Optimal Multiplexing on a Single Link: Delay and Buffer Requirements

Leonidas Georgiadis *
Aristotle University of Thessaloniki
Dept. of Electrical and Computer Eng.
P.O. Box 435
Thessaloniki, 54006 - GREECE

Roch Guérin
IBM T. J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

Abhay Parekh †
Sun Microsystems Inc.
2550 Garcia Avenue
Mountain View, CA 94043-1100

Abstract. This paper is motivated by the need to provide per session quality of service guarantees in fast packet-switched networks. We address the problem of characterizing and designing scheduling policies that are optimal in the sense of minimizing buffer and/or delay requirements under the assumption of commonly accepted traffic constraints. We investigate buffer requirements under three typical memory allocation mechanisms which represent trade-offs between efficiency and complexity. For traffic with delay constraints we provide policies that are optimal in the sense of satisfying the constraints if they are satisfiable by any policy. We also investigate the trade-off between delay and buffer optimality, and design policies that are "good" (optimal or close to) for both. Finally, we extend our results to the case of "soft" delay constraints and address the issue of designing policies that satisfy such constraints in a fair manner. Given our focus on packet switching, we mainly concern ourselves with non-preemptive policies, but one class of non-preemptive policies which we consider is based on tracking preemptive policies. This class is introduced in this paper and may be of interest in other applications as well.

Key Words: Data Networks, Scheduling, Multiplexing, Optimization, Buffer Allocation, Schedulable Regions.

1 Introduction

A key challenge in the design of integrated services networks is to support a large number of sessions with different performance requirements, while minimizing cost as measured by network resources. Session performance is mainly characterized by packet delay and loss probability, with

*This work was done while the author was at the IBM T. J. Watson Research Center.
†This work was done while the author was at the IBM T. J. Watson Research Center.

link bandwidth and buffer space being the network resources that must be expended to achieve performance.

It is clear that buffer requirements and delay are intimately related, since delay is trivially bounded above by the amount of time it takes to drain a switch with full buffers. Yet, there are more intricate factors at work when the switch implements scheduling and buffer allocation policies which discriminate among the sessions. The scheduling policy (usually implemented at the output ports of the switch) determines the order in which queued packets are served, and the buffer allocation policy determines the manner in which the buffer space is to be shared among the sessions. It turns out that for a given requirement on the loss probabilities, the choice of scheduling policy has an effect on both the delay and the total amount of buffer space required [15, 20, 25], while the choice of buffer allocation policy has an effect only on the total amount of buffer space required. To make things more complicated, for a given scheduling policy, the total amount of buffer space required is also dependent on the buffer allocation policy. These dependencies are not negligible and need to be examined carefully.

A central contribution of this paper is, therefore, to define a simple analytical model that permits meaningful evaluations of the delay and buffer requirements of policies, so that they can be properly compared. We then find policies that are optimal within this analytical model.

Our study is restricted to the case of a single link (multiplexer) and assumes a *zero-loss* environment, i.e., buffers are sized so that space is always available to store incoming data, provided the input traffic satisfies certain constraints. Our choice of zero-loss is motivated by several considerations. First, it provides us with a common basis of comparison for how each policy handles various traffic patterns. Second, it clearly represents a desirable feature, irrespective of whether an application can tolerate some losses, and we want to emphasize that providing such guarantees is indeed feasible at a reasonable cost. The traffic constraints we assume in order to be able to ensure zero-loss, are well-accepted and in-line with the requirements of standard rate control algorithms [2]. Specifically, we assume that each session has a given average rate ρ_i, an associated maximum burstiness σ_i (see Section 2.1 for a more rigorous definition), and a maximum packet size $L_{\max}$.

A basic, qualitative outline of the paper is the following: In Section 2, we define our model, introduce the scheduling policies (including a new class of policies known as Tracking policies) we are going to be using in the rest of the paper, and give a few preliminary results. In Section 3 we examine various buffer allocation policies and for each, show specific scheduling policies to be buffer-optimal, i.e., they require the minimal possible amount of buffer to ensure zero-loss, over all scheduling policies. Section 4 considers the corresponding problem of designing delay optimal scheduling disciplines, and among the class of delay optimal policies identifies those that result in low buffer occupancy as well. Here we find that the more flexible the buffer allocation policy, the lower the buffer requirements for the "best" delay optimal policy. In the last major section of the paper, Section 5, we define delay requirements differently, in that we allow packets to miss their deadlines, and design policies in which the "lateness" is distributed fairly among the sessions.

1.1 Results

In Section 3, we study three buffer structures, Flexible, Semi-flexible and Fixed, that represent different trade-offs between efficiency and complexity, and design buffer optimal policies for each. The analysis shows the surprising result that improving the complexity of the buffer structure may not improve the efficiency significantly.

However, the advantage of a more complex and, therefore, more flexible buffer allocation scheme becomes apparent in Section 4, where we show that the added flexibility results in significant advantages when delays also need to be optimized. In this section, we identify the schedulable region of the multiplexer and characterize delay optimal policies, i.e., those with maximal schedulable region. In keeping with our focus on packet switching, we design non-preemptive policies, as opposed to preemptive ones. We show that both a standard Non-Preemptive Earliest Deadline First policy ($NPEDF$) and a Tracking policy based on the Preemptive Earliest Deadline First policy ($T(PEDF)$) are delay-optimal among the class of non-preemptive policies. Based on our knowledge of policies which are optimal for either buffer or delay requirements, we proceed next in Section 4.2 with a policy which is delay-optimal and has small (near optimal) buffer requirements.

In Section 5, we consider two separate figures of merit. The first is that of minimizing the maximum lateness over all packets under any arrival pattern. We establish that under $NPEDF$ and $T(PEDF)$ maximum lateness is no more than $L_{\max}/r$ time units greater than what it is under $PEDF$, which is known to be optimal with respect to minimizing maximum lateness [11], where $L_{\max}/r$ is the transmission time of a maximum size packet over a link of speed r. Thus $NPEDF$ and $T(PEDF)$ are very close to being optimal. The second, and stronger figure of merit is the degree to which the packet lateness vector is close to being *lexicographically* minimal. We show that a particular version of $PEDF$, which we call $PEDF^*$ is lexicographically optimal among all preemptive policies. Further, we show that the tracking policy, $T(PEDF^*)$, is close to being lexicographically optimal in that under $T(PEDF^*)$ no packet is delayed by more than $L_{\max}/r$ beyond what it experiences under $PEDF^*$.

1.2 Earlier Work

Buffer-optimal policies under the fixed allocation method have been studied in [22, 7, 4, 13, 5] and the buffer optimal policy for $\sigma_i = 0$ was presented in [21]. While the case of the flexible allocation method is straightforward, our results for the semi-flexible allocation case are new. In addition, our result linking the schedulable region and buffer requirements under fixed allocation is new, as is our result on how to construct delay-optimal policies that have small buffer requirements.

The problem of scheduling tasks has received significant attention in the context of (real-time) computing systems, where important results on optimal scheduling policies and the associated schedulable region have been obtained. However many of these results assume more restrictive arrival patterns than those used in this paper: The optimality of the $PEDF$ for the class of preemptive policies was first shown in [19] for periodic arrivals and in [11] for general arrival patterns; in [16, 17] the delay-optimality of $NPEDF$ among the class of non-preemptive policies is established for periodic and so-called sporadic arrivals; the schedulable regions for $NPEDF$ and $PEDF$ have been derived in [26] for arrival streams characterized by a minimum inter-packet arrival time that is independent of packet size. The merit of using schedulable regions to guarantee quality of service in networks was recognized in [18]. The $NPEDF$ policy has been proposed in [12, 23, 24] as a link scheduling policy in a scheme designed to provide per session real-time guarantees in packet-switched networks.

Tracking policies have been proposed and studied in the context of Generalized Processor Sharing in [20, 10]. Theorems 1 and 2 appear in [20] but have been extended in this paper to include all tracking policies that obey a specific Ordering Property. The $T(PEDF)$ and $T(PEDF^*)$ policies are new as are all of the results pertaining to these policies. Finally, while the optimality of $PEDF$ for the criterion of minimizing the maximum lateness of packets was established in [11], the

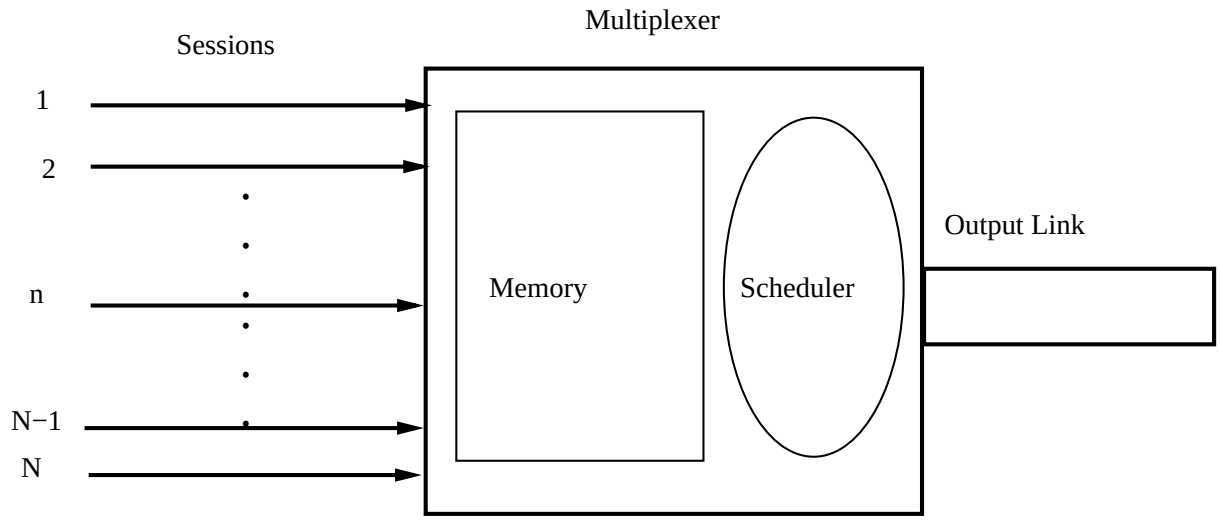

Figure 1: The multiplexer model

relationship between $PEDF$ and $NPEDF$ in this context is new. The lexicographic optimality of $PEDF^*$ is new as well.

2 Model, Definitions and Preliminary Results

2.1 Multiplexer and Traffic Models

Assume traffic flows from N sessions arrive into a multiplexer (see Figure 1) and the flow of each session is partitioned into discrete entities or packets. A packet may be arbitrarily small, but can be no larger than $L_{\max}$ bits. Arriving packets are stored in the memory of the multiplexer until they are transmitted on the output link, which is assumed to be of speed r. The multiplexer is of store-and-forward type, i.e., a packet becomes eligible for transmission only after its last bit has arrived at the multiplexer. Since there may be several eligible packets at any given time, the multiplexer has a *scheduler* which implements a *service policy*. This policy decides which of the eligible packets to transmit on the output link and then transmits this packet non-preemptively. In this paper, we assume a First-In-First-Out order of service for packets from a given session, so that the service policy only arbitrates transmission between the head-of-the-line packets from each session.

For definiteness, in the following we assume that if a packet arrives, i.e., its last bit is received, at the multiplexer at time t, it is also available for transmission at the scheduler at time t. Therefore, the scheduler takes into account the packet arrival at time t when making a scheduling decision at t. Also, when a packet is being transmitted, we say that the packet is "being served". By convention, at the time instant at which the transmission ends, the packet is not in service. So, if a packet is transmitted from time t_1 to $t_2 > t_1$, the packet is being served in the interval $[t_1, t_2)$.

Let $I_i(\tau, t)$ be the number of bits (traffic) generated by the source of session i in the interval $[\tau, t+\tau)$. Set $I_i(\tau, t+\tau) = 0$ for $t < 0$. Unless specified otherwise, assume that there exist σ_i, ρ_i such that

$$I_i(\tau, t+\tau) \leq \sigma_i + \rho_i t, \quad t \geq 0. \tag{1}$$

This model for the generated traffic is identical to the one proposed by Cruz [8] [9], and consistent with the constraints imposed by rate control algorithms that have been accepted by standard bodies [2, 1]. We refer to ρ_i and σ_i as the session traffic rate and burstiness respectively.

Let r_i be the speed of the input link over which traffic from session i is sent to the multiplexer. Since we are dealing with a store-and-forward multiplexer, a packet has to be completely received before it is delivered to the scheduler. It can then be shown using the techniques in [8] that the amount of (packetized) traffic from session i delivered to the scheduler in the interval $[\tau, \tau + t)$, $A(\tau, \tau + t)$, satisfies,

$$A_i(\tau, t + \tau) \leq L_{\max} + \min\{r_i t, \sigma_i + \rho_i t\}, \ t \geq 0. \tag{2}$$

Therefore, assuming infinite input link speeds (and using for consistency the convention $\infty \times 0 = 0$ when $t = 0$), we have,

$$A_i(\tau, t + \tau) \leq L_{\max} + \sigma_i + \rho_i t, \ t \geq 0. \tag{3}$$

To keep the discussion simple, we will mainly deal with constraints of the form (3) in this paper, and wherever possible we will mention interesting results that can be derived for more general constraints using similar arguments. More general constraints of the form of piecewise linear concave functions are presented in [8] while constraints of the form $A_i(\tau, t + \tau) \leq \overline{A}_i(t)$, $t \geq 0$, where $\overline{A}_i(t)$, $t \geq 0$, is a nondecreasing sub-additive function, are also possible [6]. We call $\overline{A}_i(t)$ the "envelope" of $A_i(\tau, t + \tau)$. For simplicity, whenever there is no possibility for confusion, we will write A_i to denote $A_i(\tau, t + \tau)$.

