On the Ordering Properties of GPS Routers for Multi-Class QoS Provision

Shaogang Chen[a] Kihong Park[a] Meera Sitharam[b]

[a]Department of Computer Sciences, Purdue University
West Lafayette, IN 47907

[b]Dept. of Computer and Information Science and Eng., University of Florida
Gainesville, FL 32611

ABSTRACT

This paper studies the ordering properties of quality of service (QoS) measures and their rendered values in multi-class QoS provision systems when generalized processor sharing (GPS)-based packet scheduling is employed at routers. GPS has been proposed as a building block for providing multiple service classes with differentiated services to applications with diverse QoS requirements. Previous works have concentrated on finding algorithms and implementations that faithfully approximate the fairness properties of GPS with some work done on deriving performance bounds when leaky-bucket traffic shaping is applied at traffic sources.

In this paper, we study the problem of facilitating effective quality of service in multi-class systems with multi-dimensional QoS vectors containing both mean- and burstiness-related QoS indicators such as packet loss rate, delay, and jitter. We study the impact of traffic burstiness and network contention on the properties of rendered QoS in the service classes. We show that under bursty traffic conditions, it is intrinsically difficult for one service class to deliver quality of service superior in both mean- and burstiness-related QoS measures—delay and jitter—over another when GPS scheduling is employed at routers. In particular, to achieve lower mean packet loss and packet loss variance, higher delay variance (i.e., jitter) must be incurred. Conversely, the price for achieving lower mean delay and delay variance is higher packet loss variance. This result is applicable to both reserved and best-effort traffic classes and affects the "programming" of GPS routers for QoS provision by constraining what application QoS requirements can be effectively met.

We show performance results in a specific GPS-based QoS provision platform—noncooperative multi-class QoS provision—which demonstrate the impact of the ordering results on providing effective quality of service to applications with diverse QoS requirements.

Keywords: Multi-class QoS provision, GPS packet scheduling, multi-dimensional QoS vectors and burstiness

1. INTRODUCTION

With the increased deployment of high-speed local- and wide-area networks carrying a multitude of information from e-mail to bulk data to voice, audio, and video, provisioning adequate quality of service (QoS) to the diverse application base has become an important problem.[1–4] This paper studies the ordering properties of QoS indicators and their rendered values in multi-class QoS provision systems when generalized processor sharing (GPS)-based packet scheduling is employed at routers.

GPS or weighted fair queueing[5,6] has been proposed as a building block to providing multiple service classes with differentiated services where a guaranteed fraction of the bandwidth—as determined by the service weight—is allocated to application traffic belonging to a fixed service class. Other things being equal, traffic streams assigned to a service class with a larger weight receive "superior" service than traffic streams assigned to service classes with smaller service weights. When coupled with admission control and traffic shaping via leaky bucket control, performance bounds can be derived for both single node and multiple node systems.[6,7]

Other author information. S. Chen: Supported in part by NSF grant ANI-9714707; chensg@cs.purdue.edu.
K. Park: Supported in part by NSF grants ANI-9714707, ESS-9806741, and grants from PRF and Sprint; park@cs.purdue.edu.
M. Sitharam: Supported in part by NSF grant CCR-9409809; sitharam@cise.ufl.edu.

Much of previous work in GPS-based packet scheduling has concentrated on finding algorithms and implementations that faithfully approximate the fairness properties of GPS.[8,9,5,10–12] This problem is nontrivial due to the fact that in the packet scheduling context, one, packets can be of variable length, and two, the task of scheduling a single packet should be uninterruptable. Variants of weighted fair queueing have been advanced with different approximation and efficiency properties. A survey of packet scheduling disciplines and their relative merits can be found in Zhang.[12]

This paper studies a *performance* aspect of GPS—assuming a perfect weighted fair queueing packet scheduler is available—of the properties of QoS rendered across multiple service classes. Pertinent is the assumption that in the modern networking environment a compendium of real-time and multimedia applications coexist characterized by diverse quality of service requirements as captured by multiple QoS indicators. That is, the QoS requirement of an application is represented by a QoS *vector* rather than a single scalar whose components can include bounds on end-to-end delay, packet loss rate, and their variances (i.e., jitter), among others. A natural question that arises in this context is:

> Given a 2-service class system with service weights $\alpha_1 > \alpha_2 > 0$, $\alpha_1 + \alpha_2 = 1$, and two i.i.d. traffic sources $\xi_1(t)$, $\xi_2(t)$ assigned to service classes 1 and 2, respectively, what is the QoS rendered to $\xi_1(t)$, $\xi_2(t)$ at the GPS switch?

In particular, if $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^s$, $s \geq 1$, denote QoS vectors depicting *rendered* (or *achieved*) QoS at the router, does $\alpha_1 > \alpha_2$ imply $\mathbf{x}^1 < \mathbf{x}^2$? For example, $\mathbf{x}^i$ may contain the components packet loss rate, packet loss variance, mean delay, delay variance as QoS indicators for which smaller values mean "better" QoS; in this case, $\xi_1(t)$ may have been assigned to service class 1 because of its larger service weight and the consequent expectancy—other things being equal—that this may lead to a better service than in class 2. An application of this are network environments where users are allowed to choose the service classes their traffic should be assigned to at routers or switches—a form of programming—where the user's decision is based, in part, on information about the switches' state.

Somewhat surprisingly, under the above assumptions, $\alpha_1 > \alpha_2$ need not imply $\mathbf{x}^1 < \mathbf{x}^2$. In fact, if $\mathbf{x}^i = (x_1^i, x_2^i)$ consists of packet loss rate and packet loss variance (or mean delay and delay variance), then both $x_1^1 < x_1^2 \wedge x_2^1 < x_2^2$ (i.e., $\mathbf{x}^1 < \mathbf{x}^2$ thus totally ordered) and $x_1^1 < x_1^2 \wedge x_2^1 > x_2^2$ (i.e., incomparable) are possible outcomes. Which situation holds depends on two factors—the burstiness of $\xi_i(t)$ and the degree of network contention. The latter, in turn, is determined by the mean traffic intensity $\mathbf{E}(\xi_i)$ relative to the service rate.

An even stronger ordering relation holds if $\mathbf{x}^i = (x_1^i, x_2^i, x_3^i, x_4^i)$ contains packet loss rate, packet loss variance, mean delay, and delay variance. Fixing the enumerated order, if $x_1^1 < x_1^2 \ \wedge \ x_2^1 < x_2^2 \ \wedge \ x_3^1 < x_3^2$ then $x_4^1 > x_4^2$, and if $x_1^1 < x_1^2 \ \wedge \ x_3^1 < x_3^2 \ \wedge \ x_4^1 < x_4^2$ then $x_2^1 > x_2^2$. That is, it is intrinsically difficult to achieve both smaller packet loss variance and delay variance in one service class over another. Letting $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2 \in \mathbb{R}^s$ denote the QoS requirements of the two applications, even if $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2 \in \mathbb{R}^s$ are totally ordered, it is not the case that $\mathbf{x}^1, \mathbf{x}^2$ will be totally ordered. This *QoS mismatch problem* occurs in GPS-based multi-class multi-dimensional QoS vector systems irrespective of whether the service classes are used for transporting reserved (i.e., guaranteed) traffic or best-effort traffic with more flexible QoS requirements.

We illustrate the implications of our results in a specific application domain: stratified best-effort QoS provision in noncooperative network environments. The traditional approach to QoS provision uses resource reservations along a route to be followed by a traffic stream so that the stream's mean data rate and burstiness can be suitably accommodated. Although research abounds,[13–16,2,17,18,4,6,7] analytic tools for computing QoS guarantees rely on shaping of input traffic to preserve well-behavedness across switches which implement some form of packet scheduling discipline such as GPS. Real-time constraints of multimedia traffic and the scale-invariant burstiness associated with self-similar network traffic[19–22] limit the shapability of input traffic while at the same time reserving bandwidth that is significantly smaller than the peak transmission rate. Thus QoS and utilization stand in a trade-off relationship with each other[23,22] and transporting application traffic over reserved channels, in general, incurs a high cost.

