Applications of Sample Path Analysis to Communication Network Control

Don Towsley*
Department of Computer & Information Science
University of Massachusetts
Amherst, MA 01003, USA

Abstract

In this paper we survey some techniques that are useful for proving the optimality of control procedures for networks. These techniques are based on the comparison of sample path realizations under different procedures in order to show that one performs better than the other. Although sample path analysis is capable of solving only a subset of problems that arise in the design of optimal control procedures, it typically requires a minimal set of assumptions regarding the workload and provides considerable insight into the behavior of different procedures. The other main technique based on Markov decision processes usually requires Markovian assumptions and provides less insight. We review different ways of comparing sample paths based on the ideas of *vector majorization* and *set dominance*. Applications of these two types of comparisons are drawn from problems in routing, internetwork flow control, and scheduling of real-time data. These applications attempt to address the problems of how to choose a state descriptor to be used for comparison purposes and of how to couple sample paths under different policies in order to facilitate the comparisons.

*This work was partially supported by the Office of Naval Research under Contract N00014-87-K-0304. It was performed while the author was on sabbatical at Laboratoire MASI, Université Pierre et Marie Curie, Paris FRANCE.

1 Introduction

The problem of controling communication networks is receiving considerable attention these days. Much of the networking literature is concerned with the design and evaluation of control procedures and protocols for routing, congestion control, flow control, error control, etc. Consequently a number of different methodologies have been developed to address this problem. These include queueing theory, discrete event simulation, Markov decision processes, and methods for analyzing sample paths. The first two of these techniques are solely concerned with evaluating the performance of control procedure whereas the last two are concerned with determining the optimum protocol or properties of that procedure.

In this paper we provide an overview of the last of the four methodologies listed above. Specifically we overview techniques that can be used to analyze sample paths with the goal of proving the optmality of a specific control procedure. Although the class of problems for which these techniques are applicable is small, they form a simple yet powerful methodology. When applicable, this methodology allows general assumptions on the workload, e.g., general arrival processes, generally distributed packet lengths, etc.. Furthermore, it can provide considerable insight into the behavior of the system and into the structure of the optimum control procedure. In many cases, it yields the optimum procedure explicitly. The methodology can also provide insight into the behavior of the system even when it is unable to yield positive results.

It is interesting to compare sample path analysis with the methodology based on Markov decision processes (MDP's). An advantage of MDP's is that they are able to handle subtle tradeoffs in performance that do not show up within sample paths. However, in order to be applied, they usually require somewhat more restrictive assumptions regarding the workload in order to ensure that the system be Markovian. Last, the policy is usually not explicitly rendered without numerical calculations and less insight is gained into the behavior of the system. The reader is referred to excelllent treatments by Ross [14] and Bertsekas [2] for further details on this methodology. trary policy. The choice must be carefully made so that one can conclude that, indeed, the optimum policy is better than the other policy. Second, one has to choose a technique for comparing the two states over time. We will describe two commonly used comparisons in the next section. Third, many problems require that the sample paths under the two policies be carefully coupled so that the comparison will hold uniformly over all sample paths. We will observe in some of our examples that this often requires exponential assumptions. There exist other surveys of sample path analysis techniques as described above (see Walrand [21, Chapter 8] for a comprehensive but terse survey); however there has appeared more formal and complete treatments of different comparison methods which have not fullly entered the sample path analysis literature. Our overview will be based on these new ideas.

Before we conclude this section we briefly mention that there exists a second class of sample path analysis techniques used in network control. These techniques, which we will refer to as *on-line sample path analysis techniques* in order to distinguish it from the off-line sample path analysis of interest to us, are used to set parameters present in control protocols in order to optimize performance. These techniques are concerned with analyzing a sample path in order to obtain a *sample path gradient* [17] so as to determine how the control parameter should be changed. Such analyses have been developed and evaluated for network routing [5, 10], and load balancing in distributed systems [6, 13]. Although an interesting area of research, they will not be covered in this paper.

The remainder of the paper is organized in the following manner. Section 2 describes two different types of comparisons that are useful in sample path analysis. The next two sections, 3 and 4, present applications of these comparison methods to the problem of routing and to the problem of scheduling packets with real-time constraints. The paper concludes with Section 5, a short summary of the paper.

2 Sample Path Comparison Techniques

We begin this section with the introduction of several vectorial comparisons that are commonly used in off line sample path analysis. Following this, we introduce some stochastic orderings based on these vectorial comparisons.