Note: The following general remark regarding the validity of the results under finite input link speeds can be made. Under constraint (3), the session traffic pattern $A_i(0, t) = L_{\max} + \sigma_i + \rho_i(t - \tau)$, $i = 1, \cdots, N$, is feasible. This traffic pattern, which we refer to as the "greedy" pattern, will be used in the various arguments in the sequel. Since, however, the greedy pattern is not consistent with (2), results depending on it will not hold in general for finite input link speeds. On the other hand, since (2) is stronger than (3), results that depend only on the inequality $A_i(\tau, t+\tau) \leq L_{\max} + \sigma_i + \rho_i t$, will also hold for finite input link speeds. For example, upper bounds on buffer size will generally hold, while lower bounds may not.

In the sequel and unless otherwise specified, we make the stability assumption

$$\sum_{i=1}^{N} \rho_i \leq r. \tag{4}$$

We denote by $\mathcal{C}(\vec{\rho}, \vec{\sigma})$ the set of vectors of session traffic arrivals, $\vec{A} = \{A_1, \cdots, A_N\}$ that are constrained by (3) and (4), with rate and burstiness vectors $\vec{\rho}$ and $\vec{\sigma}$ respectively.

Next, we introduce some notation needed in the rest of the paper. Let the scheduler implement policy π and let $\vec{A}$ be the session traffic arrival vector. We denote by $S_i^{\pi}(\tau, t, \vec{A})$ the number of session i bits served in the interval $[\tau, t)$ and by $Q_i^{\pi}\left(t, \vec{A}\right)$ the number of session i bits stored at time t. Define $M_i^{\pi}(\vec{\rho}, \vec{\sigma})$ as the largest amount of bits from session i that can be stored in the memory under policy π and under any traffic vector $\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})$, i.e.,

$$M_i^{\pi}(\vec{\rho}, \vec{\sigma}) = \sup_{t \geq 0} \sup_{\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})} Q_i^{\pi}\left(t, \vec{A}\right). \tag{5}$$

The delay of a packet is defined as the time it spends in the system, i.e., the sum of the time spent waiting in the memory since its last bit arrives and the time taken to transmit it on the output

link. The maximum delay experienced by packets in session i under any traffic vector $\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})$, is denoted by $D_i^{\pi}(\vec{\rho}, \vec{\sigma})$.

For notational convenience, when there is no possibility for confusion we may not indicate explicitly the dependence of the quantities defined above on $\vec{\rho}$, $\vec{\sigma}$, $\vec{A}$ or π.

2.2 Tracking Service Disciplines

In this section we introduce the notion of Tracking Service Disciplines which is used in several instances in the following sections. This notion was introduced in [20] for the purpose of tracking the Generalized Processor Sharing (GPS) discipline. It turns out that the fundamental properties of these policies (see Theorems 1 and 2 below) hold for tracking policies other than GPS, and this enables us to prove the delay and buffer optimality of various tracking service disciplines.

Given a preemptive policy π, the notion of tracking is to derive a work-conserving, non-preemptive policy $T(\pi)$ that operates as follows: Let $f_p^{\pi}(t)$ be the time at which packet p departs from a multiplexer that implements policy π assuming that there are no arrivals after time t. Then at each decision epoch t of $T(\pi)$, the server schedules a packet with the minimum value of $f_p^{\pi}(t)$ over all eligible packets present in the system at time t. Thus, $T(\pi)$ attempts to preserve the order in which packets depart under the preemptive system. At each decision epoch t, the $T(\pi)$ server picks the next packet that would depart from the system under the preemptive system if no more packets were to arrive after time t. Since more than one packet may leave the preemptive system simultaneously, ties are broken arbitrarily.

When π obeys the following *Ordering Property,* we can establish a tight coupling between the sample paths of π and $T(\pi)$:

> Let packets p and p' both be in the system at time τ and suppose that packet p completes service before packet p' if there are no arrivals after time τ. Then packet p will also complete service before packet p' for any pattern of arrivals after time τ. Further, if p and p' leave the system simultaneously when there are no arrivals after time τ, then they leave the system simultaneously for any pattern of arrivals after time τ.

The ordering property essentially requires that future arrivals do not modify the relative priorities of packets waiting to be transmitted. A consequence of the ordering property is that if the tracking server schedules a packet p at time τ before another packet p' that is also backlogged at time τ, then packet p cannot leave later than packet p' in the preemptive system.

This leads to the following results (first developed in the context of Generalized Processor Sharing in [20]). Let f_p be the time at which packet p departs from the preemptive system and let $\hat{f}_p$ be the time it departs from the tracking system. Then:

Theorem 1 *Suppose the ordering property holds for the preemptive system. For all packets p,*

$$\hat{f}_p - f_p < \frac{L_{\max}}{r}. \tag{6}$$

Proof. The proof follows along the lines of the proof of Theorem 1 in [21]. We present it here for the convenience of the reader.

Since both the preemptive and tracking systems are work conserving disciplines, their busy periods coincide, i.e., the preemptive system server is in a busy period iff the tracking server is in a busy period. Hence it suffices to prove the result for each busy period. Consider any given busy period and denote the time that it begins as time zero. Let L_k be the length of the kth packet (packet k) to depart under the tracking server and let a_k be its arrival time. We now show that for $k = 1, 2, ...$:

$$\widehat{f}_k < f_k + \frac{L_{\max}}{r}$$

Let m be the largest integer that satisfies both $0 < m \leq k-1$ and $f_m > f_k$. Thus

$$f_m > f_k \geq f_i \quad \text{for } m < i < k. \tag{7}$$

Then packet m is transmitted before packets $m+1 \ldots, k$ in the tracking system, but after all these packets in the preemptive system. If no such integer m exists then set $m = 0$. Now for the case $m > 0$, packet m begins transmission at $\widehat{f}_m - \frac{L_m}{r}$, so from the Ordering Principle:

$$\min\{a_{m+1}, ..., a_k\} > \widehat{f}_m - \frac{L_m}{r} \tag{8}$$

Since packets $m+1, ..., k-1$ arrive after $\widehat{f}_m - \frac{L_m}{r}$, they receive all their service in the preemptive system after time $\widehat{f}_m - \frac{L_m}{r}$. Also, from (7), they receive all their service before packet k departs at time f_k. Thus

$$\begin{aligned} f_k &\geq \frac{1}{r}(L_{m+1} + \ldots + L_{k-2} + L_{k-1} + L_k) + \min\{a_{m+1}, ..., a_k\} \\ &> \widehat{f}_m - \frac{L_m}{r} + \frac{1}{r}(L_{m+1} + \ldots + L_{k-2} + L_{k-1} + L_k). \end{aligned}$$

Since the right hand side of the above inequality is equal to $\widehat{f}_k - L_m/r$, we finally obtain,

$$\widehat{f}_k < f_k + \frac{L_m}{r} \leq f_k + \frac{L_{\max}}{r}. \tag{9}$$

If $m = 0$, then $p_1, ..., p_{k-1}$ all leave the preemptive system before packet k does, and since the tracking server is work-conserving,

$$f_k \geq \widehat{f}_k > \widehat{f}_k - \frac{L_m}{r}.$$

□

Theorem 2 *Suppose the ordering property holds for the preemptive policy π: Then for all times $t \geq 0$ and for each session i:*

$$Q_i^{T(\pi)}(t) - Q_i^{\pi}(t) \leq L_{\max}. \tag{10}$$

Proof. Follows from Theorem 1 and identical arguments as in Theorem 2 of [21] □

3 Buffer Allocation Mechanisms and Buffer Requirements

In this section, we address the problem of designing scheduling policies with minimal buffer requirements. We will assume that the session burstiness vector $\vec{\sigma}$ (or the supremum over all the

session burstiness vectors) is known and fixed, and that the rate vector $\vec{\rho}$, while known, can vary as long as it satisfies the stability condition (4). An important factor that affects the design of such policies and the corresponding buffer sizes, is the flexibility of the buffer allocation mechanism (the function of assigning memory locations to arriving packets) used in the multiplexer. We consider three natural ways in which the multiplexer can structure its buffers:

1. Flexible Allocation (FL): Packets from all arrival streams share a common pool of memory, i.e. buffers are not allocated by session. This provides the most efficient use of memory, but may be difficult to implement since the multiple input links require that multiple parallel writes be implemented by a single control logic. In addition, a dynamic linked list structure is also needed to maintain packet order. In this case, the minimum multiplexer buffer size needed when policy π is implemented, B^{π}_{FL}, is,

$$B^{\pi}_{FL} = \sup_{\vec{\rho}} \sup_{t \geq 0} \sup_{\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})} \sum_{i=1}^{N} Q^{\pi}_i \left(t, \vec{A} \right). \tag{11}$$

2. Semi-Flexible Allocation (SE): There are b^{π}_i bits of buffer allocated to packets from session i. The value of b^{π}_i cannot be changed after $t = 0$, however, the multiplexer is allowed to allocate the buffers based on the knowledge of $\vec{\rho}$ and $\vec{\sigma}$. This limits the amount of memory sharing, but only requires the multiplexer to be programmable so that the allocations can match the session traffic characteristics. The link list structure then becomes simpler to implement than with a flexible allocation. Also, the multiple parallel writes can now be implemented through separate control logic modules. In this case,

$$B^{\pi}_{SE} = \sup_{\vec{\rho}} \sum_{i=1}^{N} \sup_{t \geq 0} \sup_{\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})} Q^{\pi}_i \left(t, \vec{A} \right) = \sup_{\vec{\rho}} \sum_{i=1}^{N} M^{\pi}_i (\vec{\rho}, \vec{\sigma}). \tag{12}$$

3. Fixed Allocation (FI): There are $\bar{b}^{\pi}_i$ bits of buffer allocated to packets from each session i that should be sufficient for all $\vec{\rho}$ consistent with (4), i.e.,

$$\bar{b}^{\pi}_i \geq \sup_{\vec{\rho}} \sup_{t \geq 0} \sup_{\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})} Q^{\pi}_i \left(t, \vec{A} \right).$$

   Therefore,

$$B^{\pi}_{FI} = \sum_{i=1}^{N} \sup_{\vec{\rho}} \sup_{t \geq 0} \sup_{\vec{A} \in \mathcal{C}(\vec{\rho}, \vec{\sigma})} Q^{\pi}_i \left(t, \vec{A} \right). \tag{13}$$

   Note that knowledge of $\vec{\rho}$ is not useful in the design of a Fixed Allocation policy since, according to the definition, the allocated buffer space $\bar{b}^{\pi}_i$ is fixed and sufficient to accommodate all possible $\vec{\rho}$ consistent with (4).

Clearly, we have that

$$B^{\pi}_{FL} \leq B^{\pi}_{SE} \leq B^{\pi}_{FI},$$

while the complexity and cost of implementation reduces from FL to SE to FI.

Given $\alpha \in \{\text{Flexible, Semi-Flexible, Fixed}\}$, policy π^* is buffer-optimal policy among the class of admissible policies $\mathcal{C}$, if

$$B^{\pi^*}_{\alpha} \leq B^{\pi'}_{\alpha}, \text{ for all } \pi' \in \mathcal{C}.$$

We also define,

$$B_\alpha := \inf_{\pi \in \mathcal{C}} B_\alpha^\pi. \tag{14}$$

Unless otherwise specified, in the following, the class of admissible policies, $\mathcal{C}$, will be the class of work-conserving non-preemptive policies.

3.1 Buffer-Optimal Multiplexers

In this section we address the issue of determining B_α (as defined in (14)) and the scheduling policies that achieve B_α, for flexible, semi-flexible and fixed buffer allocation multiplexers.