This makes it important to organize today's best-effort bandwidth, as examplified by the Internet, into *stratified* services with graded QoS properties such that the QoS requirements of a compendium of applications can be effectively met. This is particularly useful for applications that possess diverse but—to varying degrees—flexible QoS requirements. It would be overkill to transport such traffic over reserved channels. On the other hand, relying on homogenous best-effort service, characteristic of today's Internet, would be equally unsatisfactory. A dual architecture capable of supporting reserved and stratified best-effort service is needed which, in turn, helps amortize the cost of inefficiencies stemming from overprovisioned resources for guaranteed traffic through the filling-in effect.[24]

Recently, microeconomic/game-theoretic approaches to resource allocation have received significant interest with application domains spanning a number of different contexts.[25–32] The overall goal of this area is to formulate a resource allocation problem in the framework of microeconomics and game theory, and show that under certain conditions, the system achieves "desirable" allocations from stability, fairness, and optimality points-of-view. The latter are important in making stratified best-effort bandwidth practically usable by QoS-sensitive applications. In addition, predictable service—both in terms of dynamic stability and the rendering of appropriate QoS—is a crucial prerequisite to feasibly realizing such an architecture.

To demonstrate the relevance of our results in a specific setting, we construct a noncooperative multi-class QoS provision model where users are assumed to be selfish, and packets are routed over switches where—as a function of their enscribed priority—differentiated service is delivered. The diverse spectrum of application QoS requirements is modeled using individual utility functions. Users or applications can choose both the service classes and the traffic volumes assigned to them. The interaction of users behaving selfishly in accordance with their QoS preferences leads to a noncooperative game whose game-theoretic properties are analyzed in Park *et al.*.[33] A distributed architecture for WAN environments and its performance is studied in Chen *et al.*.[34] In this paper, we use our QoS ordering results to illustrate how they affect the provision of service classes with stable QoS properties that match the diverse QoS requirements of heterogenous applications.

The rest of the paper is organized as follows. In Section 2, we describe the network model and overall set-up. This is followed by Section 3 which gives a qualitative analysis of the QoS ordering problem under GPS packet scheduling. Section 4.1 presents a simulation study showing the impact of network contention and burstiness, in particular, self-similar bursty traffic. Section 5 shows the influence of QoS ordering in multi-class multi-dimensional QoS vector systems where users behave selfishly. We conclude with a discussion of our results and future work.

2. NETWORK MODEL

2.1. Switch Model

The switch model is depicted in Figure 2.1. A switch or router is shared by two traffic classes—*reserved* and *nonreserved* (or best-effort)—where the former constitutes background or cross traffic and the latter is the aggregate application traffic. That is, $\lambda^{NR} = \sum_{i=1}^{n} \lambda_i$ where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the mean arrival rates of n application traffic sources. The service rate of the system is given by μ and we will assume that the switch implements a form of GPS with service weights $\alpha_1, \alpha_2, \ldots, \alpha_m$ where $\alpha_j \geq 0$, $j \in [1, m]$, and $\sum_{j=1}^{m} \alpha_j = 1$. Here, m denotes the number of service classes. The total service rate μ is split between the two traffic classes $\mu = \mu^R + \mu^{NR}$. Service class j of the nonreserved traffic class thus receives a service rate of $\alpha_j \mu^{NR}$.

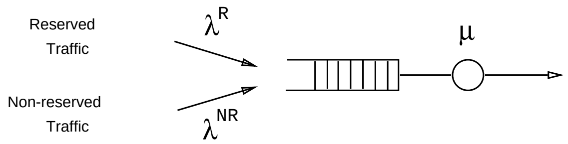

Figure 2.1. Dual traffic classification at output-buffered switch with shared priority queue implementing weighted fair queueing.

2.2. Application Model

We present the application model here for completeness and to formalize the meaning of *heterogenous application preferences*. It is used in Section 5 when describing the simulation set-up of a noncooperative multi-class multi-dimensional QoS provision system and its behavior under the QoS ordering constraint.

Given a generic network model where packets are tagged by priority labels receiving differentiated service at switches, we need a framework and control mechanism which is able to exploit this feature to provide service to applications with diverse QoS needs such that the collective good of the whole system is maximized. A *utility* *function* is a map $U : \mathbb{R}^s \to \mathbb{R}_+$, $s \geq 1$, from QoS vectors to the nonnegative reals indicating the level of satisfaction or utility a certain quality of service brings to an application or user. It is a purely theoretical tool to reason about application behavior assuming certain qualitative shapes about its preferences. Figure 2.2 shows two candidate utility functions, on the left, for "nonurgent" e-mail, and on the right, for a real-time video application. The packet loss rates have been exaggerated for illustrative purposes.

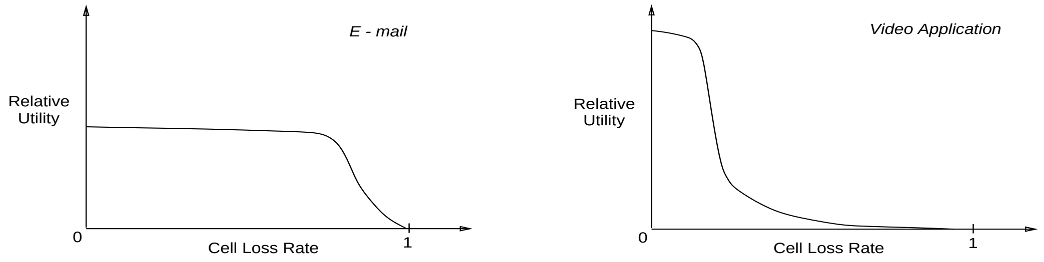

Figure 2.2. Utility functions. E-mail application (left) and video application (right).

The shapes of the utility functions indicate that non-urgent e-mail is much more tolerant to high packet loss, and unless the loss rate is "exceedingly" high, the e-mail application is almost equally satisfied whether the loss rate is 0 or somewhat higher. The video application, on the other hand, can only tolerate much smaller loss rates, and its utility is concentrated toward 0. We will be particularly interested in *step utility* functions where users convey their QoS preference via (upper) bounds on the QoS indicators such as packet loss rate, delay, and jitter.

3. QUALITATIVE ANALYSIS OF QOS ORDERING PROPERTIES

3.1. Problem Statement

As mentioned in the Introduction, the essence of the QoS ordering problem can be posed in the context of a 2-service class system: Given a 2-service class system with service weights $\alpha_1 > \alpha_2 > 0$, $\alpha_1 + \alpha_2 = 1$, and two i.i.d. traffic sources $\xi_1(t)$, $\xi_2(t)$ assigned to service classes 1 and 2, respectively, what is the QoS rendered to $\xi_1(t)$, $\xi_2(t)$ at the GPS switch? In particular, if $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^s$, $s \geq 1$, denote QoS vectors depicting *rendered* (or *achieved*) QoS at the router, does $\alpha_1 > \alpha_2$ imply $\mathbf{x}^1 < \mathbf{x}^2$?