Definition 1 *Vector* $\boldsymbol{X} = (X_1, \cdots, X_K)$ *is said to majorize vector* $\boldsymbol{Y} = (Y_1, \cdots, Y_K)$ *(written* $\boldsymbol{Y} \prec \boldsymbol{X}$*) iff*

$$\sum_{i=1}^{k} \hat{Y}_i \leq \sum_{i=1}^{k} \hat{X}_i, \quad k = 1, \cdots, K-1 \tag{1}$$

$$\sum_{i=1}^{K} \hat{Y}_i = \sum_{i=1}^{K} \hat{X}_i, \tag{2}$$

where the notation $\hat{X}_i$ *is taken to be the i-th largest element of* $\boldsymbol{X}$*.*

There are many applications where it is useful to replace equation (2) with

$$\sum_{i=1}^{K} \hat{Y}_i \leq \sum_{i=1}^{K} \hat{X}_i, \tag{3}$$

In this case, vector $\boldsymbol{X}$ is said to weakly majorize vector $\boldsymbol{Y}$ (written $\boldsymbol{Y} \prec_w \boldsymbol{X}$)

In the case that the components of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are integers, these comparisons have the following useful properties.

Lemma 1 *If $\boldsymbol{Y} \prec_w \boldsymbol{X}$, then*

1. $(\hat{Y}_1, \cdots, \hat{Y}_{k-1}, \hat{Y}_k - 1, \hat{Y}_{k+1}, \cdots, \hat{Y}_K) \prec_w (\hat{X}_1, \cdots, \hat{X}_{k-1}, \hat{X}_k - 1, \hat{X}_{k+1}, \cdots, \hat{X}_K)$,
2. $(\hat{Y}_1, \cdots, \hat{Y}_{K-1}, \hat{Y}_K + 1) \prec_w (\hat{X}_1, \cdots, \hat{X}_{k-1}, \hat{X}_k + 1, \hat{X}_{k+1}, \cdots, \hat{X}_K)$,
3. $(\hat{Y}_1, \cdots, \hat{Y}_{K-1}, \hat{Y}_K - 1) \prec_w \boldsymbol{X}$,
4. $\boldsymbol{Y} \prec_w (\hat{X}_1, \cdots, \hat{X}_{k-1}, \hat{X}_k + 1, \hat{X}_{k+1}, \cdots, \hat{X}_K)$.

Proof. Property 1 corresponds to Lemma 5.D.2 in [12, p. 135]. Properties 2-4 follow in a straightforward manner from the definition of weak majorization. ∎

Majorization also satisfies properties 1 and 2. The reader is referred to Marshall and Olkin [12] for an excellent treatment of majorization and weak majorization.

Another useful ordering among sets is the *dominance relation.*

Definition 2 *Set $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$ is said to dominate set $\boldsymbol{Y} = \{y_1, y_2, \cdots, y_m\}$ (written $\boldsymbol{Y} \prec_d \boldsymbol{X}$) if $n \geq m$ and $\hat{X}_i \geq \hat{Y}_i$, $i = 1, 2, \cdots m$.*

We define the following three vector operations

- $Large(\boldsymbol{X}, k) = \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_k\}, \;\; 0 \leq k \leq n.$
- $Small(\boldsymbol{X}, k) = \{\hat{X}_{n-k+1}, \cdots, \hat{X}_n\}, \;\; 0 \leq k \leq n.$
- $Shift(\boldsymbol{X}, x) = \{\hat{X}_i - x \mid \hat{X}_i \geq x\}.$

The following lemma, proven in [18] gives conditions under which dominance is preserved when set operations, the $Large$ operation, and the $Shift$ operation are performed on $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Lemma 2 *If $\boldsymbol{Y} \prec_d \boldsymbol{X}$, then:*

1. $\boldsymbol{Y} + \{x\} \prec_d \boldsymbol{X} + \{x\}$, *for* $x > 0$,
2. $\boldsymbol{Y} \prec_d \boldsymbol{X} - \{x_n\}$, *when* $n > m$,
3. $\boldsymbol{Y} - \{y\} \prec_d X$, *where* $y \in \boldsymbol{Y}$,

4. $\boldsymbol{Y} - \{y\} \prec_d \boldsymbol{X} - \{x\}$, *where* $x \in \boldsymbol{X}$, $y \in \boldsymbol{Y}$, *and* $x \leq y$,

5. *Assume that* $R = \{x_1, \cdots, x_n\}$ *where* $x_i \geq x_{i+1}$, $1 \geq i < n$ *and* $S = \{y_1, \cdots, y_m\}$ *where* $y_i \geq y_{i+1}$, $1 \leq i < m$. *Then* $R - \{x_k\} \succ S - \{y_j\}$ *for* $k \geq j$,

6. $Shift(\boldsymbol{Y}, x) \prec_d Shift(\boldsymbol{X}, x)$.