Proposition 1 *For flexible allocation, $B_{FL} = N L_{\max} + \sum_{i=1}^N \sigma_i$, and this value is achieved by any work-conserving service policy.*

Proof. Suppose π is some work-conserving policy. Consider an arbitrary busy period that starts at τ_0 and ends at τ_1. Notice that the maximum number of bits from session i that can enter the system in the interval $[\tau_0, t)$, $\tau_0 \le t \le \tau_1$, is $L_{\max} + \sigma_i + \rho_i(t - \tau_0)$, i.e., the maximum number of bits that can be in the system corresponds to the greedy traffic pattern, starting from time τ_0. Since π continuously serves packets in $[\tau_0, t)$, we have,

$$\max_{\tau_0 \le t < \tau_1} \sum_{i=1}^N Q_i(t) \le N L_{\max} + \sum_{i=1}^N (\sigma_i + \rho_i(t - \tau_0)) - r(t - \tau_0)$$

Constraint (4) implies that the right hand side in the previous inequality reaches its maximum at time $t = \tau_0$. Thus

$$B_{FL} \le N L_{\max} + \sum_{i=1}^N \sigma_i.$$

This bound is achieved under the greedy traffic pattern. □

Note: The previous argument can be extended in a straightforward fashion if session i has envelope $L_{\max} + \min\{r_i t,\ \sigma_i + \rho_i t\}$. Let $\overline{\tau} \ge 0$ be the earliest time at which the slope of the function $\sum_{i=1}^N \min\{r_i t,\ \sigma_i + \rho_i t\}$ becomes less than or equal to r. Then, observing that the maximum of

$$N L_{\max} + \sum_{i=1}^N \min\{r_i(t - \tau_0),\ \sigma_i + \rho_i(t - \tau_0)\} - r(t - \tau_0)$$

occurs at time $t = \tau_0 + \overline{\tau}$ and following identical arguments we conclude that

$$B_{FL} = N L_{\max} + \sum_{i=1}^N \min\{r_i \overline{\tau},\ \sigma_i + \rho_i \overline{\tau}\} - r\overline{\tau}.$$

Next, we investigate the buffer requirements of the semi-flexible allocation.

Proposition 2 *For semi-flexible allocation, $B_{SE} \ge L_{\max}(2N - 1) + \sum_{i=1}^N \sigma_i$.*

Proof. Fix an integer $K \geq 1$, and consider the following arrival rates.

$$\rho_i^K = \frac{K}{(1+K)^i} r, \quad i = 1, \ldots, N-1,$$

and

$$\rho_N^K \leq \frac{1}{(1+K)^{N-1}} r.$$

We assume that all packets are of size $L = L_{\max}$. Let $T = L/r$ be the time taken to transmit a packet. Let the system operate under a scheduling policy π and denote by $\vec{G}$ the following traffic pattern. A packet of length L from session N arrives at time 0, and no more traffic from session N arrives afterwards. The greedy traffic patterns from sessions 1 to $N-1$ arrives at time 0^+. By time 0^+ we mean "immediately after", i.e., at time $\varepsilon > 0$, where ε is arbitrarily small. Thus, since π is work-conserving and non-preemptive, the packet from session N will be transmitted in the interval $[0, T)$. We will use this notation in the sequel, but will avoid the incorporation of ε, since it would complicate the discussion unnecessarily. Note also that although the packets are of constant length, the greedy pattern of each session can still appear at the input link to the multiplexer. However, the number of packets from session i that will be delivered to the scheduler at time 0^+ is $\lfloor (L+\sigma_i)/L \rfloor$. The rest of the bits, $L + \sigma_i - \lfloor (L+\sigma_i)/L \rfloor L$ must wait in memory until a complete packet is formed.

Define

$$\tilde{M}_i^\pi(\vec{\rho}^K, \vec{\sigma}) = \sup_{t \geq 0} Q_i^\pi(t, \vec{G}).$$

Note the difference between $\tilde{M}_i^\pi(\vec{\rho}^K, \vec{\sigma})$ and $M_i^\pi(\vec{\rho}^K, \vec{\sigma})$: in the latter we take in addition the supremum over *all* arrival patterns consistent with (3) and (4). We will show that

$$\sum_{i=1}^{N-1} \tilde{M}_i^\pi(\vec{\rho}^K, \vec{\sigma}) \geq (N-1)L + \sum_{i=1}^{N-1} \sigma_i + (N-1)\frac{K}{K+1} L. \tag{15}$$

Since we clearly have that $M_i^\pi(\vec{\rho}^K, \vec{\sigma}) \geq \tilde{M}_i^\pi(\vec{\rho}^K, \vec{\sigma})$ and $M_N^\pi \geq L + \sigma_N$, (15) implies that for any K,

$$\sum_{i=1}^{N} M_i^\pi(\vec{\rho}^K, \vec{\sigma}) \geq NL + \sum_{i=1}^{N} \sigma_i + (N-1)\frac{K}{K+1} L$$

and letting $K \to \infty$ we conclude that

$$B_{SE} \geq \lim_{K \to \infty} \left(\inf_{\pi \in C} \sum_{i=1}^{N} M_i^\pi(\vec{\rho}^K, \vec{\sigma}) \right) \geq (2N-1)L + \sum_{i=1}^{N} \sigma_i,$$

as desired.

For simplicity in the notation, we will drop the dependence on $\vec{\rho}^K$ and $\vec{\sigma}$ in the rest of this proof. To show (15), let us consider the following slightly more general system Π, that consists of sessions 1 to $N-1$. The buffer content of session i, $1 \leq i \leq N-1$, at time 0 is $Q_i(0)$ and session i, $1 \leq i \leq N-1$, sends traffic greedily at rate ρ_i^K after time 0, but it cannot use the server in the interval $[0, T)$. Considering the traffic of sessions 1 to $N-1$ only, the original system differs from Π only in the initial conditions (in the original system we have the special case $Q_i(0) = L + \sigma_i$). Note that under both systems, the traffic from sessions 1 to $N-1$ cannot use the server in the interval $[0, T)$ (by definition in system Π, while in the original system a packet from session N is served in $[0, T)$)

Setting $n = N - 1$, we will show that for system Π, under *any* policy (including idling) π_n,

$$\sum_{i=1}^{n} \tilde{M}_i^{\pi_n} \geq \sum_{i=1}^{n} Q_i(0) + n\frac{K}{K+1}L, \tag{16}$$

which is equivalent to (15).

For the proof of (16) we will use induction on n. For $n = 1$, (16) is clearly true since session 1 will have to wait at least until time T before it is served (notice that session 1 will have to wait even longer if $Q_1(0) + K/(K+1)L < L$ since there will be no complete packet in the multiplexer.) Assume now that (16) is true for n. Consider a system Π consisting of $n+1$ sessions and let τ be the first time that session $n+1$ is served under an arbitrary policy π_{n+1}. Note that since by the definition of system Π no session can use the server in $[0, T)$, we have that $T \leq \tau$. The following two possibilities arise.

1. $\tau \geq (K+1)^n T$. Consider policy π_n that serves only packets from session 1 to n in exactly the same manner as policy π_{n+1}. Whenever π_{n+1} serves a packet from session $n+1$, π_n idles. Note that π_n satisfies the requirements of the inductive hypothesis for n. Therefore, using the fact that $\tilde{M}_i^{\pi_{n+1}} = \tilde{M}_i^{\pi_n}$, $1 \leq i \leq n$, we have

$$\sum_{i=1}^{n} \tilde{M}_i^{\pi_{n+1}} = \sum_{i=1}^{n} \tilde{M}_i^{\pi_n} \geq \sum_{i=1}^{n} Q_i(0) + n\frac{K}{K+1}L.$$

   Since session $n+1$ was not served in the interval $[0, \tau)$, we have also

$$\tilde{M}_{n+1}^{\pi_{n+1}} \geq Q_{n+1}(0) + r\tau\frac{K}{(K+1)^{n+1}} \geq Q_{n+1}(0) + \frac{K}{K+1}L$$

   and therefore (16) holds for $n+1$.

2. $T \leq \tau < (K+1)^n T$. The traffic served from sessions 1 to n in the interval $[0, \tau)$ is at most $r\tau - L$ (it may be less if there are no packets from sessions 1 to n to be served at some time in $[T, \tau)$ or the server idles). Therefore, the sum of the buffer contents of sessions 1 to n at time τ is

$$\begin{aligned} \textstyle\sum_{i=1}^{n} Q_i^{\pi_{n+1}}(\tau) &\geq \textstyle\sum_{i=1}^{n} Q_i(0) + \tau \sum_{i=1}^{n} \frac{K}{(K+1)^i} r - (r\tau - L) \\ &= \textstyle\sum_{i=1}^{n} Q_i(0) + r\tau\left(1 - \frac{1}{(K+1)^n}\right) - (r\tau - L) \\ &= \textstyle\sum_{i=1}^{n} Q_i(0) + L - \frac{r\tau}{(K+1)^n} \end{aligned} \tag{17}$$

   Since a packet from session $n+1$ is served in the interval $[\tau, \tau+T)$, we can apply the inductive hypothesis to the policy π_n that schedules only packets from sessions 1 to n after time τ in exactly the same manner as π_{n+1}, and with initial buffer contents $Q_i^{\pi_{n+1}}(\tau)$, $1 \leq i \leq n$. Using also (17) we get

$$\begin{aligned} \textstyle\sum_{i=1}^{n} \tilde{M}_i^{\pi_{n+1}} &\geq \textstyle\sum_{i=1}^{n} Q_i^{\pi_{n+1}}(\tau) + n\frac{K}{K+1}L \\ &\geq \textstyle\sum_{i=1}^{n} Q_i(0) + (n+1)\frac{K}{K+1}L + A, \end{aligned}$$

   where

$$A = L - \frac{r\tau}{(K+1)^n} - \frac{K}{K+1}L.$$

   Since the buffer requirements of session $n+1$ are at least

$$Q_{n+1}(0) + \frac{r\tau K}{(K+1)^{n+1}},$$

we finally have that

$$\sum_{i=1}^{n+1} \tilde{M}_i^{\pi_{n+1}} \geq \sum_{i=1}^{n+1} Q_i(0) + (n+1)\frac{K}{K+1}L + B,$$

where

$$\begin{aligned} B &= L - \tfrac{r\tau}{(K+1)^n} - \tfrac{K}{K+1}L + \tfrac{r\tau K}{(K+1)^{n+1}} \\ &= L - \tfrac{r\tau}{(K+1)^{n+1}} - \tfrac{K}{(K+1)}L. \end{aligned}$$

Since by assumption $\tau < (K+1)^n T$, we have $r\tau/(K+1)^{n+1} < L/(K+1)$ and, therefore, $B > 0$. Hence, the induction hypothesis holds for $n+1$. □

Before dealing with the fixed allocation case, we present a preemptive service policy called *Rate Proportional Processor Sharing (RPPS)* that was introduced in [21]. Recall that under our model, bits of a packet p are only eligible for service once the last bit of packet p has arrived. Let a session be backlogged at time t, if a positive amount of eligible session i traffic is queued at time t. Then the $RPPS$ server ensures that for any session i, if session i is continuously backlogged in the interval $[\tau, t]$, then

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} \geq \frac{\rho_i}{\rho_j}, \quad j = 1, 2, ..., N. \tag{18}$$

Notice that if i and j are both continuously backlogged in the interval, then (18) is met with equality. Also note that the $RPPS$ policy obeys the Ordering Property discussed in Section 2.2. The following result is adapted from [21].

Proposition 3 *For fixed allocation, $B_{FI} = 2NL_{\max} + \sum_{i=1}^{N}\sigma_i$, and this value is achieved by T(RPPS).*

Proof. We show first that the buffer requirements of session i under any policy π are at least $2L_{\max} + \sigma_i$. Consider the following arrival pattern. The system is empty at time 0. A packet of length $L_{\max}$ from session $j \neq i$ arrives at time 0. At time 0^+ traffic from session i arrives greedily. Since π is work-conserving and non-preemptive, the packet from session j begins service at time 0 and the traffic from session i cannot begin service before time $L_{\max}/r$. Therefore, the queue size of session i at the time t when traffic from session i is first served is at least

$$Q_i^{\pi}(t) \geq \frac{L_{\max}}{r}\rho_i + L_{\max} + \sigma_i.$$

Letting $\rho_i \to r$, we conclude that the buffer requirements of session i are at least $2L_{\max} + \sigma_i$, and this implies that

$$B_{FI} \geq 2NL_{\max} + \sum_{i=1}^{N}\sigma_i.$$

To see that $T(RPPS)$ meets this bound note that since under $RPPS$ the rate of service received by session i is at least ρ_i, [21],

$$Q_i^{RPPS}(t) \leq \sigma_i + L_{\max}.$$

Applying Theorem 2 and summing over i, we get the desired result. □

Since $B_{FI} \geq B_{SE}$, from Propositions 2 and 3 we immediately get the following result.

Corollary 1 $2NL_{\max} + \sum_{i=1}^{N} \sigma_i = B_{FI} \geq B_{SE} \geq L_{\max}(2N-1) + \sum_{i=1}^{N} \sigma_i$

Notes.

1. Although Corollary 1 indicates that the semi-flexible allocation does not provide significant savings in terms of buffer requirements compared to the fixed allocation, we will see in the next sections that when packet delays are also considered, the semi-flexible allocation provides the flexibility of designing delay-optimal policies with low buffer requirements. This remark not withstanding, it should be pointed out that the *T(RPPS)* policy, which from Proposition 3 has low buffer requirements under fixed allocation, is also capable of providing low, albeit not optimal, delay bounds.