Let $\Theta = \{ \boldsymbol{\theta}^i : i \in [1, n] \}$, $\boldsymbol{\theta}^i = (\theta_1^i, \theta_2^i, \ldots, \theta_s^i)^T$, denote the set of QoS requirements for a population of n applications. Given that the QoS rendered by a service class $j \in [1, m]$ is an *induced* phenomenon depending on the total traffic influx q_j to class j, the question arises how well the induced QoS levels match the needs of the constituent application QoS requirements. This is assuming that GPS packet scheduling is used at a switch with service weights ordered $\alpha_1 > \alpha_2 > \cdots > \alpha_m$. As part of the general problem, we are interested in answering a very basic but fundamental question: *If* Θ is totally ordered, can QoS be rendered at the m service classes such that the performance QoS vector set $X = \{ \mathbf{x}^j : j \in [1, m] \}$, $\mathbf{x}^j = (x_1^j, x_2^j, \ldots, x_s^j)^T$, is also linearly ordered? To maintain comparability, we will assume that the n input processes are i.i.d.

To fix a reference point, consider a 2-application/2-service class/2-dimensional QoS vector system with packet loss rate and packet loss variance as the two QoS indicators. We would like to know whether the following implication holds,

$$\alpha_1 > \alpha_2 \implies (c_1, \sigma_1) < (c_2, \sigma_2), \tag{3.1}$$

where $\mathbf{x}^j = (c_j, \sigma_j)$ is the QoS rendered at service class $j \in \{1, 2\}$. For example, if $(\theta_c^1, \theta_\sigma^1) < (\theta_c^2, \theta_\sigma^2)$ then it is natural for user 1 to assign her traffic to service class 1. Here $\boldsymbol{\theta}^i = (\theta_c^i, \theta_\sigma^i)$ is the QoS requirement of user $i \in \{1, 2\}$

As a second reference point that is more comprehensive, we will be interested in a 2-application/2-service class/4-dimensional QoS vector system where the two additional QoS measures consist of mean delay and delay variance. The corresponding implication to check is

$$\alpha_1 > \alpha_2 \implies (c_1, \sigma_1^c, d_1, \sigma_1^d) < (c_2, \sigma_2^c, d_2, \sigma_2^d) \tag{3.2}$$

where the first two components are as before and the last two components represent mean delay and delay variance*, respectively. As before, if $(\theta_c^1, \theta_{\sigma^c}^1, \theta_d^1, \theta_{\sigma^d}^1) < (\theta_c^2, \theta_{\sigma^c}^2, \theta_d^2, \theta_{\sigma^d}^2)$ then it is natural to assign user 1's traffic to service class 1.

3.2. QoS Ordering of Packet Loss

First, we give a qualitative analysis of the packet drop mean/variance ordering question, i.e., implication (3.1). The queueing set-up is the one shown in Figure 2.1 (Section 2.1) with our two applications comprising the nonreserved or best-effort traffic.

Let $\xi(t)$ denote the (discrete time) stochastic process corresponding to the reserved cross traffic with mean $\mathbf{E}(\xi) = \lambda^R$. In the following development, we will assume a *zero buffer capacity* switch where the degree of contention is solely determined by the *instanteneous* packet arrivals. We will model reservedness by assuming $\xi(t) \le \mu$ and

$$\eta(t) = [\, \mu - \xi(t)\,]^+ \tag{3.3}$$

where $[\,\cdot\,]^+ \equiv \max\{\,\cdot\,, 0\,\}$, and η is the available service rate to the nonreserved traffic class—itself a stochastic process determined by ξ. Our goal is to ascertain the influence of the cross traffic process $\xi(t)$—both its mean and variance—on the aforementioned ordering questions. Burstiness may also stem from the application traffic itself, however, in the present context, we will view $\xi(t)$ as the sole control variable.

The instantaneous packet loss rates of the rendered service class QoS vectors $\mathbf{x}^j = (c_j, \sigma_j)$, $j = 1, 2$, can be expressed as

$$c_j(t) = [\, 1 - \alpha_j \eta(t)/q_j \,]^+. \tag{3.4}$$

Here, we have used the isolatedness property of GPS. Thus $c_j(t)$ is a stochastic process with $0 \le c_j(t) \le 1$. Note that the packet loss rate rendered by service class j is determined by its traffic volume q_j and therefore its "relative goodness" vis-à-vis other service classes is determined by the *normalized weight* $\omega_j = \alpha_j/q_j$, $j = 1, 2$. As mentioned above, to maintain comparability, we need the input processes q_1, q_2 to be the same. To further condense the problem to its essentials—namely dependence of QoS ordering on ξ—we set $q_1 = q_2 = q^*$ where q^* is constant.

Since, by assumption, q_j is fixed, we may assume without loss of generality that

$$\omega_1 \ge \omega_2.$$

That is, service class 1 is "better" than service class 2, certainly with respect to packet loss rate since $c_1(t) \le c_2(t)$, $\forall t \in \mathbb{R}_+$, which follows from (3.4). This also trivially implies

$$\mathbf{E}(c_1) \le \mathbf{E}(c_2).$$

The variance, however, is more tricky. Let $\mathbf{V}$ denote the variance operator. Then

$$\mathbf{V}(c_j) \;=\; \int_{\eta \le \frac{1}{\omega_j}} p(\eta)(1 - \omega_j \eta)^2 d\eta \;-\; \mathbf{E}(c_j)^2 \tag{3.5}$$

since for $\eta \le 1/\omega_j$, $c_j = 1 - \omega_j \eta$. By $\omega_1 \ge \omega_2$, the second moment term in (3.5) satisfies

$$\begin{aligned} \int_{\eta \le \frac{1}{\omega_1}} p(\eta)(1 - \omega_1 \eta)^2 d\eta \;&\le\; \int_{\eta \le \frac{1}{\omega_1}} p(\eta)(1 - \omega_2 \eta)^2 d\eta \\ &\le\; \int_{\eta \le \frac{1}{\omega_2}} p(\eta)(1 - \omega_2 \eta)^2 d\eta. \end{aligned}$$

Since $\mathbf{E}(c_1) \le \mathbf{E}(c_2)$, the two terms in (3.5) contribute in opposite directions and both $\mathbf{V}(c_1) \le \mathbf{V}(c_2)$ and $\mathbf{V}(c_1) \ge \mathbf{V}(c_2)$ are possible depending on the distribution of η, $p(\eta)$.

*To avoid further cluttering of notation, we depict the *standard deviation* of the packet drop and queueing delay processes while continuing to refer to variances in the text.

If $p(\eta)$ is concentrated toward $\max\{1/\omega_1, 1/\omega_2\}$—i.e., the distribution of ξ is concentrated toward 0—then c_1 and c_2 are close to 0 with high probability. Since $c_1(t) \le c_2(t)$, in the degenerate case when $c_1(t) = 0$, it is certainly possible to have

$$\mathbf{E}(c_1) \le \mathbf{E}(c_2), \qquad \mathbf{V}(c_1) \le \mathbf{V}(c_2) \tag{3.6}$$

as desired in (3.1).

Let us consider the case when $p(\eta)$ is concentrated toward 0, i.e., the distribution of ξ is concentrated toward μ. Under such conditions of high cross traffic, $c_1(t), c_2(t) > 0$ with high probability and we will make the *approximation* $c_j(t) = 1 - \omega_j \eta(t)$. Since $1 - \omega_j \eta(t) = 1 - \omega_j(\mu - \xi(t))$, we have

$$\mathbf{V}(c_j) = \omega_j^2 \mathbf{V}(\xi). \tag{3.7}$$

That is, the variance of the packet loss rate is proportional to the variance of the cross traffic process with constant of proportionality ω_j^2.