7. $\boldsymbol{Y} \prec_d Large(\boldsymbol{X}, |\boldsymbol{Y}|)$.

These majorization and dominance comparisons can be used to define orderings among random variables. We consider the majorization comparison first.

Definition 3 *A real valued function* ϕ *defined on* $I\!R^n$ *is said to be Schur-convex if*

$$x \prec y \Rightarrow \phi(x) \leq \phi(y)$$

We can now define an ordering between two vectorial random variables based on majorization.

Definition 4 *Let* $\boldsymbol{X}, \boldsymbol{Y} \in I\!R^n$ *be random variables. Random variable* $\boldsymbol{X}$ *is said to be larger than* $\boldsymbol{Y}$ *in the sense of Schur-convex order iff*

$$E[\phi(\boldsymbol{Y})] \leq E[\phi(\boldsymbol{X})], \ \forall \text{ Schur-convex } \phi.$$

The ordering is written as $\boldsymbol{Y} \leq_{scx} \boldsymbol{X}$.

A similar class of functions and ordering among random variables exist exist in the context of weak majorization. We will refer to such functions as *weak Schur-convex functions* and to the ordering as *weak Schur-convex order* which will be written as $\boldsymbol{Y} \leq_{wscx} \boldsymbol{X}$. Examples of such functions include $\phi(\boldsymbol{N}) = \sum_{k=1}^{K} N_k$ and more generally $\phi(\boldsymbol{N}) = \sum_{k=1} f(N_k)$ where $f : I\!R \rightarrow I\!R$ is increasing convex (convex in the case of majorization).

In the case of the dominance relation between sets, we define the following ordering.

Definition 5 *Let* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *be random variables that take as values sets of nonnegative valued r.v.'s such that* $|\boldsymbol{Y}| \leq |\boldsymbol{X}|$. *We say that* $\boldsymbol{Y}$ *is less than* $\boldsymbol{X}$ *in the sense of dominance order (written* $\boldsymbol{Y} \leq_d \boldsymbol{X}$*) iff* $\hat{Y}_i \leq_{st} \hat{X}_i$, $1 \leq i \leq |\boldsymbol{Y}|$.

Recall (or see [15], pp. 251-284, for example) that for two r.v.'s X, Y we write $X \geq_{st} Y$ if $Pr(X \geq x) \geq Pr(Y \geq x)$ for all $x \in R$.

We conclude this section with a brief description of the problem of choosing a state description for the purpose of sample path analysis. Unfortunately, there is no simple set of guidelines for choosing an appropriate state. Typically one of the components of the state is chosen to be the metric with which one intends to compare two policies - queue length information at time $t > 0$ if one wishes to establish that some policy minimizes queue lengths, workload information at time $t > 0$ if the objective is to study the effect of a policy on packet delays, number of losses by time $t > 0$ if the objective is to compare the behavior of buffer overflow policies. However, as we will observe in a later example, the inclusion of such a metric may not always be necessary. In many applications it is also necessary to introduce auxiliary information into the state description whose sole purpose is to aid in proving the orderings among the control procedures. Many proofs based on sample path analysis use some form of induction and auxiliary state information is often useful in carrying out the induction step.

3 Applications to Routing

Consider a single arrival stream of packets feeding K identical servers (channels). Assume that the service times form an i.i.d. sequence of exponentially distributed r.v.'s and that each server has a capacity to store B packets (including the one in service). Let $0 < a_1 < \cdots < a_n < \cdots$ be the sequence of arrival times, i.e., the n-th packet arrives at time a_n, and let $\{\tau_n\}_{n=1}^{\infty}$ denote the interarrival times, $\tau_n = a_n - a_{n-1}$, $n = 1, 2, \cdots$, $a_0 = 0$. The packets arrive at a controller which routes them to the different servers. We consider a class of routing policies, Σ, that have instantaneous queue length information available to them and that are required to route jobs to some queue that has available space, if one exists. Define SQ to be the policy that always routes a job to the queue with the least number of jobs. In case of a tie, any rule can be used to choose the destination queue.

Let $\boldsymbol{N}^{\pi}(t) = (N_1^{\pi}(t), \cdots, N_K^{\pi}(t))$ denote the joint queue lengths at time $t > 0$ under policy $\pi \in \Sigma$. Let $L^{\pi}(t)$ denote the number of jobs lost due to buffer overflow under policy π by time t. The following theorem, taken from [20], states that under SQ, the number of jobs that are rejected by any time t is minimized (in a stochastic sense). Moreover, the vector $\boldsymbol{N}^{\pi}(t)$ is shown to be larger than $\boldsymbol{N}^{SQ}(t)$ in the sense of weak Schur-convex order, for any $\pi \in \Sigma$ and all times t. Based on this last result one can immediately conclude that the total number of jobs present in the system at any time t is minimized under the SQ policy.

