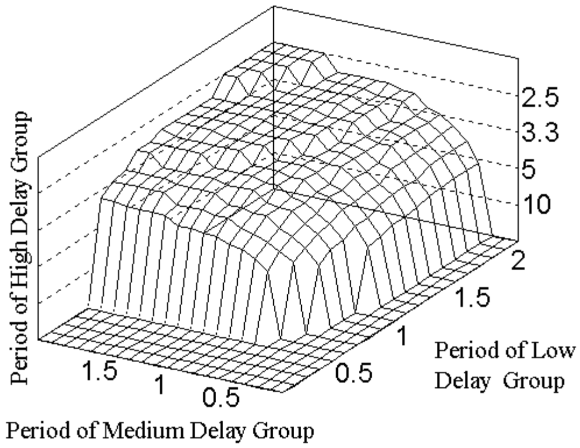
With the schedulability graphs we can directly compare the efficiency of two multiplexers Σ_1 and Σ_2 as



(a) Maximum Utilization.

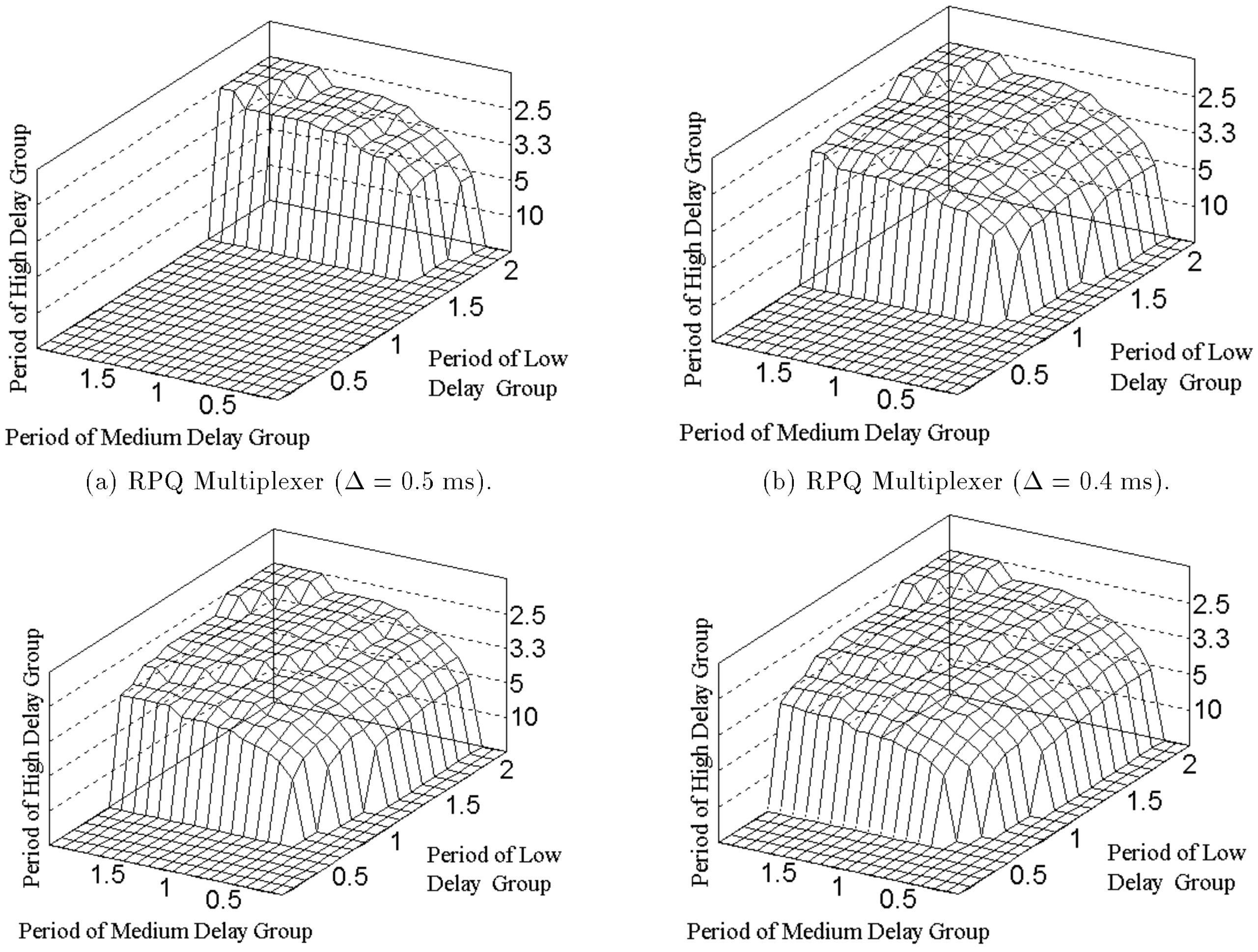


(b) SP Multiplexer.



(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).

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 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[1] and the EDF multiplexer, respectively. We can clearly see that EDF admits more traffic 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 multiplexer 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.

[1]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.

6 Conclusions

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. We used examples to compare the achievable utilizations of the RPQ, EDF, and SP multiplexers. The examples illustrated that the RPQ multiplexer introduces a significant efficiency gain as compared to an SP multiplexer, and it has an efficiency that is similar or identical to an EDF multiplexer even if the time between queue rotations is relatively long.

Acknowledgement

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

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] S. Guo and N. D. Georganas. Resource and Connection Admission Control in Real-Time Transport Protocols with Deterministic QoS Guarantees. In *Proc. IEEE Infocom '95*, pages 1095–1102, April 1995.

[7] J. Liebeherr and D. E. Wrege. High-Performance Multiplexing in Multiservice Networks with Delay Guarantees. Technical Report CS-94-30, Department of Computer Science, Universiy of Virginia, July 1994.

[8] J. Liebeherr, D. E. Wrege, and Domenico Ferrari. Admission Control in Networks with Bounded Delay Services. Technical Report CS-94-29, Department of Computer Science, Universiy of Virginia, July 1994.

[9] 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.

[10] 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.

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

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

[13] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing Delay Jitter Bounds in Packet Switching Networks. In *Proc. Tricomm '91*, pages 35–46, April 1991.

[14] G. G. Xie and S. S. Lam. Delay Guarantee of Virtual Clock Server. Technical Report TR-94-24, University of Texas at Austin, October 1994. Presented at *9th IEEE Workshop on Computer Communications*, Marathon, Florida, October 1994.

[15] H. Zhang. Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines. to appear: *Computer Communications*: Special Issue on System Support for Multimedia Computing.

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

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