By $\omega_1 \ge \omega_2$, we now have $\mathbf{V}(c_1) \ge \mathbf{V}(c_2)$. Assuming strict inequality $\omega_1 > \omega_2$ between the two service classes, we get

$$\mathbf{E}(c_1) < \mathbf{E}(c_2), \qquad \mathbf{V}(c_1) > \mathbf{V}(c_2). \tag{3.8}$$

That is, the apparently "superior" service class 1 has a higher variance than service class 2 although it still has a smaller mean packet loss rate. Returning back to the original question of whether (3.1) can be achieved assuming $\omega_1 > \omega_2$, we conclude that under *high cross traffic conditions*, $c_1 < c_2$ but $\sigma_1 > \sigma_2$, and $\mathbf{x}^1 = (c_1, \sigma_1)$ and $\mathbf{x}^2 = (c_2, \sigma_2)$ become incomparable.

3.3. Numerical Estimation

Although the closed forms of the mean and variance of c_j are, in general, difficult to obtain, their numerical approximations are straightforward to compute assuming the distribution of the cross traffic process ξ is well-behaved. Here, we show the transition behavior, (3.6) $\mapsto$ (3.8), as a function of mean cross traffic when the background traffic process is Poisson with rate λ^R. Since $\mathbf{V}(c_j) = \mathbf{E}(c_j^2) - \mathbf{E}(c_j)^2$, we compute the first moment using

$$\mathbf{E}(c_j) = \sum_{k=0}^{\infty} [1 - \omega_j(\mu - k)]^+ \frac{e^{-\lambda^R} {\lambda^R}^k}{k!},$$

and similarly for the second moment $\mathbf{E}(c_j^2)$.

Figure 3.1 (left) and (middle) plot the estimated mean and variance values as a function of λ^R. We have used the parameter set $\alpha_1 = 0.7$, $\alpha_2 = 0.3$, $q_1 = q_2 = 450$ (thus giving $\omega_1 > \omega_2$), $\mu = 900$, with λ^R ranging from 10 to 500. Since ξ is Poisson, $\mathbf{E}(\xi) = \mathbf{V}(\xi) = \lambda^R$. Figure 3.1 (left) shows that mean packet loss is ordered as $\mathbf{E}(c_1) < \mathbf{E}(c_2)$ as expected. In Figure 3.1 (middle) we observe that up until $\lambda^R \approx 240$ when $\mathbf{E}(c_1) = 0$, we have $\mathbf{V}(c_1) < \mathbf{V}(c_2)$, mainly due to the fact that $\mathbf{E}(c_1) = 0$ for most of the interval. However, after $\lambda^R > 200$, approximately in tandem with $\mathbf{E}(c_1)$ becoming positive, $\mathbf{V}(c_1) > 0$, and after $\lambda^R > 250$, we have

$$\mathbf{V}(c_1) > \mathbf{V}(c_2)$$

as predicted by the analysis. Notice that the transition is fairly abrupt with $\mathbf{V}(c_1) < \mathbf{V}(c_2)$ holding mostly for the degenerate case when $\mathbf{E}(c_1) = 0$, i.e., $c_1(t) = 0$.

One drawback of using Poisson cross traffic to discern the burstiness effect is that the mean and variance are the same (λ^R) and thus cannot be independently varied. For illustrative purposes, we use a white Gaussian noise background traffic process where the mean and variance of the process can be *independently* varied. We stress that this is not meant to be taken as a realistic traffic model (we study the impact of self-similar cross traffic in Section 4.1) but as a generic tool to discern the effect of burstiness on the packet loss ordering relation.

If we vary the mean of the cross traffic process, it turns out to have a "sigmoidal" shape as in Figure 3.1 (middle) of the Poisson cross traffic case. That is, the overall contention level as determined by the average input rate is

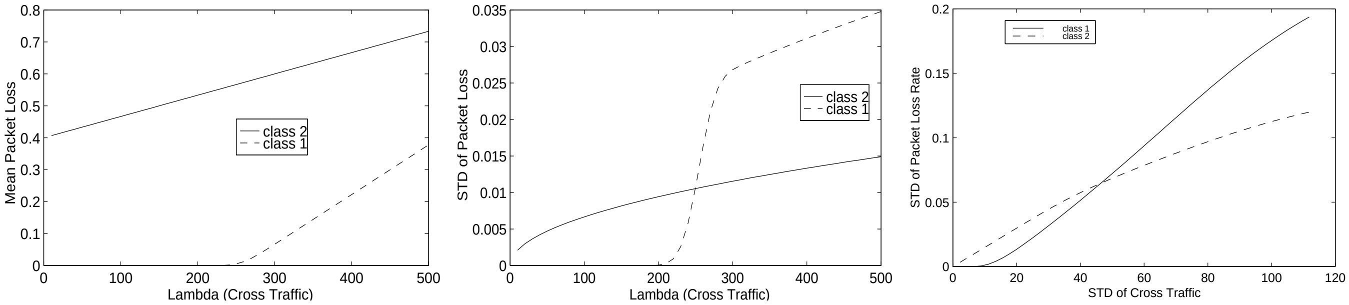

Figure 3.1. Left: Estimated mean packet loss of 2-service class system as a function of Poisson cross traffic parameter λ^R. Middle: Estimated standard deviation of packet loss rate of 2-service class system as a function λ^R. Right: Estimated standard deviation of packet loss rate as a function of standard deviation of white noise cross traffic.

of import; this is discussed further in Section 4.1. If we set the mean at a level where the sigmoidal transition is just beginning to happen and—keeping the mean fixed—vary the standard deviation of the cross traffic process, we observe the reordering phenomenon shown in Figure 3.1 (right). That is, as the standard deviation of the cross traffic process increases, the fluctuation of the packet drop process as experienced by the "better" service class (class 1) exceeds that of service class 2. Furthermore, the spread in variability experienced by the two service classes follows approximately the cone shape predicted by (3.7).

4. SIMULATION STUDY

4.1. Set-Up

In this section, we study the ordering problem of the more comprehensive case, implication (3.2), where the additional QoS measures—mean delay and delay variance (i.e., jitter)—are incorporated. Given the import of network contention on the ordering relation, we study the impact of network resources on QoS ordering. We also incorporate more realistic traffic conditions in the form of self-similar background traffic[35,22] that possess varying degrees of long-range dependence.

The simulation results of this section were carried out using LBNL's Network Simulator *ns* (version 2), suitably modified to transport application/background traffic using modules running on top of UDP. The routing modules were changed to implement an idealized form of GPS (perfect insularity and work conservation), operating on fixed size packets where processing overhead and other efficiency issues are ignored. We implement a topology corresponding to Figure 2.1 with three concurrent connections routed over a bottleneck link. Traffic flow is one-way, and multiplexing takes place at the bottleneck switch where the input traffic from the incoming links impinge. Packet drop, queueing delay, and throughput are measured at the bottleneck switch. Events were recorded at 10 ms granularity.

4.2. Impact of Network Contention

Figure 4.1 (left) shows the traffic profile of two constant bit rate applications (service class 1 & 2), a self-similar background traffic process with long-range dependence captured by a Hurst parameter estimate of 0.75 (service class 0), and their aggregate traffic. Figure 4.1 (middle) shows the packet drop traces at the router for the two service classes where the weights were set at $\alpha_1 = 0.6$, $\alpha_2 = 0.4$. That is, service class 1 is the "better" service class. Figure 4.1 (right) shows the packet drop traces for the same set-up except that the bottleneck link bandwidth was increased from 2.8 Mbps to 3.3 Mbps (keeping the buffer capacity fixed). As is evident from visual inspection of the plots, service class 1 exhibits a smaller mean packet loss rate than service class 2—as expected—since it possesses a larger service weight than service class 2. However, in the case of the variance of the packet drop process, we observe that the *opposite* is true. That is, in spite of the higher service weight $\alpha_1 = 0.6 > \alpha_2 = 0.4$, the variance of the packet drop process in service class 1 is *higher* than the variance in service class 2 (0.08 vs. 0.05 standard deviation); i.e., $\mathbf{E}(c_1) < \mathbf{E}(c_2)$ but $\mathbf{V}(c_1) > \mathbf{V}(c_2)$. This is more clearly shown in Figure 4.2 (top-left) which shows the *measured* standard deviation at the router for the service classes 1 and 2. Clearly, even *in time*, the fluctuation experienced by service class 1 *dominates* that of service class 2.