Theorem 1

$$L^{SQ}(t) \quad \leq_{st} \quad L^{\pi}(t), \tag{4}$$

$$\boldsymbol{N}^{SQ}(t) \quad \leq_{wscx} \quad \boldsymbol{N}^{\pi}(t) \tag{5}$$

for all $\pi \in \Sigma$, $t > 0$ provided that $\boldsymbol{N}^{\pi}(0) =_{st} \boldsymbol{N}^{SQ}(0)$ and $L^{SQ}(0) =_{st} L^{\pi}(0)$.

Proof. We condition on the arrival times, service times, and initial queue lengths. The proof is by induction on event times (i.e. arrival times or departure times), $t_0 = 0, t_1, t_2, \cdots$. Specifically, we will show that

$$L^{SQ}(t) \quad \leq \quad L^{\pi}(t), \tag{6}$$

$$\boldsymbol{N}^{SQ}(t) \quad \prec_w \quad \boldsymbol{N}^{\pi}(t). \tag{7}$$

for all sample paths. We couple the systems at time $t = 0$ so that $L^{\pi}(0) = L^{SQ}(0)$ and $\boldsymbol{N}^{\pi}(0) = \boldsymbol{N}^{SQ}(0)$. Although capital letters are usually reserved to denote random variables, within the proof of the theorem, as well as within all proofs to follow, they also indicate the values of the variables at specific time instants, on a sample path.

To carry out a forward induction we need to couple the systems in the following manner. First, we couple the service completion times at the k-th largest queue under both policies, $k = 1, \cdots, K$. Furthermore, if any queue is empty, we assume that the server serves a fictitious customer and that a customer that arrives to that queue receives the remainder of this service time. The exponential assumption is required here to guarantee that all true service times form a sequence of i.i.d. r.v.'s.

Basis step. By the statement of the theorem, the relations hold for $t = t_0$.

Inductive step. Assume that the relations hold up through $t = t_i$. Clearly they hold for $t_i < t < t_{i+1}$. For $t = t_{i+1}$ we consider the following two cases.

Case 1. Service completion. Relation (6) clearly holds at $t = t_{i+1}$ since there are no losses at service completions. Hence we focus on relation (7). Suppose that the next event is a completion from the k-th largest queue under both policies. If this corresponds to real completions from these queues, then property 1 of Lemma 1 yields relation (7) at $t = t_{i+1}$. If there are no completions or a completion under SQ, then relation (7) follows trivially at time $t = t_{i+1}$ (by application of property 3 of Lemma 1 in the case of a service completion under SQ). Let us now consider a real completion under π but not SQ. Suppose that the completion occured at the k-th largest queue. Since this completion is a fictitious one under SQ it follows that $\hat{N}_k^{SQ}(t_i) = 0$. Then, by the definition of $\hat{N}_k$, we get, $\hat{N}_j^{SQ}(t_i) = 0$ for all $j = k, .., K$. On the other hand, since the service completion is a real one under π, we get $\hat{N}_k^{\pi}(t_i) \geq 1$, which implies

$$\sum_{i=j}^{K} \hat{N}_i^{\pi}(t_i) \geq 1 > \sum_{i=j}^{K} \hat{N}_i^{SQ}(t_i) = 0, \;\; j = k, \cdots, K \tag{8}$$

Also recall that by the induction hypothesis

$$\sum_{i=1}^{j} \hat{N}_i^{\pi}(t_i) \geq \sum_{i=1}^{j} \hat{N}_i^{SQ}(t_i), \;\; j = 1, \cdots, k-1 \tag{9}$$

Since the first $k-1$ queues with the largest queue length at t_i, remain the ones with the largest queue lengths at t_{i+1}, we get

$$\sum_{i=1}^{j} \hat{N}_i^{\pi}(t_{i+1}) = \sum_{i=1}^{j} \hat{N}_i^{\pi}(t_i) \geq \sum_{i=1}^{j} \hat{N}_i^{SQ}(t_i) = \sum_{i=1}^{j} \hat{N}_i^{SQ}(t_{i+1}), \;\; j = 1, \cdots, k-1 \tag{10}$$

Moreover, due to the service completion at the k-th largest queue

$$\sum_{i=j}^{K} \hat{N}_i^{\pi}(t_{i+1}) = \sum_{i=j}^{K} \hat{N}_i^{\pi}(t_i) - 1 \geq \sum_{i=j}^{K} \hat{N}_i^{SQ}(t_i) = \sum_{i=j}^{K} \hat{N}_i^{SQ}(t_{i+1}), \;\; j = k, \cdots, K \tag{11}$$

The last two relations ensure that equation (7) holds at time t_{i+1}.