2. In [4], it was shown that when $\sigma_i = 0$, $i = 1, \ldots, N$, and when the First-Come-First-Served (FCFS) policy is employed,

$$Q_i(t) \leq L_{\max}\left(1 - \frac{\rho_i}{r}\right) + \frac{\rho_i}{r} N L_{\max}.$$

   By summing over all i, we conclude that $B_{SE}^{FCFS} \leq (2N-1)L_{\max}$. Together with Proposition 2, this implies that when $\sigma_i = 0$, $i = 1, \ldots, N$, the FCFS is buffer-optimal for semi-flexible allocation. However, this is not true for general σ_i, as the following example shows.

   Consider the following arrival pattern. A packet of length $L_{\max}$ together with a burst of size σ_j arrives from each of the sessions $j \neq i$ at time 0. At time 0^+ a packet p from session i of length $L_{\max}$ arrives, followed immediately by a burst of packets of total size σ_i. After time 0, traffic from session i arrives at rate ρ_i. Assume also that $\sum_{i=1}^{N} \rho_i = r$. It is easy to see that at the time t when packet p enters service,

$$Q_i(t) = \left(\frac{(N-1)L_{\max} + \sum_{j \neq i}^{N} \sigma_j}{r}\right) \rho_i + L_{\max} + \sigma_i.$$

   Summing over i we see that

$$B_{SE}^{FCFS} \geq (2N-1)L_{\max} + 2\sum_{i=1}^{N} \sigma_i - \sum_{i=1}^{N} \frac{\rho_i \sigma_i}{r} \geq (2N-1)L_{\max} + 2\sum_{i=1}^{N} \sigma_i - \max_{1 \leq i \leq N} \sigma_i,$$

   which by Corollary 1 can be larger than B_{SE} in general.

4 Buffer requirements v/s Delay. Delay Optimal Policies

In this section, we address the issue of designing scheduling policies that provide predetermined delay bounds to each of the sessions and have low buffer requirements. We start with a result that we need later and which is of independent interest. It expresses the relationship that exists between bounds on the delays and buffer requirements.

Recall the definition of $D_i^{\pi}(\vec{\rho}, \vec{\sigma})$ and $M_i^{\pi}(\vec{\rho}, \vec{\sigma})$ from Section 2. In Theorem 3, we establish a useful lower bound on $D_i^{\pi}(\vec{\rho}, \vec{\sigma})$ as a function of $M_i^{\pi}(\vec{\rho}, \vec{\sigma})$ and the characteristics (σ_i, ρ_i) of session i.

Theorem 3 *For any zero-loss multiplexer implementing policy π that serves packets from session i in a FCFS order, it holds,*

$$D_i^{\pi}(\vec{\rho}, \vec{\sigma}) \geq \frac{M_i^{\pi}(\vec{\rho}, \vec{\sigma}) - \sigma_i - L_{\max}}{\rho_i} \tag{19}$$

Proof. We drop the superscript of π and the dependence on $\vec{\rho}, \vec{\sigma}$ for notational convenience. Consider the traffic pattern under which the supremum in the definition of M_i (see eq. (5)) is achieved and let t^* be such that $Q_i(t^*) = M_i$ under this traffic pattern (since M_i is a supremum, it may not be achieved at any time, however the same argument as the one that follows can be used by using appropriate "epsilons"). We focus on the first complete packet present in the queue of session i at time t^* (since the multiplexer is of store-and-forward type, $M_i \geq L_{\max}$ and therefore there is always a complete packet in the queue of session i at time t^*). Let $\widehat{t} \leq t^*$ be the arrival time of that packet (recall that the arrival time of the packet is the time the last bit of the packet arrives to the scheduler). Then, since packets from session i are served in a FCFS order,

$$\rho_i(t^* - \widehat{t}) + \sigma_i + L_{\max} \geq M_i$$

or,

$$t^* - \widehat{t} \geq \frac{M_i - L_{\max} - \sigma_i}{\rho_i}, \tag{20}$$

where we have used the fact that due to the FCFS property, the amount of traffic stored in the buffer at time t^* is at most the amount of work that session i can generate in the time interval $[\widehat{t}, t^*]$ which in turn is bounded by and $\rho_i(t^* - \widehat{t}) + \sigma_i + L_{\max}$. Letting $\widehat{d}$ be the delay of the packet at the head of the queue at time t^*, we have from (20)

$$\widehat{d} \geq \frac{M_i - L_{\max} - \sigma_i}{\rho_i}.$$

□

Notes:

1. One of the reviewers suggested the following bound on the delay. Consider the last complete packet in the queue of session i at time t^*. The amount of traffic that needs to be transmitted in order for this packet to be sent on the output link is at least $M_i - L_{\max}$. Therefore, even if the scheduler is allocated solely to session i, the delay of the this last packet will be at least $(M_i - L_{\max})/r$. Therefore, we have another bound

   $$D_i^{\pi}(\vec{\rho}, \vec{\sigma}) \geq \frac{M_i^{\pi}(\vec{\rho}, \vec{\sigma}) - L_{\max}}{r}. \tag{21}$$

   In general, the bounds (19) and (21) do not imply each other and therefore a tighter bound can be obtained by taking the maximum of the two. For our purposes, bound (19) is sufficient.

2. Clearly, bound (21) holds for general traffic envelopes. Bound (19) can also be extended to general envelopes. Indeed consider that the session envelope is $A_i(t)$, where $A_i(t)$ is (strictly) increasing and sub-additive. Then, repeating the arguments in the proof of Theorem 3 we conclude that

   $$D_i^{\pi}(\vec{\rho}, \vec{\sigma}) \geq A_i^{(-1)}\left(M_i^{\pi}(\vec{\rho}, \vec{\sigma})\right),$$

   where $A_i^{(-1)}(x)$ is the inverse of $A_i(t)$. If $A_i(t)$ is nondecreasing, a similar formula can be given by going through the obvious modifications to account for intervals where $A_i(t)$ is not strictly increasing.

4.1 Delay-Optimal Policies

In this section, we address first the issue of designing delay-optimal policies. In the next section, we address the issue of designing policies that are delay-optimal and also have low buffer requirements.

To proceed, we need some notation and definitions. Let the non-negative vector $\vec{D} = (D_1, ..., D_N)$ be a list of required upper bounds on delay so that no session i packet is delayed by more than D_i time units in the multiplexer. The *deadline* of packet p from session i that arrives at time a_p is defined as $d_p = a_p + D_i$. If f_p is the finishing time of p, its *lateness* is defined as $l_p = f_p - d_p$. Given a zero-loss multiplexer that implements service policy π, the vector $\vec{D} = (D_1, ..., D_N)$ is *schedulable* under π if for all arrival patterns consistent with (3) and (4), and for all sessions i, no session i packet is delayed by more than D_i time units. The *schedulable region* Ω^π of the policy π is the set of all vectors schedulable under π. Given a class of admissible policies $\mathcal{C}$, the *schedulable region of* $\mathcal{C}$ is $\bigcup_{\pi \in \mathcal{C}} \Omega^\pi$ and a vector is *schedulable in* $\mathcal{C}$ if it belongs to the schedulable region of $\mathcal{C}$. We define a scheduling policy π^* to be delay-optimal in $\mathcal{C}$ if

$$\Omega^\pi \subseteq \Omega^{\pi^*} \tag{22}$$

for all policies $\pi \in \mathcal{C}$.

It has been shown in [11], that under any arrival pattern the Preemptive Earliest Deadline First ($PEDF$), i.e., the policy that at any instant schedules the packet with the smallest deadline first (ties are resolved by picking one of the packets with equal minimal deadlines in an arbitrary fashion), minimizes the maximum lateness of all the packets. This implies that the $PEDF$ policy is delay-optimal among all scheduling policies. To see this, assume there that the vector $\vec{D}$ is schedulable under a policy π. Then the lateness of every packet under π is nonpositive, and therefore the maximum lateness of all packets is nonpositive under π. But then, the same conclusion is true for $PEDF$ (since it minimizes the maximum lateness) and therefore the lateness of all packets under $PEDF$ is nonpositive. This means of course that the vector $\vec{D}$ is schedulable under $PEDF$, which implies that the schedulable region of $PEDF$ is a superset of that of π. For non-preemptive policies, no policy is known that minimizes the maximum lateness of all packets over all arrival patterns. However, we will show that under constraints (3) and (4) the non-preemptive EDF ($NPEDF$) (i..e., the policy that behaves like $PEDF$ but it takes decisions only at packet transmission completions or upon arrival of a new packet in an empty system) and the $PEDF$ tracking policy ($T(PEDF)$) are delay-optimal. We will also provide the schedulable region of these non-preemptive policies. We note that in general $NPEDF$ may differ significantly from $T(PEDF)$. This is demonstrated in Example 1 of Section 5.2, where we also show that in the important special case of fixed size packets, $T(PEDF)$ and $NPEDF$ behave identically.

We now proceed to show the optimality of both the $NPEDF$ and $T(PEDF)$ policies.

Theorem 4 *The $NPEDF$ and $T(PEDF)$ policies are delay-optimal among the class of non-preemptive policies. The schedulable regions of $NPEDF$ and $T(PEDF)$ consists of the set of vectors which satisfy the constraints*

$$\begin{aligned} (k+1)L_{\max} + \sum_{n=1}^{k} \sigma_{i_n} &\leq D_{i_k}\left(r - \sum_{n=1}^{k-1} \rho_{i_n}\right) + \sum_{n=1}^{k-1} \rho_{i_n} D_{i_n}, \quad 1 \leq k \leq N-1 \\ N L_{\max} + \sum_{n=1}^{N} \sigma_{i_n} &\leq D_{i_N}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_{i_n} D_{i_n}, \end{aligned}$$

whenever $D_{i_1} \leq D_{i_2} \leq \ldots \leq D_{i_N}$.

The Theorem is a conclusion of the following two lemmas. The first one establishes the necessary conditions for a vector to be schedulable under any non-preemptive policy, and the second demonstrates the sufficiency of these constraints for schedulability under $NPEDF$ and $T(PEDF)$. Let $U(t) = 1$ if $t \geq 0$ and 0 otherwise.

Lemma 1 *Let $D_1 \leq D_2 \leq \ldots \leq D_N$. If the vector $\{D_1, \ldots, D_N\}$ is schedulable under a non-preemptive policy then necessarily,*

$$\frac{L_{\max}}{r} \leq D_1, \tag{23}$$

$$\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(t - D_i))U(t - D_i) + L_{\max} \leq rt, \quad \frac{L_{\max}}{r} \leq t < D_N, \tag{24}$$

and

$$\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(t - D_i)) \leq rt, \quad t \geq D_N. \tag{25}$$

Proof. We follow the method of proof in [26]. Assume that all packets meet their deadlines under a non-preemptive policy. Clearly, we should have $(L_{\max}/r) \leq D_1$, since otherwise maximum length packets from any session are not schedulable. Consider the following arrival pattern. At time 0 the last bit of a packet of maximum length from session N, together with a burst of bit-size packets of total size σ_N arrives in the system. At time 0^+ the last bit of a packet of maximum length from session i, $1 \leq i \leq N-1$, together with a burst of bit-size packets of total size σ_i arrives. Afterwards, packets from session i, $1 \leq i \leq N$, arrive in bit-size at fixed rate ρ_i. Let $(L_{\max}/r) \leq t < D_N$. Since all packets meet their deadlines at time t, all packets from session i that arrived before or at time $t - D_i$ must be transmitted by t. The number of bits contained in these packets is $(L_{\max} + \sigma_i + \rho_i(t - D_i))U(t - D_i)$. Therefore, the number of bits from sessions 1 to $N-1$ that must be transmitted by time t is $\sum_{i=1}^{N-1}(L_{\max} + \sigma_i + \rho_i(t - D_i))U(t - D_i)$. Since the policy is non-preemptive and the packet from session N arrives first, the number of bits transmitted by time t from the rest of the sessions is at most $rt - L_{\max}$ and this implies (24). To show inequality (25), let $t \geq D_N$ and observe as before that the number of bits from all the sessions that can be transmitted by time t can be at most rt while the number of bits that must be transmitted is $\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(t - D_i))U(t - D_i)$. □

Lemma 2 *Let $D_1 \leq D_2 \leq \ldots \leq D_N$. Any vector $\{D_1, \ldots, D_N\}$ that satisfies the constraints of Lemma 1 is schedulable under both $NPEDF$ and $T(PEDF)$.*

Proof. Let $W(t, d)$ be the amount of work in the system with deadlines at most d at time t under either $NPEDF$ or $T(PEDF)$. We show that for all $t \geq 0$, $W(t, t) = 0$ which implies the lemma.

If the server is idle at time t, then since both policies are work-conserving, we have $W(t, t) = 0$. Assume therefore that the server is serving a packet at time t and define s as follows. If the server is serving a packet with deadline larger than t at time t, set $s = t$. Otherwise, let $s \leq t$ be the smallest time such that the server is continuously busy serving packets with deadlines at most t in the interval $[s,\ t)$. Let $\mathcal{P}$ be the set of packets with deadlines at most t that either are served in $[s,\ t)$, or are in the system at time t. If $\mathcal{P} = \emptyset$, then clearly $W(t, t) = 0$. Assume therefore that $\mathcal{P} \neq \emptyset$ and let e be the packet with the earliest arrival time, a_e, among the packets in $\mathcal{P}$. Observe

that the amount of work of the packets in $\mathcal{P}$ is $W(t,t) + r(t-s)$ and that all this work arrives at or after time a_e. Notice also that from (23) and the fact that packet e has deadline at most t, we have $t - a_e \geq D_1 \geq (L_{\max}/r)$.