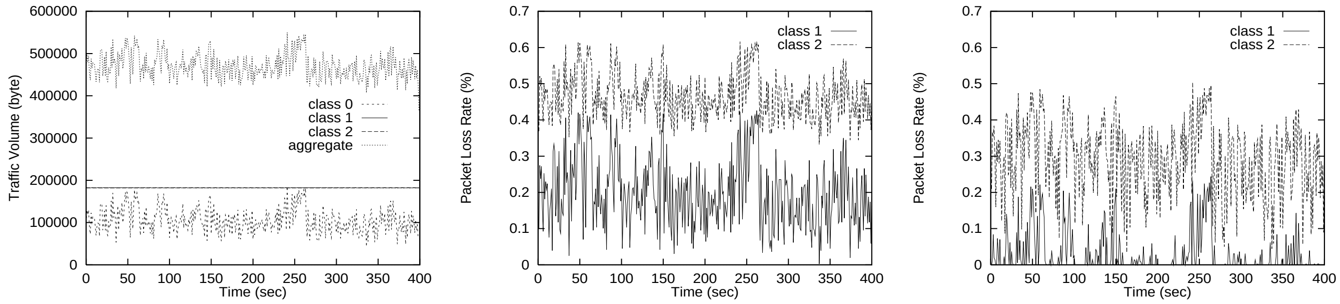

Figure 4.1. Left: Throughput trace of self-similar background traffic (service class 0) and constant application input traffic (service class 1 & 2). Middle: Packet drop trace at bottleneck router with link bandwidth 2.8 Mbps. Right: Packet drop trace at bottleneck router with same set-up except link bandwidth 3.3 Mbps.

Figure 4.1 (right) and Figure 4.2 (top-right) show the corresponding traces for the same set-up except that the bottleneck bandwidth was increased to 3.3 Mbps, decreasing the contention level. The mean standard deviation of service class 1 is now lower than that of service class 2 (0.04 vs. 0.09), and we have $\mathbf{E}(c_1) < \mathbf{E}(c_2)$ and $\mathbf{V}(c_1) < \mathbf{V}(c_2)$. Furthermore, this, again, holds true *in time* (with "high probability") as seen in Figure 4.2 (top-right).

The reasons underlying this phenomenon—dependence on bandwidth (more generally, contention level)—can be traced back to the analysis in Section 3. Pending on whether the probability distribution of the available service rate process η was concentrated toward μ or not, the approximate analysis leading to relation (3.8) could be executed or not. Other things being equal, the more η was concentrated away from μ toward 0—i.e., smaller available bandwidth—the more likely equation (3.7) (i.e., $\mathbf{V}(c_j) = \omega_j^2 \mathbf{V}(\xi)$) holds true, and "switching" of the ordering occurs. Hence varying μ has a similar influence, as does changing the mean background traffic intensity or the mean application traffic intensity. Simply put, the smaller the network resources at a switch relative to the input traffic intensity, the more faithfully the packet drop process will resemble the input process. Since the service class with the larger service weight has greater "exposure" to the input process—inclusive its burstiness—the larger weight service class will also suffer commensurately more under its consequences. Hence the degree of resource contention directly impacts how faithfully this *transfer process* takes place. In the region in-between (in parameter space), either case can occur; however, the shape of the sigmoidal transfer curve (cf. Figure 3.1 (middle)) indicates that the transition point may be sharp and thus the transition region small.

4.3. Delay Ordering

The previous reasoning immediately suggests the corollary that mean delay and delay variance should move in opposite directions from how mean packet loss and packet loss variance are moving. That is, if $\mathbf{V}(c_1) > \mathbf{V}(c_2)$ then $\mathbf{V}(d_1) < \mathbf{V}(d_2)$, and if $\mathbf{V}(c_1) < \mathbf{V}(c_2)$ then $\mathbf{V}(d_1) > \mathbf{V}(d_2)$. This is so since, assuming a "nonnegligible" buffer capacity, if resource contention is high such that $\mathbf{V}(c_1) > \mathbf{V}(c_2)$, then buffer occupancy will be close to saturation, thus suppressing the queueing delay's variability. On the other hand, if resources are plentiful and packet drops miniscule, then much of the variability of the input traffic is absorbed inside the queue, manifesting itself as variability of queueing delay.

Figures 4.2 (bottom-left) and (bottom-right)—which constitute the queueing delay measurements for the runs described earlier—confirm this conclusion. When network contention is high (left column figures), the delay variance ordering is given by $\mathbf{V}(d_1) < \mathbf{V}(d_2)$, the oppositive of the packet loss variance ordering. When the contention level is low (right column figures), we observe $\mathbf{V}(d_1) > \mathbf{V}(d_2)$ which is, again, opposite of what is the case for packet loss variance. The *domination in time* property can be seen to hold for the delay process as well.

4.4. Impact of Self-Similar Burstiness

Self-similar traffic with long-range dependence possess a form of "scale-invariant burstiness".[22] This roughly means that the variances of the time-aggregated processes do not dampen out as the time scale is increased. We have conducted experiments with self-similar background traffic possessing varying degrees of long-range dependence whose Hurst parameter† values were in the range 0.55–0.95. With respect to the ordering relations (3.1), (3.2), we

†The Hurst parameter is one of the ways to measure the long-range correlation structure present in a time series. Its range is (0.5, 1.0), and the closer the Hurst parameter is to 1.0, the more long-range dependent the underlying traffic series.

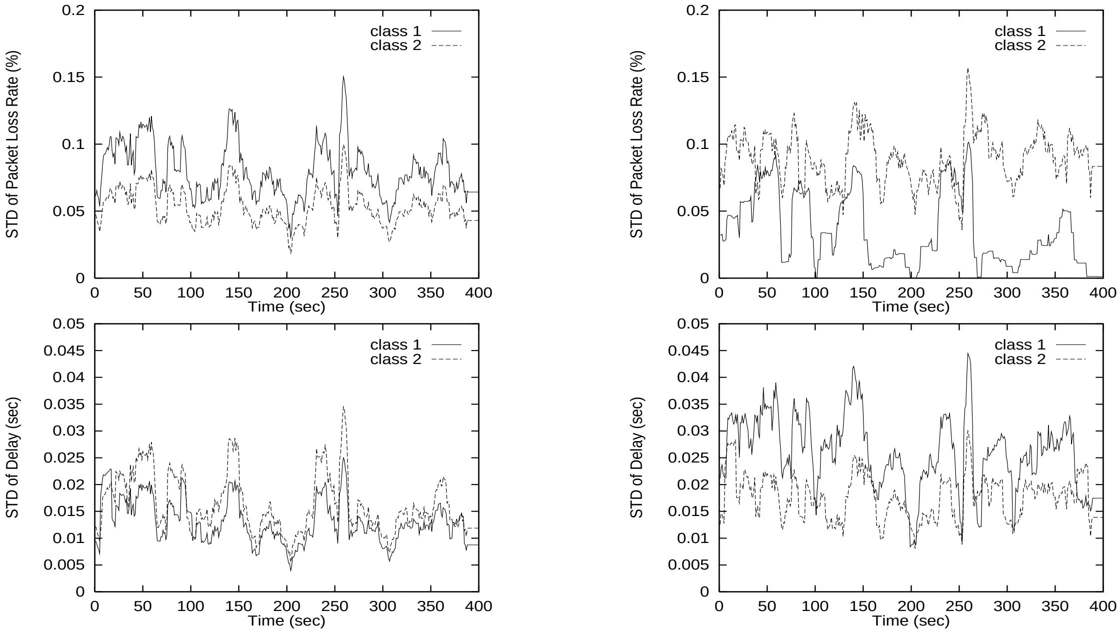

Figure 4.2. Top row: Traces of packet drop standard deviation for bottleneck bandwidth of 2.8 Mbps (left) and 3.3 Mbps (right). Bottom row: Corresponding traces for standard deviation of queueing delay at bottle switch for link bandwidth 2.8 Mbps (left) and 3.3 Mbps (right).

observed ordering behavior consistent with the conclusions advanced above. That is, the scale-invariant burstiness present in self-similar traffic did not have a marked effect on the relative quality of service rendered at the two service classes.

