Quality of Service Provision in Noncooperative Networks: Heterogenous Preferences, Multi-Dimensional QoS Vectors, and Burstiness

Kihong Park* Meera Sitharam† Shaogang Chen‡
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
{park,sitharam,chensg}@cs.purdue.edu

Abstract

This paper studies the quality of service (QoS) provision problem in noncooperative networks where applications or users are selfish and routers implement generalized processor sharing (GPS)-based packet scheduling. First, we formulate a model of QoS provision in noncooperative networks where users are given the freedom to choose both the service classes and traffic volume allocated, and heterogenous QoS preferences are captured by individual utility functions. We present a comprehensive analysis of the noncooperative multi-class QoS provision game, giving a complete characterization of Nash equilibria and their existence criteria, and show under what conditions they are Pareto and system optimal. We show that, in general, Nash equilibria need not exist, and when they do exist, they need not be Pareto nor system optimal. However, we show that for certain "resource-plentiful" systems, the world indeed can be nice with Nash equilibria, Pareto optima, and system optima collapsing into a single class. Second, we study the problem of facilitating effective QoS in systems with multi-dimensional QoS vectors containing both mean- and burstiness-related QoS measures. We extend the game-theoretic analysis to multi-dimensional QoS vector games and show under what conditions the aforementioned results carry over. Motivated by the same context, we study the impact of burstiness under multiple QoS measures on the properties of the induced QoS levels rendered by the service classes in the system. We show that under bursty traffic conditions, it is, in general, impossible for a service class to deliver quality of service superior in both mean- and burstiness-related QoS measures.

*Supported in part by NSF grants ANI-9714707, ESS-9806741, and grants from PRF and Sprint.

†Supported in part by NSF grant CCR-9409809. Present address: Dept. of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611.

‡Supported in part by NSF grant ANI-9714707.

1 Introduction

1.1 Background

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 [3, 13, 27, 33]. This paper describes a QoS provision architecture suited for best-effort environments, based on ideas from microeconomics and noncooperative game theory.

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 utility functions. Users or applications[1] can choose *both* the service classes and the traffic volumes assigned to them. The interaction

[1] We will use the terms *users*, *applications*, and sometimes, *players*, interchangeably.

of users behaving selfishly in accordance with their QoS preferences leads to a noncooperative game whose dynamic properties we seek to understand.

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 [8, 9, 12, 13, 18, 28, 33, 35, 36, 10], 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 generalized processor sharing (GPS), also known as weighted fair queueing [11, 35]. Real-time constraints of multimedia traffic and the scale-invariant burstiness associated with self-similar network traffic [29, 40, 50, 37] 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 [37] 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 supporing 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 [25].

Recently, microeconomic/game-theoretic approaches to resource allocation have received significant interest with application domains spanning a number of different contexts [7, 15, 16, 19, 21, 24, 26, 30, 34, 39, 43, 44, 47, 48]. 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 stabilibity, fairness, and optimality points-of-view. The latter are important in making stratified best-effort bandwidth practically usable by QoS-sensitive applications: predictable service, both in terms of dynamic stability and the rendering of appropriate QoS, are crucial prerequisites to feasibly realizing such an architecture.

The models and approaches proposed in the literature differ along several dimensions, some of the important ones being whether applications or users are assumed to be cooperative or selfish, whether pricing is used or not, and how much computing responsibility is delegated to the user. Several papers have addressed the issue of multi-class QoS provision in high-speed networks [7, 22, 31, 44, 43, 39]. Some of the works employ a cooperative framework or place significant computing responsibilities on the part of the user [31, 43], some investigate the effect of pricing incentives [7], and others represent flow/congestion control and routing models that only partially address the quality of service problem [22, 34, 44].