If $a_e = s$ then using the upper bound on the amount of work with deadlines at most t that can arrive in the interval $[s,t)$, determined in the proof of Lemma 1, we get

$$\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(\bar{s} - D_i))U(\bar{s} - D_i) \geq W(t,t) + r\bar{s}, \quad \bar{s} = t - s = t - a_e \geq \frac{L_{\max}}{r}. \tag{26}$$

If $(L_{\max}/r) \leq \bar{s} < D_N$, (26) and (24) imply that $W(t,t) < 0$ and therefore this case cannot occur. If $\bar{s} \geq D_N$, (26) and (25) imply that $W(t,t) = 0$.

Assume now $a_e < s$. We will show that this case cannot occur. Let q be the packet that completes service or is in service at time s and let a_q be its arrival time. By the definition of s, packet q has a larger deadline than t which implies that $a_e > a_q$. This is so since under both $NPEDF$ and $T(PEDF)$ packet e cannot leave later than packet q if $a_e \leq a_q$.

Note that since the deadline of packet q is larger that t, we have that $t - a_q < d_q - a_q \leq D_N$. Taking also into account that $a_e > a_q$, we have $t - a_e < t - a_q < D_N$. Since all the work of the packets in $\mathcal{P}$ arrives at or after time a_e, setting $\hat{s} = t - a_e$ we have that

$$\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(\hat{s} - D_i))U(\hat{s} - D_i) \geq W(t,t) + r(t-s) = W(t,t) + r\hat{s} - r(s - a_e)$$

We will show next that

$$a_e > s - \frac{L_{\max}}{r}. \tag{27}$$

which will imply that $\sum_{i=1}^{N}(L_{\max} + \sigma_i + \rho_i(\hat{s} - D_i))U(\hat{s} - D_i) > W(t,t) + r\hat{s} - L_{\max}$. This inequality together with the fact that as discussed above, $(L_{\max}/r) \leq \hat{s} < D_N$ and (24) imply that $W(t,t) < 0$ which shows that the case $a_e < s$ cannot occur.

To show (27) for $NPEDF$ observe that by the definition of this policy, packet e must have arrived after packet q entered service and therefore, $s - a_e$ is less than the time to transmit a maximum length packet.

Consider now that the system operates under the $T(PEDF)$ policy. If $f_q > f_e$, then by the definition of $T(PEDF)$, packet e must have arrived after packet q entered service and therefore, (27) is true. If $f_q \leq f_e$ then note that from the definition of $PEDF$:

$$f_q \leq a_e, \tag{28}$$

since $d_q > d_e$. Now recall from Theorem 1 that:

$$f_q > s - \frac{L_{\max}}{r}.$$

Combining this with (28) yields (27). □

The schedulable region of $PEDF$ under the arrival patterns considered in this section can be found using similar arguments as those used to prove Theorem 4. For completeness, we present this result in the next theorem.

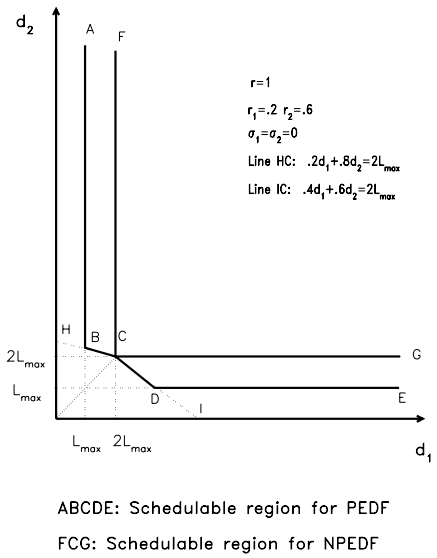

Figure 2: Schedulable regions for $\sigma_1 = \sigma_2 = 0$.

Theorem 5 *The schedulable region of $PEDF$ consists of the set of vectors which satisfy the constraints*

$$kL_{\max} + \sum_{n=1}^{k} \sigma_{i_n} \leq D_{i_k} \left(r - \sum_{n=1}^{k-1} \rho_{i_n} \right) + \sum_{i=1}^{k-1} \rho_{i_n} D_{i_n}, \quad 1 \leq k \leq N,$$

whenever $D_{i_1} \leq D_{i_2} \leq \ldots \leq D_{i_N}$.

In Figures 2 and 3, we show the schedulable regions of $PEDF$ and $NPEDF$ under various parameters. As we see, in both figures the two regions differ by two strips which have width $L_{\max}/r$. In fact, by examining the schedulable regions it is easy to see that if the vector $\{D_1, \ldots, D_N\}$ is schedulable under $PEDF$, then the vector $\{D_1 + L_{\max}/r, \ldots, D_N + L_{\max}/r\}$ is schedulable under $NPEDF$. As we will see in the next section, this is a consequence of a general result that holds for any arrival patterns. Also, we see in Figure 2, where $\sigma_1 = \sigma_2 = 0$, that any schedulable vector under $NPEDF$ has coordinates larger than $2L_{\max}/r$. Since, as is easy to see, the vector $\{2L_{\max}/r,\ 2L_{\max}/r\}$ is schedulable under the First-Come-First-Served (FCFS) policy, it follows that in this case from the point of view of schedulability there is no point in employing another scheduling policy. In fact, as can be seen from Theorem 4 this is true always when $N = 2$ and $\sigma_1 = \sigma_2 = 0$.

Note: Lemmas 1 and 2 extend in a straightforward fashion to general session envelopes $\overline{A}_i(t)$, $1 \leq i \leq N$. Indeed, defining $\overline{A}_i(t) = 0$ for $t < 0$, and replacing in the arguments the quantity $(L_{\max} + \sigma_i + \rho_i(t - D_i))U(t - D_i)$ with $\overline{A}_i(t)$, we obtain

Theorem 6 *Let $D_1 \leq D_2 \leq \cdots \leq D_N$. If session i, $1 \leq i \leq N$, has envelope $\overline{A}_i(t)$, then the $NPEDF$ and $T(PEDF)$ policies are delay-optimal among the class of non-preemptive policies and*

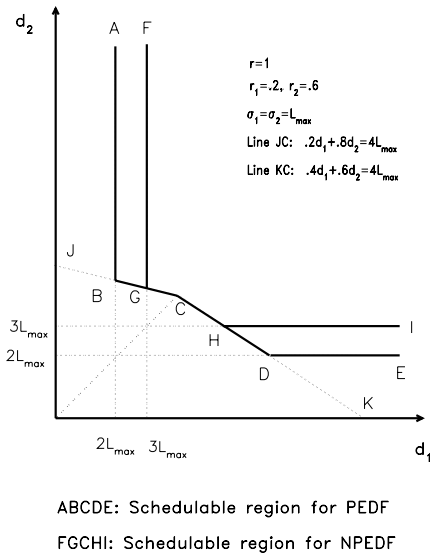

Figure 3: Schedulable regions for $\sigma_1,\ \sigma_2 \neq 0$.

their schedulable region consists of the set of vectors which satisfy the constraints

$$\frac{L_{\max}}{r} \leq D_1$$

$$\sum_{i=1}^{N} \overline{A}(t - D_i) + L_{\max} \leq rt, \quad \frac{L_{\max}}{r} \leq t < D_N,$$

and

$$\sum_{i=1}^{N} \overline{A}(t - D_i) \leq rt, \quad t \geq D_N.$$

The schedulable region of $PEDF$ under general session envelopes can be similarly derived.

4.2 Delay-Optimal Policies with Low Buffer Requirements

In this section, we address the issue of designing delay-optimal policies with low buffer requirements. We propose a policy that is delay-optimal and under semi-flexible allocation has low buffer requirements. Note that based on Proposition 1, a delay-optimal policy will also have minimum buffer requirements if a flexible allocation is used. However, we will see that the improvement over the semi-flexible case is small and may, therefore, not warrant the additional cost and complexity.

We first motivate the use of semi-flexible allocation by showing, that under fixed allocation the buffer requirements of any delay-optimal policy are at least $O(N^2)$.

Proposition 4 *Let π be any non-preemptive policy that is delay-optimal for all traffic patterns consistent with (3) and (4). Under fixed-allocation,*

$$B^{\pi}_{FI} \geq N^2 L_{\max} + N \sum_{i=1}^{N} \sigma_i.$$

Proof. Consider the vector of delays $\vec{D}$ given as the solution of the following system of equations.

$$\begin{aligned} (k+1)L_{\max} + \textstyle\sum_{n=1}^{k} \sigma_n &= D_k \left(r - \textstyle\sum_{n=1}^{k-1} \rho_n \right) + \textstyle\sum_{n=1}^{k-1} \rho_n D_n, \quad 1 \leq k \leq N-1 \\ N L_{\max} + \textstyle\sum_{n=1}^{N} \sigma_n &= D_N \left(r - \textstyle\sum_{n=1}^{N-1} \rho_n \right) + \textstyle\sum_{n=1}^{N-1} \rho_n D_n, \end{aligned}$$

The vector $\vec{D}$ is schedulable in the class of non-preemptive policies. This will follow from Theorem 4 once we show the inequality $D_k \leq D_{k+1}$, $k = 1, \ldots, N-1$, which is easily seen from the observation that by the definition of $\vec{D}$,

$$D_{k+1} \left(r - \sum_{n=1}^{k} \rho_n \right) + \sum_{n=1}^{k} \rho_n D_n \geq D_k \left(r - \sum_{n=1}^{k-1} \rho_n \right) + \sum_{n=1}^{k-1} \rho_n D_n.$$

Assume that a packet of length $L_{\max}$ from session N arrives at time 0 followed by a burst of bit-size packets of total length σ_N. At time 0^+ a packet of length $L_{\max}$ together with a burst of bit-size packets of total length σ_i arrives from session i, $i = 1, \ldots, N-1$. After time 0 session i sends traffic at rate ρ_i. Let us estimate the buffer content of session N at time D_{N-1}. Since all the packets meet their deadlines, the server must have served $(N-1)L_{\max} + \sum_{n=1}^{N-1} \sigma_n$ bits from sessions 1 to $N-1$. Therefore, the bits from class N served in $[0, D_{N-1})$ are at most

$$r D_{N-1} - (N-1) L_{\max} - \sum_{n=1}^{N-1} \sigma_n.$$

It follows that

$$Q^{\pi}_N(D_{N-1}) \geq L_{\max} + \sigma_N + D_{N-1}\rho_N - \left(r D_{N-1} - (N-1) L_{\max} - \sum_{n=1}^{N-1} \sigma_n \right). \tag{29}$$

Now let $\rho_i = r\epsilon/(N-1)$, $i = 1, \ldots, N-1$, and $\rho_N = r(1-\epsilon)$. It can be easily shown that

$$\lim_{\epsilon \to 0} \epsilon D_{N-1} = 0$$

and taking limits as $\epsilon \to 0$ in (29) we conclude that

$$M^{\pi}_N \geq \lim_{\epsilon \to 0} Q^{\pi}_N(D_{N-1}) \geq N L_{\max} + \sum_{n=1}^{N} \sigma_n$$

By interchanging the indices we conclude that

$$M^{\pi}_i \geq N L_{\max} + \sum_{n=1}^{N} \sigma_n, \quad i = 1, \ldots, N$$

and summing over i we get the desired result. □

It turns out that even under semi-flexible allocation, the delay-optimal policies $NPEDF$ and $T(PEDF)$ still have buffer requirements of at least $O(N^2)$.

Proposition 5 *With* $\alpha \in \{NPEDF,\ T(PEDF)\}$,

$$B_{SE}^{\alpha} \geq \frac{N(N+1)}{2} L_{\max} + \sum_{n=1}^{N} (N-n+1)\sigma_{i_n},$$

where $\sigma_{i_1} \geq \ldots \geq \sigma_{i_N}$.