Case 2. Arrival. SQ routes the customer to the smallest queue and π routes the customer to some arbitrary queue. Clearly the inductive hypothesis $\sum_{i=1}^{K} \hat{N}_i^{\pi} \geq \sum_{i=1}^{K} \hat{N}_i^{SQ}$ guarantees that $L^{\pi}(t_{i+1}) \geq L^{SQ}(t_{i+1})$. Thus, if a job is admitted into the π-system it will also be admitted into the SQ-system. In this case, relation (7) is ensured by property 2 of Lemma 1. On the other hand, if a job is rejected under π then the π-system is full at time t_{i+1} which makes (7) hold trivially at t_{i+1}. This completes the inductive step.

Removal of the conditioning on arrival times and service times completes the theorem. ■

Next, define a cost function of the form

$$\begin{aligned} V_{\alpha,\beta}^{\pi}(\boldsymbol{n}) &= E\left[\int_0^{\infty} e^{-\alpha t} \phi(N_k^{\pi}(t)) dt | \boldsymbol{N}(0) = \boldsymbol{n}\right] \\ &\quad + E\left[\int_0^{\infty} e^{-\beta t} (L^{\pi}(t) - L^{\pi}(t^-)) dt | \boldsymbol{N}(0) = \boldsymbol{n}\right] \end{aligned} \tag{12}$$

for a weak Schur-convex function ϕ, $\alpha, \beta > 0$, $\boldsymbol{n} \in \{0, \cdots, B\}^K$, and $\pi \in \Sigma$. Here, the first term accounts for α-discounted holding costs for jobs that are buffered in the system, whereas the second term accounts for β-discounted loss penalties for jobs that are rejected.

The discounting factors $e^{-\alpha t}, e^{-\beta t}$ guarantee that the above cost function is well defined over an *infinite* horizon (see [2] for example). We assume that the sequence $\{\tau_n\}_{n=1}^{\infty}$ is such that (12) is finite for at least a policy in Σ. The optimality of the SQ policy is established in the following corollary.

Corollary 1 *SQ minimizes the cost function in (12) over all policies in Σ.*

Proof. The proof follows from the definition of $\leq_{wscx}$ and Theorem 1. ∎

It is interesting to note that SQ maximizes throughput at the same time that it minimizes holding costs. This is not typical of control procedures for finite buffer systems. Typically one has to tradeoff throughput and queue lengths.

This result can be easily generalized using similar arguments to the case that the queue capacities vary from server to server (see [20] for details).

A similar result was proven in the case of infinite capacities at all of the servers by Ephremides, etal. [7] using similar ideas. In that paper, the authors also proved that the *cyclic policy* (C) is the optimum policy from among the class of policies that use no information whatsoever. One has to be very careful regarding how to couple the sample paths between C and an arbitrary policy so that a comparison can be made. For example, the simple coupling used in the preceding theorem does not allow one to make any kind of comparison among the joint queue lengths under C and those under some arbitrary policy. The authors make the key observation that it suffices to couple the service processes so that departures occur simultaneously *at all servers* (this is permitted by the exponential service time assumption). In doing so, the joint queue length statistics for the resulting modified system differ from the joint statistics in the real system. However, as the routing decisions do not depend on the joint queue length statistics, the statistics of the individual queue lengths are unaffected. Let $\tilde{\boldsymbol{N}}^{\pi}(t)$ denote the queue lengths for the system where the service processes have been coupled. As mentioned above the individual queue length statistics are unaffected by this coupling, hence

$$\tilde{N}_k =_{st} N_k, \;\; 1 \leq k \leq K.$$

The proof that the cyclic routing policy is the optimum policy consists of using the weak majorization arguments in the previous theorem to establish

$$\tilde{\boldsymbol{N}}^{C}(t) \leq_{wscx} \tilde{\boldsymbol{N}}^{\pi}(t), \;\; t \geq 0. \tag{13}$$

This has the consequence,

$$V^{C}_{\alpha}(\boldsymbol{N}^{C}) \leq V^{\pi}(\boldsymbol{N}^{\pi}) \tag{14}$$

where

$$V^{\pi}_{\alpha}(\boldsymbol{N}) = E\left[\int_0^{\infty} e^{-\alpha t} \sum_{k=1}^{K} f(N^{\pi}_k(t))dt|\boldsymbol{N}(0) = \boldsymbol{N}\right] \tag{15}$$

for all increasing convex functions $f : I\!R \to I\!R$. The reader should observe that the cost function does not depend on the joint queue length statistics.