Our approach differs from previous works in two significant ways. We explicate the differences in the modeling assumptions and their relevance to network QoS provision, casting our results in this light. First, we give a comprehensive noncooperative resource allocation model formulated to model multi-class QoS provision where users are endowed with heterogenous QoS preferences, they are allowed to choose *both* the service classes and traffic volumes assigned to them, and the properties associated with utility functions are derived from the networking context. The latter leads to non-concave utility functions whose impact on the game structure we analyze.

Second, we study the quality of service provision problem when QoS requirements are generalized to multiple QoS measures such as packet loss rate, mean delay, and their variances (i.e., jitter). The extension of the game-theoretic analysis to multi-dimensional QoS vector games is accompanied by a study of the effect of burstiness on the QoS rendered at different service classes. Burstiness, it turns out, can make it intrinsically difficult to deliver superior quality of service in both mean- and variance-related QoS measures (e.g., packet loss rate and mean delay vs. jitter) in one service class over another when GPS packet scheduling is employed at routers. In other words, the set of realizable QoS vectors over service classes forms a partial order, and it need not possess a top. This somewhat surprising result carries implications to what application QoS requirements can be effectively met in the noncooperative QoS provision architecture, and how routers should configure their services such that a broad spectrum of application QoS requirements can be effectively satisfied.

1.2 Basic Notations and Modeling Assumptions

Our results rely on a set of elementary assumptions which are described next. The formal network QoS provision game is defined in Section 2. We are given n applications or users and m service classes where each user $i \in [1, n]$ has a traffic demand given by its mean data rate λ_i. Each user can choose *where* and *how much* of its traffic to apportion to the m service classes given by its allocation vector $\Lambda_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{im})^T$ where $\lambda_{ij} \geq 0$ and $\sum_j \lambda_{ij} = \lambda_i$.

The QoS achieved in service class $j \in [1, m]$ is determined by a QoS function c_j (e.g., packet loss rate), and c_j is monotone in q_j where $q_j = \sum_i \lambda_{ij}$. The generalization to multi-dimensional QoS vectors is shown in Section 4.1. Each user is endowed with a utility function $U_i(\lambda_{ij}, c_j)$ which indicates the satisfaction received by user i when sending volume λ_{ij} of traffic receiving QoS level c_j through service class j. We assume that U_i is monotone in λ_{ij}, c_j.

The above assumptions are fairly natural given that all that we have said is that the QoS associated with a service class deteriorates when more traffic is pumped into it, users disapprove of bad service quality, and users don't mind sending more if the "cost" is the same. Two simple observations follow from the above. First, since c_j is a function of the allocation vectors $\Lambda_1, \Lambda_2, \ldots, \Lambda_n$, by function composition, U_i is a function of the allocation vectors and the latter constitute the only independent variables. Second, by composition of monotone functions, U_i remains monotone in λ_{ij}. These implied facts will become relevant later.

1.3 Summary of New Results

Our contributions are twofold, one concerning game-theoretic matters, and the other concerning network performance related matters in the context of the noncooperative multi-class QoS provision game. Before we state the results, three notions are of import to their understanding (defined formally in Section 2.3): Nash equilibrium, Pareto optimum, and system optimum.

Roughly speaking, a configuration is a *Nash equilibrium* if each player cannot improve its individual lot through unilateral actions affecting its traffic allocations. Thus if every player finds herself in such a "local optimum," then from the noncooperative perspective, the system is at an impasse—i.e., stable rest point. A configuration is a *Pareto optimum* if in order to improve the lot of some player, the lot of others must be sacrificed. A configuration is *system optimal* if the sum of the individual lots is maximized.

1.3.1 Game-Theoretic

From a game-theoretic perspective, we formulate a model of multi-class QoS provision in noncooperative network environments and analyze the structure of the system with respect to its equilibria and optima.