Proof. To show this proposition, we need first some definitions and observations. Consider the following greedy arrival pattern. A packet of size $L_{\max}$ together with bit-size packets of total size σ_i from session i, $i = 1, \ldots, N$ arrive at time 0. After time 0, bit-size packets from session i arrive at rate ρ_i. Let the sessions 1 to N be scheduled under a strict (non-preemptive) priority rule with session 1 having the highest priority. Let $\tau_i > 0$ be the first time at which the buffer content of session i becomes zero and define $b_i = \tau_i - \tau_{i-1}$, where $\tau_0 = 0$. Note that since the sessions are served in a strict priority order, at time τ_i the buffer contents of sessions 1 to i are zero and therefore at this time the first packet from session $i+1$ is scheduled. Let a_i be the buffer size of session i when the first packet from this session is scheduled. We then have

$$b_i = \frac{a_i}{r - \sum_{j=1}^{i} \rho_j}.$$

This is due to the fact that since the scheduler serves the sessions in strict priority and traffic from sessions 1 to i arrives at rate $\sum_{j=1}^{i} \rho_j$, the rate by which the buffer content of session i is depleted is $r' = r - \sum_{j=1}^{i} \rho_j$. Therefore it will take a_i / r' units of time to empty a buffer content of size a_i. Since traffic arrives greedily and the first packet from session i is served at time τ_{i-1}, we have

$$\begin{aligned} a_i &= L_{\max} + \sigma_i + \tau_{i-1}\rho_i \\ &= L_{\max} + \sigma_i + \left(\sum_{l=1}^{i-1} b_l \right) \rho_i \\ &= L_{\max} + \sigma_i + \left(\sum_{l=1}^{i-1} \frac{a_l}{r - \sum_{j=1}^{l} \rho_j} \right) \rho_i \end{aligned} \tag{30}$$

Assume next without loss of generality that $\sigma_1 \geq \ldots \geq \sigma_N$, and consider again the greedy arrival pattern. Let either the $NPEDF$ or the $T(PEDF)$ policy be applied. Let $D_1 = C$ and define the differences $D_{i+1} - D_i$ large enough so that both $NPEDF$ and $T(PEDF)$ schedule the sessions in a strict priority order (1 to N) in the interval $[0, \tau_N)$. According to the discussion in the previous paragraph, the buffer requirements of session i are at least a_i.

Let us now assume that $\rho_i = \epsilon^{i-1}(1-\epsilon)r$. Taking $\epsilon \to 0$, it can be seen from (30) by an inductive argument that

$$\lim_{\epsilon \to 0} a_i = iL_{\max} + \sum_{j=1}^{i} \sigma_j .$$

Therefore,

$$B_{SE}^{\alpha} \geq \sum_{i=1}^{N} iL_{\max} + \sum_{i=1}^{N} \sum_{j=1}^{i} \sigma_j = \frac{N(N+1)}{2} L_{\max} + \sum_{i=1}^{N} (N-i+1)\sigma_i .$$

□

The question now arises whether one can design policies for semi-flexible allocation, that have buffer requirements lower than $O(N^2)$. We show next that this is indeed the case. Specifically,

we construct delay-optimal policies with buffer requirements $O(N)$. The design is based on the following lemma.

Lemma 3 *Let $\vec{D} = \{D_1, \ldots, D_N\}$ be a vector that satisfies the conditions of Theorem 4, and in addition the last inequality is strict:*

$$NL_{\max} + \sum_{n=1}^{N} \sigma_{i_n} \quad < \quad D_{i_N}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_{i_n} D_{i_n}.$$

Then, we can find a vector $\vec{D}' = \{D'_1, \ldots, D'_N\}$ such that $D'_i \leq D_i$, $i = 1, \ldots, N$, and in addition $\vec{D}'$ satisfies the conditions of Theorem 4 with equality for the last constraint.

Proof. Let us assume without loss of generality that $D_1 \leq \ldots \leq D_N$. Let K be the smallest index such that

$$\min\{k+1, N\}L_{\max} + \sum_{n=1}^{k} \sigma_n < D_k\left(r - \sum_{n=1}^{k-1} \rho_n\right) + \sum_{n=1}^{k-1} \rho_n D_n, \quad K \leq k \leq N.$$

Let

$$\epsilon := \min_{K \leq k \leq N} \left\{ \frac{D_k\left(r - \sum_{n=1}^{k-1} \rho_n\right) + \sum_{n=1}^{k-1} \rho_n D_n - \left(\min\{k+1, N\}L_{\max} + \sum_{n=1}^{k} \sigma_n\right)}{r - \sum_{n=1}^{K-1} \rho_n} \right\} > 0.$$

Define a new vector $\vec{D}^{(1)}$ as follows. $D_i^{(1)} = D_i$, $i = 1, \ldots, K-1$, and $D_i^{(1)} = D_i - \epsilon$, $i = K, \ldots, N$. It is easy to see that the vector $D^{(1)}$ satisfies the inequalities of Theorem 4, and that for some k, $K \leq k \leq N$, one of them is met with equality. We will show that in addition, $D_1^{(1)} \leq \ldots \leq D_N^{(1)}$. The case $K = 1$ is trivial. Assume now that $K > 1$. Since $D_1 \leq \ldots \leq D_N$, it is sufficient to show that $D_{K-1} \leq D_K - \epsilon$. Notice first that from the definition of K we have that

$$\min\{K, N\}L_{\max} + \sum_{n=1}^{K-1} \sigma_n = D_{K-1}\left(r - \sum_{n=1}^{K-2} \rho_n\right) + \sum_{n=1}^{K-2} \rho_n D_n. \tag{31}$$

If $D_K - \epsilon < D_{K-1}$, using the definition of ϵ we would have

$$\begin{aligned} \min\{K+1, N\}L_{\max} + \textstyle\sum_{n=1}^{K} \sigma_n \quad &\leq \quad (D_K - \epsilon)\left(r - \textstyle\sum_{n=1}^{K-1} \rho_n\right) + \textstyle\sum_{n=1}^{K-1} \rho_n D_n \\ &< \quad D_{K-1}\left(r - \textstyle\sum_{n=1}^{K-2} \rho_n\right) + \textstyle\sum_{n=1}^{K-2} \rho_n D_n, \end{aligned}$$

which contradicts (31).

If

$$NL_{\max} + \sum_{n=1}^{N} \sigma_{i_n} \quad = \quad D_N^{(1)}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_n D_n^{(1)},$$

we then set $\vec{D}' = \vec{D}^{(1)}$. Otherwise define $K^{(1)}$ as the smallest integer such that

$$\min\{k+1, N\}L_{\max} + \sum_{n=1}^{k} \sigma_n < D_k^{(1)}\left(r - \sum_{n=1}^{k-1} \rho_n\right) + \sum_{n=1}^{k-1} \rho_n D_n^{(1)}, \quad K^{(1)} \leq k \leq N,$$

and create another vector $\vec{D}^{(2)}$. Note that since by construction the vector $\vec{D}^{(1)}$ satisfies one of the inequalities with equality for some k, $K \leq k \leq N$, we have that $K^{(1)} > K$. In general, if in the ith step the vector $\vec{D}^{(i)}$ satisfies

$$NL_{\max} + \sum_{n=1}^{N} \sigma_{i_n} \quad = \quad D_N^{(i)}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_n D_n^{(i)},$$

we set $\vec{D}' = \vec{D}^{(i)}$. Otherwise we define $K^{(i)}$ analogously and repeat the process to create a vector $\vec{D}^{(i+1)}$. Since $K^{(i)}$ is increasing in i and is at most N, the iteration will stop in a finite number of steps and at the end we will have the vector $\vec{D}'$. □

Theorem 7 *There is a delay-optimal policy π^* among the class of non-preemptive policies such that for all arrival patterns consistent with (3) and (4),*

$$B_{SE}^{\pi^*} \leq 2NL_{\max} + 2\sum_{i=1}^{N} \sigma_i.$$

Proof. Let $\vec{D}$ be a feasible vector of delays. If $\vec{D}$ satisfies the conditions of Theorem 4 with equality for the last constraint, set $\vec{D}' = \vec{D}$. Otherwise construct the vector $\vec{D}'$ as described in Lemma 3. Therefore, $\vec{D}'$ always satisfies,

$$NL_{\max} + \sum_{n=1}^{N} \sigma_{i_n} = D'_{i_N}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_{i_n} D'_{i_n}.$$

Let π^* be either the $NPEDF$ or the $T(PEDF)$ policy that uses vector $\vec{D}'$ as the vector of delays. Since the vector $\vec{D}'$ is schedulable by design, we have,

$$D_i^{\pi^*}(\vec{\rho}, \vec{\sigma}) \leq D'_i \leq D_i.$$

Using also Theorem 3 we conclude that

$$\rho_i D'_i + \sigma_i + L_{\max} \geq M_i^{\pi^*}(\vec{\rho}, \vec{\sigma})$$

and therefore,

$$\sum_{i=1}^{N} \rho_i D'_i + \sum_{i=1}^{N} \sigma_i + NL_{\max} \geq \sum_{i=1}^{N} M_i^{\pi^*}(\vec{\rho}, \vec{\sigma}). \tag{32}$$

Taking into account Lemma 3 and the fact that $\sum_{i=1}^{N} \rho_i \leq r$, we have

$$\begin{array}{rcl} \sum_{i=1}^{N} \rho_i D'_i & \leq & D'_{i_N}\left(r - \sum_{n=1}^{N-1} \rho_{i_n}\right) + \sum_{n=1}^{N-1} \rho_{i_n} D'_{i_n} \\ & = & NL_{\max} + \sum_{n=1}^{N} \sigma_{i_n} \end{array} \tag{33}$$

Conditions (32) and (33) imply the theorem. □

Note that because of its constructive nature, the proof of Lemma 3 provides a simple algorithm for constructing policy π^*, which is both delay-optimal and has "low" buffer requirements.

Notes:

1. As it affects the buffer requirements at subsequent nodes, it may be of interest to provide a characterization of the burstiness and rate of the session's departing traffic, when an upper bound, D_i, on its delay through the multiplexer is known. From [8, Theorem 2.1], it is known that the departing traffic of session i, $B_i(\tau, t+\tau)$, verifies,

$$B_i(\tau, t+\tau) \leq L_{\max} + \sigma_i + \rho_i D_i + \rho_i t.$$

2. From the previous note and assuming a schedulable vector $\vec{D}$, we can then obtain an upper bound on the burstiness of session i departing traffic. In those cases where $\vec{D}$ satisfies the constraints of Theorem 4 with strict inequalities, it is then possible to reduce this bound following a method similar to that of Lemma 3. The reason is again that in this case, the vector of actual session delay bounds induced by $NPEDF$ or $T(PEDF)$ is smaller (component-wise) than $\vec{D}$. In fact, assume that all the inequalities in Theorem 4 are strict, and following the method of Lemma 3, let $c > 0$ be the largest number such that the vector $\{D_i - c\}_{i=1}^N$ remains schedulable. The $NPEDF$ policy that operates with parameters $\{D_i\}_{i=1}^N$ schedules identically to the one that operates with parameters $\{D_i - c\}_{i=1}^N$ and, therefore, these policies induce the same session delays. However, the latter policy induces delay bounds $\{D_i - c\}_{i=1}^N$ since by the choice of c, $\{D_i - c\}_{i=1}^N$ is schedulable. Therefore a bound on the burstiness of session i traffic is $L_{\max} + \sigma_i + \rho_i(D_i - c)$.

3. The technique in the previous note cannot be applied to policy π^* since by design the parameters of this policy will satisfy one of the constraints in Theorem 4 with strict equality. However, since policy π^* always has smaller delay bounds than the corresponding $NPEDF$ policy, it will also have smaller burstiness bounds for the departing session traffic.

5 Optimality Criteria for Soft Deadlines

5.1 Minimization of Maximum Lateness

In the previous section, we provided the schedulable region of $NPEDF$, $T(PEDF)$ and $PEDF$ under the assumption that the arriving traffic satisfies certain constraints. In this section, we consider the problem of designing scheduling policies when the objective is to keep the lateness of all packets as low as possible. This criterion is of interest in situations where the deadlines represent a desirable time by which the packets should be transmitted, and it is important to transmit each packet as early as possible and in a fair manner relative to the transmission times of the rest of the packets. $PEDF$ is a good policy with respect to this type of objectives in the sense that among all scheduling policies, it minimizes the maximum lateness of all packets under any arrival pattern [11]. However, it is easy to construct arrival patterns for which the $NPEDF$ policy is not optimal with respect to the criterion of minimizing the maximum lateness among the non-preemptive policies. In spite of this, we show in the next Theorem, that $NPEDF$ is still a good policy in the sense that the maximum lateness under $NPEDF$ is at most $L_{\max}/r$ larger than the maximum lateness under $PEDF$ for *any* arrival pattern, i.e., even for traffic streams that do not satisfy the conditions of (3) and (4). Let a_p be the arrival time of packet p. In the rest of this section, to avoid unimportant technical complications we make the assumption that

$$\lim_{p \to \infty} a_p = \infty. \tag{34}$$

Let f_p, $\hat{f}_p$ be the finishing time of packet p under the $PEDF$ and $NPEDF$ policies respectively and let d_p be its deadline.

Theorem 8 *Under any arrival pattern:*

$$\sup_p \left\{ \hat{f}_p - d_p \right\} \leq \sup_p \{ f_p - d_p \} + \frac{L_{\max}}{r}.$$

For the proof of Theorem 8, we need the next lemma and some notation. We assume that packet numbering is according to the order in which packets enter service under the $NPEDF$ policy. Let e_p, $\hat{e}_p$ be the time packet p entered service under the $PEDF$ and $NPEDF$ policies respectively. Let also $W^\pi(t,d)$ denote the amount of work (in bits) with deadline less than or equal to d at time t in a system that employs scheduling policy π. Finally, if under the $NPEDF$ policy, at time t the server is idle or the packet in service has deadline at most d set $w^{NPEDF}(t,d)=0$. Otherwise let $w^{NPEDF}(t,d)$ be the remaining length (in bits) of the packet that is in service at time t-which by definition must have deadline larger than d.