Foss [8] has applied similar ideas to prove that the optimum routing policy when the controller has *instantaneous information regarding the unfinished work* is one that routes to the server with the smallest amount of unfinished work in the case of an i.i.d. sequence of service times. Here the system state is taken to be $\boldsymbol{U}^{\pi}(t) = (U_1^{\pi}(t), \ldots, U_K^{\pi}(t))$ where $U_k^{\pi}(t)$ is the unfinished work at server k. The proof is complicated in this variation because the elements are real valued and can be incremented by an arbitrary amount at an arrival. Consequently the properties of Lemma 1 are no longer applicable. Foss provides a novel way of coupling the sample paths which circumvents this problem. The reader is referred to [8] for details.

Similar ideas have been applied to the problem of flow control in gateways between low speed LAN's and a high speed MAN (see [19]).

4 Applications to Real-Time Data Communications

In this section we illustrate the usefulness of the dominance relation between sets through an application to the problem of scheduling packets with real-time constraints. Consider a single arrival stream feeding c identical servers. Associated with each customer is a deadline by which it must begin service. Let Σ_1 be the class of nonpreemptive, non-idling scheduling policies that do not use service time information when scheduling packets. Let ML denote the *minimum laxity scheduling policy* (ML), i.e., the non-idling policy that always schedules packets closest to their deadlines provided that they have not missed them, and throws them away if they have missed their deadlines. Under the assumption that packet lengths form an i.i.d. sequence of exponential r.v.'s independent of the arrival times and the deadlines, it is possible to show that ML minimizes the number of packets that miss their deadlines from Σ_1. Let $L^{\pi}(t)$ denote the number of packets that miss their deadlines by time $t > 0$.

Theorem 2

$$L^{ML}(t) \leq_{st} L^{\pi}(t) \tag{16}$$

for all $\pi \in \Sigma_1$, $t > 0$ provided that $L^{ML}(0) =_{st} L^{\pi}(0)$ and the system begins in the same state under both ML and π.

Proof: We need to first define an appropriate state with which to compare ML and some arbitrary policy. Define $\boldsymbol{T}^{\pi}(t) = (t_1^{\pi}(t), \cdots, t_c^{\pi}(t))$ where $t_j^{\pi}(t) = 1$ if server j is busy under π at time t and 0 otherwise. Further define $Q^{\pi}(t)$ to be the number of packets waiting in the system for service at time $t > 0$ and $\boldsymbol{E}^{\pi}(t) = \{E_1^{\pi}(t), \ldots, E_{Q^{\pi}(t)}^{\pi}\}$ to be the deadlines associated with the packets in the queue at that time, i.e., $E_i^{\pi}(t)$ is the deadline associated with the i-th packet in the queue.

The proof is similar in flavor to that in the routing theorem. We condition on the service times, deadlines, and arrival times and establish

$$\boldsymbol{E}^{\pi}(t) \prec_d \boldsymbol{E}^{ML}(t), \tag{17}$$

$$\boldsymbol{T}^{\pi}(t) \prec_d \boldsymbol{T}^{M}L(t) \tag{18}$$

for all sample paths and for $t > 0$. (Note that according to the definition of $\boldsymbol{T}^{\pi}(t)$, the second relation is identical to $\boldsymbol{T}^{\pi}(t) \prec_w \boldsymbol{T}^{M}L(t)$.) The proof is by forward induction and uses the properties of the dominance relation "$\prec_d$". Once these relations have been established, one can deduce that the number of completions under ML is larger than or equal to the number of completins under π. Conservation of flow arguments then yield that the number of losses is less under ML than under π. Removal of the conditioning then yields the desired result. The details of the proof can be found in [18]. ∎

Remark. This proof did not require that the performance metric of interest, $L^{\pi}(t)$ be included as part of the state desciption. Instead, the state description contained auxiliary information regarding the occupancies of the channels and the deadlines of the packets awaiting transmission.

Remark. A similar result can be established for synchronous systems where the packet lengths are constant. This is a suitable model for ATM based systems.

It is worthwhile to observe that, unlike the example dealing with routing, we had to restrict our attention to a class of *non-idling policies.* This occurs in many problems concerned with determining the best *non-preemptive policy* where a decision must be made regarding what packet to schedule into service. In the routing problem, all packets were identical and had no distinguishing characteristics. In the real-time scheduling problem this is no longer true as packets may have different deadlines. Hence, the decision of whether to schedule a customer with some deadline d_1 or to wait for the possible arrival of a customer with deadline $d_2 < d_1$ is difficult to capture using sample path techniques (or other techniques) except in special cases. One special case arises when the deadlines are deterministic. In this case it is not difficult to establish that no purpose is served idling a channel. This result has been established in [3]. Otherwise, the best that one can do is to establish structural properties regarding the optimal policy in the class of *non-preemptive potentially idling policies.* We describe one such result in the context of real-time scheduling.