It is important to note that the *sample mean* of all the self-similar background traffic used were held *constant* to preserve comparability. Otherwise, if dissimilar ordering relations were observed it would not be clear to which cause to attribute it to: mean traffic intensity or self-similar burstiness. With this normalization in hand, one may have conjectured to see a switch in the ordering from $\mathbf{V}(c_1) < \mathbf{V}(c_2)$ to $\mathbf{V}(c_1) > \mathbf{V}(c_2)$ as the degree of scale-invariant burstiness was increased (as evidenced in Figure 3.1 (right) for a different context). However, for the resource configurations that we tested, this was not the case. This may be, in part, due to the fact that statistical differences in the variances of the time-aggregated processes are observable only after about the 1–5 sec mark. That is, if one computes the variance of the different Hurst parameter traffic series at the lowest time granularity (10 ms), then the variances are indistinguishable. Similarly up to the 1 sec mark. For the resource configurations that we tested, the correlation structure present at the 1–5 sec time scale and above may not have been significant vis-à-vis the short-range correlations at smaller time scales in influencing queueing behavior. This is also consistent with the discussion of time scale and long-range dependence given in Grosslauser et al.[36] and Ryu et al..[37]

5. APPLICATION TO NONCOOPERATIVE QOS PROVISION GAME

This section presents simulation results of noncooperative multi-class QoS provision games with multi-dimensional QoS vectors. They confirm the transition behavior and ordering results presented above. First, we give a formal definition of the noncooperative multi-class QoS provision problem followed by the simulation set-up.

5.1. Definition of Network QoS Provision Game

Assume we are given m service classes and n applications or players represented by their mean arrival rates $\lambda_1, \ldots, \lambda_n$ and utility functions $U_1, \ldots, U_n$. We arrive at a resource allocation problem in the following way. Let $\lambda_{ij} \geq 0$, $i \in [1, n]$, $j \in [1, m]$, denote the traffic volume of the i'th application assigned to service class j. Thus, $\lambda_i = \sum_{j=1}^{m} \lambda_{ij}$. That is, application i is given the freedom to choose which service classes to assign her traffic to and how much. We also consider the special case when traffic assignments are restricted to be "all in one bag," i.e., $\lambda_{ij} \in \{\lambda_i, 0\}$, for all $j \in [1, m]$.

Let $\Lambda = (\lambda_{ij} : i, j)$ denote the resource assignment matrix, and let $c_1, c_2, \ldots, c_m$ be the packet loss rates of the m service classes. Each packet loss rate is a function of Λ,

$$c_j = c_j(\Lambda), \qquad j \in [1, m].$$

Assuming isolatedness, we have $c_j = c_j(q_j)$ where $q_j = \sum_{i=1}^{n} \lambda_{ij}$ is the total traffic volume assigned to class j. We will also make the assumption that c_j is monotone in q_j, i.e., $dc_j/dq_j \geq 0$, a property satisfied by virtually all service disciplines of interest. Isolatedness and monotonicity will be the only two properties needed of a packet scheduling discipline. We will also make the assumption that $dU_i/dc \leq 0$. That is, making the packet loss rate smaller[‡] can never decrease the utility experienced by player i. The model can be extended to the case when application QoS requirements are represented by multi-dimensional QoS vectors $\mathbf{x} \in \mathbb{R}^s$, $s \geq 1$. For example, in addition to packet loss rate, $\mathbf{x}$ may specify delay requirements as well as restrictions on their fluctuations such as jitter.

The *weighted utility* of application i, given assignment Λ, is defined as $\bar{U}_i(\Lambda) = \sum_{j=1}^{m} \lambda_{ij} U_i(c_j)$. Subject to the above constraints, the static optimization problem can be formulated as

$$\max_{\Lambda} \ \bar{U}(\Lambda) \ = \ \sum_{i=1}^{n} \bar{U}_i(\Lambda). \tag{5.1}$$

This is a nonlinear programming problem with equality constraints.

Any Λ^* that satisfies (5.1) is called *system optimal*. Thus system optimality corresponds to optimizing the usual resource allocation objective function. An assignment Λ^* is *Pareto optimal* if for all Λ,

$$\forall i : \ \bar{U}_i(\Lambda^*) \leq \bar{U}_i(\Lambda) \ \Longrightarrow \ \forall i : \ \bar{U}_i(\Lambda^*) = \bar{U}_i(\Lambda).$$

That is, Pareto optimality states that total utility $\bar{U}$ can only be improved at the expense of one or more individual utility $\bar{U}_i$. In general, Pareto optimality does not imply system optimality. But, clearly, Λ being system optimal implies Λ is Pareto optimal.

The formulation of Nash equilibrium needs a further definition. Given Λ, let $\Lambda_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{im})$ denote the i'th player's assignment vector. Λ_i is also called the *strategy* of player i. Let

$$\mathcal{L}_i(\Lambda) = \{ \Lambda' : \Lambda'_k = \Lambda_k, k \neq i, \text{ and } \|\Lambda'_i\|_1 = \lambda_i \}$$

where $\|x\|_1 = \sum_{j=1}^{m} |x_j|$. That is, $\mathcal{L}_i(\Lambda)$ is the set of all *unilateral* strategies for player i. An assignment Λ^* is a *Nash equilibrium* if $\forall i \in [1, n]$, $\forall \Lambda \in \mathcal{L}_i(\Lambda^*)$,

$$\bar{U}_i(\Lambda) \leq \bar{U}_i(\Lambda^*).$$

That is, in a Nash equilibrium, player i cannot improve its individual utility $\bar{U}_i$ by unilaterally changing its strategy.

In general, a system optimal assignment need not be a Nash equilibrium and little can be said about the relation between system optimality, Pareto optimality, and Nash equilibria. In the context of the noncooperative network environment where every player acts selfishly, we are interested in characterizing assignments that are Nash since they represent stable fixed points of the system—i.e., equilibria. From a resource allocation perspective, we would also like to know under what conditions Nash equilibria are Pareto and system optimal.

5.2. Simulation of Noncooperative QoS Provision Game

We implement the network set-up described in Sections 2.1 and 5.1 with n applications—grouped into several application classes each with a different QoS requirement)—m service classes, and background traffic given by a Poisson process with rate λ^R. Our simulation model is more comprehensive in that it incorporates *pricing* which is used to entice high-QoS applications and low-QoS applications to populate disjoint service classes such that resources are better match and utilized in both the Pareto and system optimality sense. The noncooperative multi-class QoS provision game with pricing is more difficult to analyze, and it is one of the subjects under current study.

We associate prices $p_1, p_2, \ldots, p_m$ with the service classes, and applications incur a cost of $\lambda_{ij} p_j$ for sending a traffic volume of λ_{ij} tagged by service class identifier $j \in [1, m]$. Each application is assigned a one-time budget

[‡]An analogous assumption holds for the multi-dimensional QoS vector case.