Lemma 4 *For every t and d, and every policy π,*

$$W^{PEDF}(t,d) \leq W^\pi(t,d) \tag{35}$$

$$W^{NPEDF}(t,d) + w^{NPEDF}(t,d) \leq W^{PEDF}(t,d) + L_{\max} \tag{36}$$

Proof. To show (35), note that $W^\pi(t,d) = A(0,t,d) - S^\pi(0,t,d)$, where $A(\tau,t,d)$ is the amount of work with deadline at most d that arrived in the interval $[\tau,t)$, and $S^\pi(\tau,t,d)$ is the amount of work with deadline at most d served under policy π in the interval $[\tau,t)$. It therefore suffices to show that

$$S^{PEDF}(0,t,d) \geq S^\pi(0,t,d).$$

Define $\bar{t}$ as the supremum of times $t' \leq t$ such that at time t', $PEDF$ serves traffic with deadline larger than d or does not serve any traffic, and π serves traffic with deadline at most d. If there is no such time, i.e., in the interval $[0,t)$ $PEDF$ serves traffic with deadline at most d whenever π does so, set $\bar{t}=0$. At time t' sufficiently close but smaller than $\bar{t}$, there is no backlogged traffic with deadline at most d under $PEDF$ (otherwise, by definition $PEDF$ would be serving such traffic). Therefore, $S^{PEDF}(0,\bar{t}^-,d) = A(0,\bar{t}^-,d) \geq S^\pi(0,\bar{t}^-,d)$. In the interval $[\bar{t},t)$, $PEDF$ always serves packets with deadline at most d whenever π does so. Note also that $PEDF$, by definition, is serving these packets at the highest rate (link rate). Therefore, $S^{PEDF}(\bar{t},t,d) \geq S^\pi(\bar{t},t,d)$. We conclude that

$$\begin{aligned} S^{PEDF}(0,t,d) &= S^{PEDF}(0,\bar{t}^-,d) + S^{PEDF}(\bar{t},t,d) \\ &\geq S^\pi(0,\bar{t}^-,d) + S^\pi(\bar{t},t,d) \\ &= S^\pi(0,t,d). \end{aligned}$$

We use induction on the instants at which packets begin service under $NPEDF$ to prove (36) as follows. We assume that the first packet arrives in the system at time 0 and, therefore, $e_1 = \hat{e}_1 = 0$. Relation (36) holds trivially at time $\hat{e}_1$. Assuming that (36) holds up to time $t = \hat{e}_p$ and for all d, we will show that it holds for all t in the interval $(\hat{e}_p,\ \hat{e}_{p+1}]$ and all d, and therefore up to time $t = \hat{e}_{p+1}$ and for all d. Since by (34) $\lim_{p\to\infty} \hat{e}_p = \infty$, we will conclude that (36) holds for all t and d. It fact, it is sufficient to show (36) only for t in $(\hat{e}_p,\ \hat{f}_p]$ since by definition $w^{NPEDF}(\hat{e}_{p+1},d) \leq L_{\max}$ and either $\hat{f}_p = \hat{e}_{p+1}$ or, if $\hat{f}_p < \hat{e}_{p+1}$, then under both policies, $W^{PEDF}(t,d) = W^{NPEDF}(t,d) = 0$ for $t \in [\hat{f}_p,\ \hat{e}_{p+1})$ and $W^{PEDF}(\hat{e}_{p+1},d) = W^{NPEDF}(\hat{e}_{p+1},d)$ (since both policies are work-conserving.)

Furthermore, notice that we need to show (36) only for $t \in (\hat{e}_p,\ \hat{f}_p)$. Indeed if $\hat{f}_p < \hat{e}_{p+1}$, from the argument of the previous paragraph we conclude that (36) holds for $t = \hat{f}_p$ and all d. If on the other hand $\hat{f}_p = \hat{e}_{p+1}$, denoting $W(t^-) := \lim_{s\to t^-} W(s)$, we will have $W^{NPEDF}(\hat{f}_p^-,d) \leq W^{PEDF}(\hat{f}_p^-,d) + L_{\max}$ (the limits exist since both functions of t are piecewise linear) since (36)

holds for $t \in [\hat{e}_p,\ \hat{f}_p)$. Since any arrival that might occur at time $\hat{f}_p$ will increase the corresponding workload under both policies by the same amount, we have

$$W^{NPEDF}(\hat{f}_p, d) \leq W^{PEDF}(\hat{f}_p, d) + L_{\max}.$$

If the next packet to enter service under $NPEDF$, packet $p+1$, has deadline at most d, (36) holds for $t = \hat{f}_p$ since then $w^{NPEDF}(\hat{f}_p, d) = 0$. If on the other hand packet $p+1$ has deadline larger than d, then from the operation of $NPEDF$ and (35) we conclude that $0 = W^{NPEDF}(\hat{f}_p, d) \geq W^{PEDF}(\hat{f}_p, d) = 0$ and (36) follows since $w^{NPEDF}(\hat{e}_{p+1}, d) \leq L_{\max}$.

Let therefore, $t \in [\hat{e}_p,\ \hat{f}_p)$. Under $NPEDF$, the packet with deadline d_p is continually served in the interval $[\hat{e}_p,\ \hat{f}_p)$. That is, for any $d \geq d_p$, the amount of work in the system with deadline at most d, is depleted at the highest rate under non-preemptive EDF. Therefore, (36) holds for $d \geq d_p$ for all $t \in [\hat{e}_p,\ \hat{f}_p)$ provided that it is true at $\hat{e}_p$.

Let now $d < d_p$. Since d_p is the smallest deadline in the system under the non-preemptive EDF policy at time $\hat{e}_p$, by (35), $0 = W^{NPEDF}(\hat{e}_p, d) \geq W^{PEDF}(\hat{e}_p, d) \geq 0$, i.e., there is no work in the system with deadlines less than d_p at time $\hat{e}_p$, under both policies. Then since the same amount of work arrives in the system under both policies and no work with deadline at most d is served by the non-preemptive EDF, we have

$$\begin{aligned} W^{NPEDF}(t,d) &= W^{PEDF}(t,d) + S^{PEDF}(\hat{e}_p, t, d) &&\leq W^{PEDF}(t,d) + (t - \hat{e}_p)r \\ &= W^{PEDF}(t,d) + (\hat{f}_p - \hat{e}_p)r - (\hat{f}_p - t)r &&\leq W^{PEDF}(t,d) + L_{\max} - w^{NPEDF}(t). \end{aligned}$$

□

Proof of Theorem 8. Assume first that $U := \sup_q \{f_q - d_q\} \geq 0$. Since no deadline is missed by more than U under $PEDF$, the scheduler must be able to transmit all the traffic backlogged at time d_p with deadlines at most d_p within an interval of length U. Therefore $W^{PEDF}(d_p, d_p) \leq Ur$, and from (36) we conclude that $W^{NPEDF}(d_p, d_p) + w^{NPEDF}(d_p, d_p) \leq Ur + L_{\max}$. Packets that arrive after time d_p have deadlines larger than d_p and therefore they cannot be scheduled before packet p under $NPEDF$. Therefore, the maximum delay of packet p after time d_p is $(W^{NPEDF}(d_p, d_p) + w^{NPEDF}(d_p, d_p))/r$ which implies that for any packet p,

$$\hat{f}_p - d_p \leq U + \frac{L_{\max}}{r},$$

as desired.

Assume next that $U < 0$. Consider the $PEDF$ and $NPEDF$ policies that operate with packet delay bounds $D_p' = D_p + U$. Notice that these are valid bounds, i.e., $D_p' \geq 0$, since clearly $D_p \geq d_p - f_p$ and, therefore,

$$\begin{aligned} D_p &\geq \inf_q \{d_q - f_q\} \\ &= -\sup_q \{f_q - d_q\} = -U. \end{aligned}$$

Let $d_p' = a_p + D_p' = d_p + U$. Observe that since all delay bounds are decreased by the same amount, the new $PEDF$ and $NPEDF$ policies behaves identically to the original ones and therefore, the finishing times of the packets do not change. Also,

$$U' := \sup_q \left\{ f_q - d_q' \right\} = U - U = 0.$$

Therefore, applying the argument corresponding to the case $U \geq 0$, we have

$$\begin{aligned} \frac{L_{\max}}{r} &= U' + \frac{L_{\max}}{r} \geq \hat{f}_p - d'_p \\ &= \hat{f}_p - d_p - U \end{aligned}$$

i.e., we again have $\hat{f}_p - d_p \leq U + L_{\max}/r$. □

Corollary 2 *If under any arrival pattern the vector of packet deadlines $\{d_i\}_{i=1}^{\infty}$ is schedulable under $PEDF$, then the vector $\{d_i + (L_{\max}/r)\}_{i=1}^{\infty}$ is schedulable under $NPEDF$.*

Proof. Applying Theorem 8 to the $NPEDF$ policy that operates with deadlines $\{d_i+(L_{\max}/r)\}_{i=1}^{\infty}$, we have that

$$\sup_i \left\{ \hat{f}_i - d_i - \frac{L_{\max}}{r} \right\} \leq \sup_i \left\{ f_i - d_i - \frac{L_{\max}}{r} \right\} + \frac{L_{\max}}{r} = \sup_i \{f_i - d_i\} \leq 0.$$

The first inequality follows from Theorem 8 and the fact that the $PEDF$ policy that operates with deadlines $\{d_i + (L_{\max}/r)\}_{i=1}^{\infty}$ schedules identically as the $PEDF$ policy that operates with deadlines $\{d_i\}_{i=1}^{\infty}$. The equality that follows is simply a mathematical equality, while the second inequality is an immediate consequence of the assumption that the vector of packet deadlines $\{d_i\}_{i=1}^{\infty}$ is schedulable under $PEDF$. □

5.2 Lexicographic Optimization.

A stronger optimality criterion than minimizing the maximum lateness, one which relates closer to fairness, is the criterion of lexicographic optimization of packet lateness, which is defined below.

Let $\{l_i\}_{i=1}^n$, $\{u_i\}_{i=1}^n$, be two n-dimensional vectors and let $\pi_l(i)$, $\pi_u(i)$, be index permutations such that

$$l_{\pi_l(1)} \geq \ldots \geq l_{\pi_l(n)}, \qquad u_{\pi_u(1)} \geq \ldots \geq u_{\pi_u(n)}.$$

The vector $\{l_i\}_{i=1}^n$ is called *lexicographically smaller* than the vector $\{u_i\}_{i=1}^n$, denoted as $\{l_i\}_{i=1}^n \leq_{lex} \{u_i\}_{i=1}^n$, if

1. $l_{\pi_l(1)} \leq u_{\pi_u(1)}$
2. $l_{\pi_l(i)} > u_{\pi_u(i)}$ for some $i = 2, \ldots, n$ implies that $l_{\pi_l(j)} < u_{\pi_u(j)}$ for some $j < i$.

Let $\mathcal{V}$ be a set of n-dimensional vectors. Vector $\{l_i^*\}_{i=1}^n \in \mathcal{V}$ is lexicographically optimal in $\mathcal{V}$ if $\{l_i^*\}_{i=1}^n \leq_{lex} \{u_i\}_{i=1}^n$ for all $\{u_i\}_{i=1}^n \in \mathcal{V}$. Note that condition 1 implies that a lexicographically optimal point is also a point that minimizes its maximum coordinate. The opposite is not always true.

The property of the lexicographically optimal vector that relates to fairness is that if one attempts to reduce coordinate i by picking another vector in $\mathcal{V}$, then necessarily another coordinate that is larger than coordinate i will have to be increased (see [3, Section 6.5.2]).

It turns out that if preemptions are allowed, one of the $PEDF$ policies is lexicographically optimal. Specifically, let $PEDF^*$ be the policy that serves preemptively the packets with the earliest deadline

first, and that among the packets with the earliest deadline serves first the packets with the shortest remaining service time, i.e., time to transmit at rate r the remaining bits in the packet. Among packets with the same deadline and the same remaining service time, $PEDF^*$ selects one in an arbitrary fashion. To provide a precise formulation of the optimality of $PEDF^*$, we will assume that the number of arrivals in finite intervals is finite and

$$\limsup_{t\to\infty} \frac{A(0,t)}{rt} < 1,$$

where $A(0,t)$ is the work that arrives to the system up to time t. These constraints imply that the busy periods of any work-conserving policy, as well as the number of packets served within a busy period are finite.

Theorem 9 *Among all policies, $PEDF^*$ minimizes lexicographically the lateness vector of the packets that arrive during any busy period.*

Before proving this Theorem we need the next lemma which is a direct consequence of the above definition of lexicographical ordering.

Lemma 5 *If $\{l_i\}_{i=1}^n \leq_{lex} \{u_i\}_{i=1}^n$, and $\{l_i^1\}_{i=1}^m \leq_{lex} \{u_i^1\}_{i=1}^m$, then*

$$\left\{\{l_i\}_{i=1}^n, \{l_i^1\}_{i=1}^m\right\} \leq_{lex} \left\{\{u_i\}_{i=1}^n, \{u_i^1\}_{i=1}^m\right\}.$$

Let us now define

s_p: Service time of packet p.