Let Σ_2 denote the class of nonpreemptive policies that do not use service time information and let Σ_{ML} denote the class of nonpreemptive, service time independent policies that either allows a server to remain idle when there is work in the queue or uses the ML rule to schedule a customer into service. The following result can be established under the assumption of general arival times, general deadlines, and an i.i.d. sequence of service times independent of the arrival times and deadlines.

Theorem 3 *For any policy $\pi \in \Sigma_2$, there exists a policy $\pi^* \in \Sigma_{ML}$ such that $L^{\pi^*}(t) \leq_{st} L^{\pi}(t)$ for $t > 0$ provided that $L^{\pi^*}(0) =_{st} L^{\pi}(0)$ and the system begins in the same state under both π^* and π.*

Proof: Consider any policy $\pi \in \Sigma_2$ and not in Σ_{ML}. Policy $\pi^* \in \Sigma_2$ is constructed so as to emulate the behavior of π in the sense that whenever π schedules a packet into transmission, π^* also schedules a packet into transmission using the ML rule and whenever π allows a channel to remain idle, π^* does so also. Hence the proof consists of coupling the arrival times, and service completion times under the two policies and establishing that π^* always has a packet awaiting transmission whenever π does. Specifically, given a sample realization, a forward induction argument can be used to establish

$$\begin{aligned} \boldsymbol{E}^{\pi}(t) &\prec_d \boldsymbol{E}^{\pi^*}(t), \\ \boldsymbol{T}^{\pi}(t) &= \boldsymbol{T}^{\pi^*}(t). \end{aligned}$$

The desired result follows from this, conservation of flow and removal of conditioning on service times, deadlines, and arrival times. ∎

The lack of a necessity of assuming exponential service times is typical for these kinds of problems.

If we consider the class of *preemptive resume policies*, then it is typically easier to show that the optimum policy will not idle a server. For example, in the context of real-time scheduling, if the deadline is *until the end of service*, then the *earliest deadline policy* (ED) minimizes the losses in a stochastic sense from the class of policies that do not use service time information. Let Σ_p denote this class of policies.

Theorem 4 *Under the assumption that service times form an i.i.d. sequence of exponential r.v.'s, independent of arrival times and deadlines,*

$$L^{ED}(t) \leq_{st} L^{\pi}(t) \tag{19}$$

$\forall \pi \in \Sigma_p$, $t > 0$, when the system begins in the same state at time $t = 0$ under ED and π and $L^{ED}(0) =_{st} L^{\pi}(0)$.

Proof. The proof is similar to those given or sketched earlier. We omit the proof except for two details. First, the service completions are coupled under π and ED. The memoryless property of the exponential r.v. is required to ensure that service times remain exponentially distributed in spite of preemptions. The independence is required to ensure that we can give different packets the same completion times under the two policies.

The second detail corresponds to the appropriate comparison to make between the two policies. In the previous two theorems, the system structure ensured that the number of packets in the system under ED or the idling ED policy was always greater than or equal to the number under the arbitrary policy π. This may no longer be true in our new problem and a correction is required to account for the additional packets that have completed under ED. Instead, we establish the following relations,

$$C^{\pi}(t) \quad \leq \quad C_{ED}(t), \tag{20}$$

$$Small(\boldsymbol{E}^{\pi}(t), |\boldsymbol{E}^{\pi}(t)| + C^{\pi}(t) - C^{ED}(t)) \quad \prec_d \quad \boldsymbol{E}^{ED}(t) \tag{21}$$

where $C^{\pi}(t)$ is the number of completions under policy $\pi \in \Sigma_p$. The remainder of the proof is identical to the proofs of previous two theorems. ∎

Anther interesting application of the dominance relation can be found in the control of videotext systems [11]. Last, the majorization relation is also of use in real-time communications. Bacceli, Liu, and Towsley [1] studied the effect that different scheduling policies have on packet response times in a tandem network of queues using such a comparison. They establish the following relations,

$$\boldsymbol{W}^{FIFO}(n) \leq_{scx} \boldsymbol{W}^{\pi}(n) \leq_{scx} \boldsymbol{W}^{LIFO}(n), \quad \forall \pi \in \Sigma_3$$

where $\boldsymbol{W}^{\pi}(n)$ is a vector containing the response times of the first n packets and Σ_3 is the class of non-preemptive, non-idling policies that use neither deadline nor service time information. From this they were able to argue that the LIFO policy maximizes the fraction of packets that traverse the network by their deadlines provided that the cumulative distribution for the deadlines is concave. The reader is referred to [1] for details.