B_i “sufficient” for the simulation duration. We also assume that assignments are of the type “all in one bag,” i.e., unsplittable. The *selfishness* behavior of applications is modeled in the following way. Given application i’s QoS requirement vector $\boldsymbol{\theta}^i$, the application seeks out a *cheapest* service class j such that all its QoS requirements are satisfied. That is, $\mathbf{x}^j \leq \boldsymbol{\theta}^i$ and p_j is minimal. Thus, applications are assumed to assign a nonzero utility to “money.” If no such service class j exists—i.e., $\forall j \in [1, m]$, $\mathbf{x}^j \not\leq \boldsymbol{\theta}^i$—then i submits its traffic to a service class j' that most closely meets its QoS requirements, however, paying a price of $p_{j'} + \delta$ where $\delta > 0$ is a bid parameter. The current price of service classes is continuously computed and updated by the system (realized by a computational market that monitors these events), with the new price p'_j set as the maximum of the “bids” submitted in the previous “round.”

The price decrease mechanism is affected in the following way. Let A_j denote the set of applications $i \in [1, n]$ currently assigned to service class $j \in [1, m]$. Let

$$\boldsymbol{\chi}^j = \frac{1}{|A_j|} \sum_{i \in A_j} \boldsymbol{\theta}^i. \tag{5.2}$$

That is, $\boldsymbol{\chi}^j = (\chi^j_1, \ldots, \chi^j_s)^T$ is the average application QoS requirement vector of applications currently assigned to class j. If $\boldsymbol{\chi}^j - \mathbf{x}^j > \mathbf{0}$ and $\|\boldsymbol{\chi}^j - \mathbf{x}^j\| > \Theta$ where $\Theta > 0$ is a system parameter, then

$$p'_j \leftarrow \max\{\, p_j - \delta,\, 0 \,\}.$$

In other words, the system itself exerts a downward pressure on the price of a service class j if the QoS rendered in the service class—i.e., $\mathbf{x}^j$—is significantly better than the QoS required by the constituent applications. Hence, if the system is underutilized, services are rendered at nominal prices or for free. One may use a number of different norms $\|\cdot\|$ (we have used the sup norm) depending on the QoS vector make-up and the objectives at hand.

The asymmetric price adjustment mechanism stems from our work with *many-switch* systems (also called network of switches in[32]) where each user or application makes its QoS requirement known using performance bounds. The QoS requirement vector is then enscribed in the information carried by a packet stream, and routers along a path inspect the QoS requirement vector and a per-connection *rendered* QoS vector (also enscribed in the packet header), and then computes—*on behalf of the application*—which service class to assign the packet to. There are a small set of such managers running at every router whose algorithms are known to the user and who can be accessed by a demux key also specified in the packet header. The design of such managers and the dynamics of the many-switch system leads to interesting distributed control problems which is described elsewhere.[34]

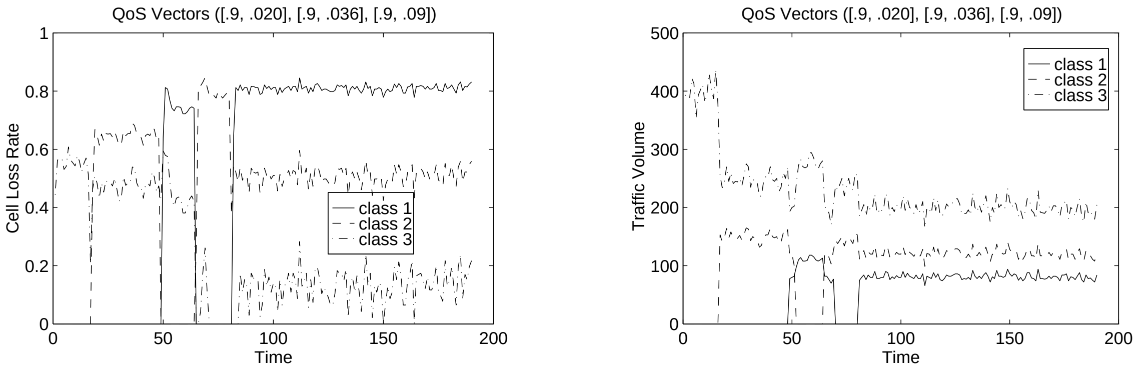

Figure 5.1. Same packet loss requirement but different variance requirements. Left: Cell loss trace shows inverted ordering where service class with least variance has highest cell loss rate. Right: Corresponding traffic volume trace q_1, q_2, q_3.

5.3. Effect of Burstiness

Figure 5.1 shows the trace of a 3-application/3-service class/2-dimensional QoS vector system where the components of the QoS vectors are packet loss rate and its standard deviation. There are 15 applications grouped into three application classes of 5 applications each. The QoS requirements *within* an application class are homogenous;

however, each application acts independently of the others in the same class. The QoS requirements associated with the three application classes are given by $(0.9, 0.02)$, $(0.9, 0.036)$, and $(0.9, 0.09)$. That is, all users have the *same* packet loss bound 0.9 but different bounds on the standard deviation. This allows us to discern the effect of the burstiness-related QoS requirement.

A high packet loss bound was used to create exaggerated, nondegenerate (i.e., non-zero) loss behavior in each of the three classes whose dynamics are easily illustrated. The service weights were set to $\alpha_1 = 0.2$, $\alpha_2 = 0.3$, and $\alpha_3 = 0.5$. The application traffic demands λ_i, $i = 1, 2, \ldots, 15$ were set to 85×5, 49×5, and 46×5. The service rate was $\mu = 900$ and the cross traffic rate was $\lambda^R = 500$.

Figure 5.1 shows the time evolution of packet drops in the three service classes for the system described above. We observe that the applications' bounds on packet loss rate are all satisfied. However, as predicted by the analysis, the application with the most stringent QoS requirement—in both cases requiring a standard deviation bound of 0.02 and 0.021, respectively—ends up receiving the worst *actual* packet loss rate rendered although they are still below the required packet loss rate thresholds. We note that even though for this particular configuration the system settles into a Nash equilibrium after a transient period, if the packet loss requirement 0.9 is decreased to 0.8 (keeping everything else fixed), selfishness—as modeled by the decision procedure above—leads to cyclic behavior.

5.4. Degenerate Assignment

Figure 5.2 shows the trace of a 2-application class/2-service class/2-dimensional QoS vector system with service weights $\alpha_1 = 0.4$, $\alpha_2 = 0.6$. There were a total of 10 applications grouped into two application classes of 5 applications each, with application class QoS requirements $(0.7, 0.01)$, $(0.7, 0.04)$. The traffic volume demands λ_i, $i = 1, 2, \ldots, 10$ were 40×5 and 140×5.

Figure 5.2 (left) shows that service class 1 has both a lower packet loss rate and a lower packet loss standard deviation than service class 2. However, this is only achieved because the packet loss rate for service class 1 is zero or near zero—the degenerate case. Note that in spite of service class 1 having a service weight of $\alpha_1 = 0.4 < \alpha_2$, due to the smaller traffic volume assigned to class 1, $q_1 < q_2$, the normalized service weight satisfies $\omega_1 > \omega_2$ thus explaining the 0 packet loss rate associated with service class 1.