$s_p^{\pi}(\tau)$: remaining service time of packet p at time τ under policy π.

Also, for a given policy π, recall the notations e_p^{π}(service start time), f_p^{π} (service completion time) and l_p^{π} (lateness) of packet p.

Proof of Theorem 9. It is known [11], that for every policy π' one can find a work-conserving (non-idling, preemptive) policy π such that $l_p^{\pi'} \geq l_p^{\pi}$ for every p. Therefore, from now on we concentrate on work-conserving policies. The proof is based on the following lemma.

Lemma 6 *Let π be a work-conserving policy and suppose that at time τ during a busy period there are packets p, q, in the system such that either $d_p < d_q$ or, $d_p = d_q$ and $s_p(\tau) < s_q(\tau)$ and policy π schedules packet q first. Then there is a policy π^1 such that*

- *π^1 schedules identically to π in the interval $[0,\tau)$*
- *after time τ, policy π^1 never schedules packet q while packet p is in the system*
- *$\{l_p^{\pi^1}\} \leq_{lex} \{l_p^{\pi}\}$, where $\{l_p^{\pi^1}\}$, $\{l_p^{\pi}\}$ are the lateness vectors of the packets that arrive during the busy period under policies π^1 and π, respectively.*

Proof. To show this, we argue as follows. Denote by $[\tau_k, t_k)$, $k \geq 1$, $\tau = \tau_1 < t_1 < \tau_2 < t_2 < \cdots$, the maximum intervals of time, after time τ, during which π schedules either one of packets p or q. Consider the policy π^1 that rearranges only the scheduling of packets p, q in the intervals $[\tau_k, t_k)$, $k \geq 1$ by scheduling first packet p until it is completely transmitted, and then packet q. Policy π^1 satisfies the first two conditions of the lemma. To show that it also satisfies the third condition, consider the following cases. Note that by construction we have $f_q^{\pi^1} \geq f_q^{\pi}$.

1. $f_q^{\pi^1} = f_q^{\pi}$. In this case, $l_q^{\pi^1} = l_q^{\pi}$. However, by construction of π^1, we have $f_p^{\pi^1} \leq f_p^{\pi}$ and therefore, $l_p^{\pi^1} \leq l_p^{\pi}$. It follows that $\{l_p^{\pi^1},\ l_q^{\pi^1}\} \leq_{lex} \{l_p^{\pi},\ l_q^{\pi}\}$. Since the lateness of the rest of the packets in the busy period do not change, the result follows from Lemma 5.

2. $f_q^{\pi^1} > f_q^{\pi}$. In this case, $f_q^{\pi^1} = f_p^{\pi}$ and $f_p^{\pi^1} < f_p^{\pi}$. We need to distinguish two sub-cases.

   (a) $d_p < d_q$. Then, $l_q^{\pi^1} = f_q^{\pi^1} - d_q < f_p^{\pi} - d_p = l_p^{\pi}$. Also, clearly $l_p^{\pi^1} < l_p^{\pi}$. Therefore, we again have $\{l_p^{\pi^1},\ l_q^{\pi^1}\} \leq_{lex} \{l_p^{\pi},\ l_q^{\pi}\}$.

   (b) $d_p = d_q$. In this case, $l_q^{\pi^1} = l_p^{\pi}$. However, since the remaining service time of packet p does not exceed that of packet q at time τ, we have $f_p^{\pi^1} \leq f_q^{\pi}$, i.e., $l_p^{\pi^1} \leq l_q^{\pi}$. We conclude again that $\{l_p^{\pi^1},\ l_q^{\pi^1}\} \leq_{lex} \{l_p^{\pi},\ l_q^{\pi}\}$.

Using now repeatedly the proposition, it can be shown that if policy π does not schedule packet p according to $PEDF^*$ in the busy period, one can construct a policy that schedules p according to $PEDF^*$ and is no worse than π in the lexicographic sense. Repeating this procedure for all packets within the busy period, we eventually obtain the $PEDF^*$ policy, which by construction is no worse than π in the lexicographic sense. □

The next observation, which follows directly from Theorem 1, shows that $T(PEDF^*)$ is "almost" lexicographically optimal.

Corollary 3 *Let $\{t_p\}_{p=1}^n$ and $\{l_p\}_{p=1}^n$ be the lateness vectors for a given set of arrivals when the service policy is $PEDF^*$ and $T(PEDF^*)$ respectively. Then for each packet arrival p:*

$$l_p < t_p + \frac{L_{\max}}{r}.$$

The natural question to ask at this point is the performance of $NPEDF$ with respect to the lexicographic criterion. The following example shows that for certain arrival patterns the policy is far from being lexicographically optimal.

Example 1. Assume that the server works at rate 1(bits/unit of time) and let packets arrive as follows: At time 0 a maximum size packet with deadline $ML_{\max}$ arrives and at time 0^+, K ($< L_{\max}$) 1-bit packets with deadline $ML_{\max}^-$ arrive. Thereafter, at each time $t = L_{\max}, 2L_{\max}, \ldots, ML_{\max}$, a maximum size packet with deadline $t + L_{\max}$ arrives. Under $PEDF^*$, the K 1-bit packets will be transmitted upon arrival and will depart from the system by time K, and under both $NPEDF$ and $T(PEDF^*)$ the packet arriving at time 0 goes into service upon arrival and stays in service until it departs at time $L_{\max}$. The difference between the two non-preemptive policies manifests itself after time $L_{\max}^+$, when $T(PEDF^*)$ serves the K 1-bit packets, while $NPEDF$ serves the packet with deadline $2L_{\max}$. Observe that in fact under $NPEDF$, all 1-bit packets leave the system after time $(M-1)L_{\max}$. Under $PEDF^*$ the K 1-bit packets will be transmitted consecutively starting from

time 0 and, therefore, their lateness is a most $-(ML_{\max} - K)$. Under $T(PEDF^*)$, the lateness of these packets is at most $-(ML_{\max} - K) + L_{\max}$, as guaranteed by Corollary 3. However, under any of the NPEDF policies, the lateness of the K packets is at least $-L_{\max}$, i.e., the lateness of all the K 1-bit packets has increased by at least $(M-1)L_{\max} - K$ relative to the lateness provided by the (lexicographically optimal) $PEDF^*$ policy. □

The example notwithstanding, $NPEDF$ is almost lexicographically optimal for fixed size packets since as we show in the next proposition, in this case $T(PEDF^*)$ behaves like $NPEDF$. Note that in general two $NPEDF$ policies may differ only by the rules by which packets with the same deadlines are selected for transmission. When, however, all packets have the same length, the resulting packet lateness are identical in *value* (although may differ by packet indices) under any $NPEDF$ policy and, therefore, all these policies have the same performance as far as lexicographic optimization is concerned.

Proposition 6 *For all arrival patterns, $T(PEDF^*)$ behaves behaves like $NPEDF$ when all packet sizes are fixed at L.*

Proof. Suppose not. Then for some sequence of arrivals, there must exist packets p and q with $d_p > d_q$, such that the following two conditions hold:

$$f_p^{T(PEDF^*)} < f_q^{T(PEDF^*)}, \tag{37}$$

i.e., packet p departs before packet q under the tracking policy; and

$$a_q \leq f_p^{T(PEDF^*)} - L. \tag{38}$$

i.e., both packets have arrived before either is scheduled by the tracking policy (a unit service rate is assumed).

Now if $f_p^{PEDF^*} > f_q^{PEDF^*}$, then from (38) the tracking policy must schedule packet q before packet p which contradicts (37). If $f_p^{PEDF^*} \leq f_q^{PEDF^*}$, then by the definition of $PEDF^*$ we must have,

$$a_q \geq f_p^{PEDF^*}. \tag{39}$$

Combining (38) and (39):

$$f_p^{PEDF^*} \leq f_p^{T(PEDF^*)} - L,$$

which contradicts Theorem 1. □

6 Conclusions and Extensions

This paper was motivated by the need to support multiple sessions with varying traffic characteristics and performance requirements in fast packet-switched networks. It addressed the problem of characterizing and designing policies that are optimal in the sense of minimizing buffer and/or delay requirements, under the assumption of commonly accepted traffic constraints. Buffer optimal policies were investigated for three typical memory allocation methods, that represent different trade-offs between efficiency and complexity. The aspect of also minimizing delay was then taken into account, and it was shown that delay and buffer requirements could not be jointly optimized

unless some level of flexibility was available in allocating memory. Delay optimal policies were investigated and the results were used to construct policies that are both delay-optimal and have low (near-optimal) buffer requirements. Finally, the important problem of designing fair policies for users with soft deadlines was also addressed, and optimal or near optimal policies were identified.

The main conclusions of this paper are the following. If the only objective is to have low buffer requirements the fixed allocation mechanism is adequate in practice. If however, good delay performance is also required, fixed allocation leads to large buffer requirements. In contrast, under the semi-flexible allocation, delay-optimal policies with low buffer requirements can be designed. While it is easier to implement $NPEDF$ than $T(PEDF)$, $T(PEDF)$ may be the policy of choice if it is desirable to apportion lateness in packet finishing times in a fair manner.

The class of tracking policies that was introduced in this paper may be of independent interest in other applications. The natural direction in which the results should be extended is to multiple nodes. This has been the focus of [14], which partially addresses some of these issues.

Acknowledgments

We are grateful to the associate editor, Prof. R. L. Cruz, for many suggestions that not only enhanced the overall presentation of the paper, but also helped clarify and improve numerous subtle arguments in the proofs.

References

[1] Framework for providing additional packet mode bearer services. CCITT recommendation I.122, CCITT Subworking Party XVIII/1-2, 1988.

[2] ATM UNI specification version 3.1. Technical report, ATM Forum, September 1994.

[3] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

[4] A. Birman, P. C. Chang, J. S.-C. Chen, and R. Guérin. Buffer sizing in an ISDN frame relay switch. Technical Report RC 14386, IBM Research, IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, August 1989.

[5] A. Birman, H. R. Gail, S. L. Hantler, Z. Rosberg, and M. Sidi. An optimal service policy for buffer systems. *Journal of the ACM*, 42(3):641–657, May 1995. (See also IBM Research Report RC 16641, April 1991).

[6] 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.

[7] I. Cidon, I. Gopal, G. Grover, and M. Sidi. Real-time packet switching: A performance analysis. *IEEE J. Sel. Areas Commun.*, SAC-6(9):1576–1586, December 1988.

[8] R. L. Cruz. A calculus of delay, Part I: Network element in isolation. *IEEE Trans. Inform. Theory*, IT-37(1):114–131, January 1991.

[9] R. L. Cruz. A calculus of delay, Part II: Network analysis. *IEEE Trans. Inform. Theory*, IT-37(1):132–141, January 1991.

[10] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internetworking Research and Experience*, 1(1), 1990.

[11] M. L. Dertouzos. Control robotics: The procedural control of physical processes. In *Proc. IFIP Cong., 1974*, pages 807–813, 1974.

[12] D. Ferrari and D. C. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal Sel. Areas Comm.*, SAC-8:368–379, April 1990.

[13] H. R. Gail, G. Grover, R. Guérin, S. L. Hantler Z. Rosberg, and M. Sidi. Buffer size requirements under longest queue first. *Performance Evaluation*, 18(2), September 1993. (See also Proc. Perf. Dist. Syst. Integr. Commun. Sys., 1992).

[14] L. Georgiadis, R. Guérin, V. Peris, and K. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, 4(4):482–501, August 1996.

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

[16] K. Jeffay and R. Anderson. On optimal scheduling of periodic and sporadic tasks. Technical Report TR 88–11–06, University of Washington, University of Washington, Dept. Comput. Science FR-35, Seattle, WA 98195, November 1988.

[17] K. Jeffay, D. F. Stanat, and C. U. Martel. On non-preemptive scheduling of periodic and sporadic tasks. In *Proc. Real-Time Systems Symposium*, pages 129–139, San Antonio, TX, 1991. IEEE.

[18] A. Lazar and G. Pacifici. Control of resources in broadband networks with quality of service guarantees. *IEEE Communications Magazine*, September 1991.

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

[20] A. K. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks.* PhD thesis, Department of Electrical Engineering and Computer Science, MIT, February 1992.

[21] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing approach to flow control in Integrated Services Networks—The Single Node Case. *ACM/IEEE Transactions on Networks*, 1(3):344–357, June 1993.

[22] G. Sasaki. Input buffer requirements for round robin polling systems. In *Proc. Allerton Conference on Communication, Control and Computing*, 1989.

[23] D. Verma, H. Zhang, and D. Ferrari. Delay jitter control for real-time communication in a packet switching network. In *Proc. TRICOMM'91*, pages 35–46, Chapel Hill, North Carolina, April 1991.

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

[25] L. Zhang. *A New Architecture for Packet Switching Network Protocols.* PhD thesis, Department of Electrical Engineering and Computer Science, MIT, August 1989.

[26] Q. Zheng. *Real-time Fault-tolerant Communication in Computer Networks.* PhD thesis, University of Michigan, 1993.