5 Summary

We have attempted to provide an overview of the methodology available for performing sample path analysis so as to derive properties of optimal control procedures. The focus of the paper has been on two methods for comparing vectors or sets and illustrating their use in several applications to routing and real-time communications. We have attempted also to give some examples on how to choose a state description appropriate for the problem and on how to couple sample paths. This latter problem defies easy solution and the reader is referred to the cited papers for further details.

We have said little regarding the choice of proof techniques. In most of the problems that we have worked on, we have used *forward induction arguments.* Alternate proof techniques include *backward induction* (see [8] for an example) and *interchange arguments* (see [4] for an example).

Last, there is a mathematical theory that forms a basis for sample path coupling that is beyond the scope of this article. However, the key idea behind it is a theorem by Strassen [16] that allows one to couple different systems in such a way that marginal statistics are preserved. The reader is again referred to the cited references for more details on how sample path coupling is undertaken.

References

[1] F. Baccelli, Z. Liu, D. Towsley, "Optimal Scheduling of Parallel Processing Systems With and Without Real-Time Constraints", to appear in *Journal of the ACM.*

[2] D.P.Bertsekas, *Dynamic Programming*, Prentice Hall, 1987.

[3] P. Bhattacharya and A. Ephremides, "Scheduling of Time-Critical Messages," *Proc. 1988 Conf. Inf. Sciences Systems* pp. 623-628, 1988.

[4] C. Buyukkoc, P. Varaiya, J. Walrand, "The $c\mu$ Rule Revisited", *Adv. Appl. Prob.* **17**, pp. 237-238, 1985.

[5] C.G. Cassandras, M.V. Abidi, D. Towsley. "Distributed routing with on-line marginal delay estimation", *IEEE Transactions on Communications*, April 1990.

[6] C.G. Cassandras and J.-I. Lee, "Applications of Perturbation Techniques to Optimal Resource Sharing in Discrete Event Systems", *Proc. 1988 American Control Conf.*, pp. 450-455, 1988.

[7] A.Ephremides, P.Varaiya and J.Walrand, 'A simple dynamic routing problem', *IEEE Trans. on Automatic Control*, vol AC-25, 1980.

[8] S. Foss, "Comparison of Service Disciplines in G/GI/m Queues", INRIA Rapport de Recherche, No. 1097, October 1989.

[9] B.Hajek, 'Optimal control of two interactive service stations', *IEEE Trans. on Aut. Control*, vol AC-29, pp. 491-498, 1984.

[10] P. Heidelberger, D. Towsley. "Sensitivity analysis from sample paths using likelihoods", *Management Science*, **35**, 1475-1488, December 1989.

[11] Z. Liu and P. Nain, "Optimal Scheduling in Some Multi-Queue Single-Server Systems", to appear in *Proc. INFOCOM'90*, June 1990.

[12] A.W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.

[13] S. Pulidas, D. Towsley, J.A. Stankovic. "Imbedding gradient estimators in load balancing algorithms", *Proceedings of Eighth International Conference on Distributed Computing*, pp. 482-490, June 1988.

[14] S.M. Ross, *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, 1970.

[15] S.M.Ross, *Stochastic processes*, John Wiley and Sons, 1983.

[16] "The Existence of Probability Measures with Given Marginals", *Ann. Math. Statist.* **36**, pp. 423-439, 1965.

[17] R. Suri, "Perturbation Analysis: The State of the Art and Research Issues Explained via the GI:G/1 Queue", *Proc. of IEEE*, **77**, 1, pp. 114-136, January 1989.

[18] D. Towsley, S.S. Panwar, "Comparison of Service and Buffer Overflow Policies for Multiple Server Queues that Serve Customers with Deadlines", COINS Technical Report, U. Massachusetts, July 1989.

[19] D. Towsley, S. Fdida, H. Santoso, "Flow Control Protocols for Interconnected High Speed Networks", to appear in *Proc. of NATO Workshop on High Speed Networks*, Sophia-Antipolis, France, 1990.

[20] D. Towsley, P. Sparaggis, C.G. Cassandras, "Stochastic Ordering and Optimal Routing Control for a Class of Finite Capacity Queueing Systems", submitted to 1990 CDC Conference, Honolulu, Hawaii.

[21] J.Walrand, *An introduction to queueing networks*, Prentice Hall, chapter 8, 1988.

[22] R.R.Weber, 'On the optimal assignment of customers to parallel queue', *J. of Applied Prob.*, vol. 15, pp. 406-413, 1978.

[23] W.Whitt, 'Deciding which queue to join', *Oper. Res.*, vol. 34, pp. 55-62, 1986.

[24] W.Winston, 'Optimality of the shortest line discipline', *J. of Applied Prob.*, vol. 14, pp. 181-189, 1977.