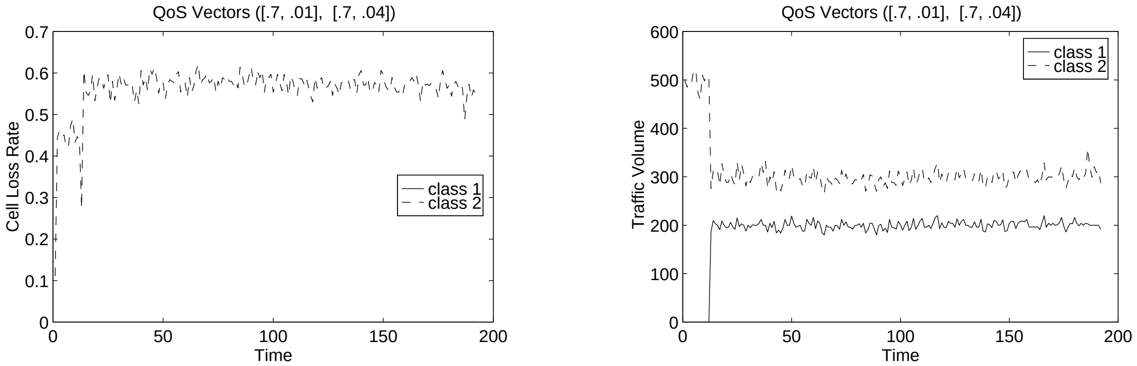

Figure 5.2. Degenerate case where QoS delivered obeys the same order as that required by constituent applications $(0.7, 0.01) < (0.7, 0.04)$. First component is packet loss rate and second component is variance. Left: Shows degenerate QoS rendered for service class 1 where packet loss rate and variance are both 0. Right: Corresponding traffic volume trace.

6. CONCLUSION

We have presented a study of the ordering properties of quality of service (QoS) indicators and their rendered values in multi-class QoS provision systems when generalized processor sharing (GPS)-based packet scheduling is employed at routers. We have shown that under bursty traffic conditions, it is intrinsically difficult for a service class to render superior QoS in both mean- and variance-related QoS measures vis-à-vis some other service class. In particular, considering QoS vectors comprising of mean packet loss, packet loss variance, mean queueing delay, and queueing delay variance, independent of whether network contention is high or not, it is impossible for a service class to deliver better quality of service in each of the QoS measures over some other service class. This has been shown to hold under self-similar traffic conditions with varying degrees of long-range dependence.

We have shown the effect of the QoS ordering properties in an application domain for stratified best-effort QoS provision where users possess heterogenous QoS requirements and behave selfishly to optimize individual performance. We have shown that except in degenerate situations where a service class exhibits near-zero packet loss—and hence near-zero variance—achieving superior QoS in both mean packet loss and packet loss variance is intrinsically difficult. Current work is directed at giving a rigorous treatment of the QoS ordering phenomenon from the queueing theory perspective.

REFERENCES

1. A. Campbell, C. Aurrecoechea, and L. Hauw, "A review of QoS architectures," To appear in *ACM Multimedia Systems Journal* , 1996.
2. A. Elwalid and D. Mitra, "Analysis, approximations and admission control of a multi-service multiplexing system with priorities," in *Proc. IEEE INFOCOM '95*, pp. 463–472, 1995.
3. J. Kurose, "Open issues and challenges in providing quality of service guarantees in high-speed networks," *Computer Communication Review* **23**(1), pp. 6–15, 1993.
4. R. Nagarajan and J. Kurose, "On defining, computing and guaranteeing quality-of-service in high-speed networks," in *Proc. IEEE INFOCOM '90*, pp. 2016–2025, 1992.
5. A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Journal of Internetworking: Res. Exper.* **1**, pp. 3–26, 1990.
6. A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Networking* **1**(3), pp. 344–357, 1993.
7. A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Trans. Networking* **2**(2), pp. 137–150, 1994.
8. J. Bennet and H. Zhang, "Hierarchical packet fair queueing algorithms," in *Proc. ACM SIGCOMM '96*, pp. 143–156, 1996.
9. J. Bennet and H. Zhang, "WF2Q: Worst-case fair weighted fair queueing," in *Proc. IEEE INFOCOM '96*, pp. 120–128, 1996.
10. P. Goyal, H. Vin, and H. Cheng, "Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks," in *Proc. ACM SIGCOMM '96*, pp. 157–168, 1996.
11. A. Greenberg and N. Madras, "How fair is fair queuing?," *Journal of the ACM* **39**(3), pp. 568–598, 1992.
12. H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proceedings of the IEEE* **83**(10), 1995.
13. R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Trans. Inform. Theory* **37**(1), pp. 114–131, 1991.
14. R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE J. Select. Areas Commun.* **13**(6), pp. 1048–1056, 1995.
15. G. de Veciana, G. Kesidis, and J. Walrand, "Resource management in wide-area ATM networks using effective bandwidths," *IEEE J. Select. Areas Commun.* **13**(6), pp. 1081–1090, 1995.
16. A. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control in high speed networks," *IEEE/ACM Trans. Networking* **1**(3), 1993.
17. 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), pp. 482–501, 1996.
18. A. Lazar and G. Pacifici, "Control of resources in broadband networks with quality of service guarantees," *IEEE Network Magazine* , pp. 66–73, October 1991.
19. W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking* **2**, pp. 1–15, 1994.
20. V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," in *Proc. ACM SIGCOMM '94*, pp. 257–268, 1994.
21. W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," in *Proc. ACM SIGCOMM '95*, pp. 100–113, 1995.

22. K. Park, G. Kim, and M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. IEEE International Conference on Network Protocols*, pp. 171–180, October 1996.
23. K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," in *Proc. SPIE International Conference on Performance and Control of Network Systems*, pp. 296–310, 1997.
24. H. T. Kung, T. Blackwell, and A. Chapman, "Credit-based flow control for ATM networks: credit update protocol, adaptive credit allocation, and statistical multiplexing," in *Proc. SIGCOMM '94*, pp. 101–114, 1994.
25. R. Cocchi, S. Shenker, D. Estrin, and L. Zhang, "Pricing in computer networks: motivation, formulation, and example," *IEEE/ACM Trans. Networking* **1**(6), pp. 614–627, 1993.
26. D. Ferguson, C. Nikolaou, and Y. Yemini, "An economy for flow control in computer networks," in *Proc. IEEE INFOCOM '89*, pp. 110–118, 1989.
27. Y. Korilis, A. Lazar, and A. Orda, "Achieving network optima using Stackelberg routing strategies," *IEEE/ACM Trans. Networking* **5**(1), pp. 161–173, 1997.
28. J. Kurose and R. Simha, "A microeconomic approach to optimal resource allocation in distributed computer systems," *IEEE Trans. on Computers* **38**(5), pp. 705–717, 1989.
29. S. Low and P. Varaiya, "A new approach to service provisioning in ATM networks," *IEEE/ACM Trans. Networking* **1**(5), pp. 547–553, 1993.
30. A. Orda, R. Rom, and N. Shimkin, "Competitive routing in multiuser communication networks," *IEEE/ACM Trans. Networking* **1**(5), pp. 510–521, 1993.
31. J. Sairamesh, D. Ferguson, and Y. Yemini, "An approach to pricing, optimal allocation and quality of service provisioning in high-speed networks," in *Proc. IEEE INFOCOM '95*, pp. 1111–1119, 1995.
32. S. Shenker, "Making greed work in networks: a game-theoretic analysis of switch service disciplines," in *Proc. ACM SIGCOMM '94*, pp. 47–57, 1994.
33. K. Park, M. Sitharam, and S. Chen, "Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness," To appear in *Proc. 1st International Conference on Information and Computation Economies* , 1998.
34. S. Chen and K. Park, "A distributed protocol for multi-class QoS provision in noncooperative many-switch systems." To appear in *Proc. International Conference on Network Protocols*, 1998.
35. W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic," in *Proc. ACM SIGCOMM '93*, pp. 183–193, 1993.
36. M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," in *Proc. ACM SIGCOMM '96*, pp. 15–24, 1996.
37. B. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities," in *Proc. ACM SIGCOMM '96*, pp. 3–14, 1996.