Nash Equilibria and Existence Conditions We give a complete characterization of Nash equilibria and their existence conditions. We show that Nash equilibria need not exist and we show that this is attributable to the assumptions that U_i is monotone in c_j and c_j is monotone in λ_{ij} (see Section 1.2). Since it is difficult to imagine network systems where this does not hold, existence of Nash equilibria is the exception rather than the norm. Specifically, the monotonicity conditions, unless concocted, do not give rise to U_i that are quasi-concave[2] (much less concave).

Relationship to Pareto and System Optimality We analyze the conditions under which Nash equilibria (if they exist) are Pareto and system optimal. The latter is shown to be related to the Pareto optimality of a certain normal form configuration derived from Nash equilibria. We also show that there are Nash equilibria that are Pareto but not system optimal, and that there exist Nash equilibria that are not Pareto optimal and vice versa.

Resource-Plentiful Systems We show that for certain "resource-plentiful" systems, Nash equilibria, Pareto optima, and system optimal all conincide collapsing into a single class. This item is interesting from the perspective that it gives a sufficient condition under which Nash equilibria are guaranteed to be desirable in the optimality sense. We also show that for resource-plentiful systems a certain self-optimization procedure leads to quick, robust convergence to globally optimal Nash equilibria.

We note that this is the first comprehensive analysis of a noncooperative multi-class QoS provision system where the utility function depends on both the player i as well as the particular service class j where traffic has been assigned. Orda et al. [34] study the only other equally comprehensive QoS provision model formulated in the routing context (actually isomorphic to the present model under a certain transformation). However, the results they prove—when utility depends on both i and j—are existence and uniqueness results for Nash equilibria which crucially depend on the utility function being concave (convex in the case of their cost function). We will show in Sections 1.4 and 2.3 that in the networking context concavity need not hold.

1.3.2 Multi-Dimensional QoS Vectors and Burstiness

We investigate the problem of effectively facilitating quality of service in systems with multi-dimensional

[2]Recall that a (vector) function $f(x)$ is *quasi-concave* (*quasi-convex*) iff for all ϵ the set $\{x : f(x) \geq \epsilon\}$ ($\{x : f(x) \leq \epsilon\}$) is convex.

QoS vectors containing both mean- and burstiness-related QoS measures (e.g., packet loss rate, delay, and jitter).

Extension of Game-Theoretic Analysis We extend the game-theoretic analysis to multi-dimensional QoS vector games containing $s \geq 1$ different QoS measures. The monotonicity assumptions described in Section 1.2 are generalized to the s-dimensional QoS vector case. We show that the main results carry over if a uniformity assumption is placed either on application preference or on QoS vector functions.

Effect of Burstiness on QoS We study the impact of multiple QoS measures—sometimes with conflicting requirements imposed by heterogeneous user needs—on the characteristics of QoS rendered by the service classes. We show that under bursty traffic conditions, it is impossible for a service class to deliver superior QoS in both mean- and burstiness-related QoS measures (e.g., mean delay vs. jitter) vis-à-vis some other service class if weighted fair queueing is employed at routers. In particular, considering the four QoS measures packet loss rate, packet loss variance, mean delay, and delay variance, if service class j achieves a lower packet loss rate and mean delay than some other service class j', then j must exhibit either a higher packet loss variance or a higher delay variance vis-à-vis service class j'. In the case when only one jitter variable—say, packet loss variance—is considered, then a total ordering among service classes is possible, however, via the degenerate situation where the superior service class attains zero or near-zero packet loss.

The service class ordering results for multi-dimensional QoS vector systems, under bursty traffic conditions, show the existence of intrinsic limitations to achieving targeted differentiated service when using GPS scheduling. The particular ordering achieved depends on the degree of resource contention present in the system, and we demonstrate this in the context of self-similar traffic with varying degrees of scale-invariant burstiness.

1.4 Related Work

Microeconomic Approaches to Resource Allocation In recent years, there has been a surge of work in "microeconomic approaches to resource allocation" where ideas and tools from microeconomics and game theory have been applied in the formulation and solution of problems arising in flow control, routing, file allocation, load balancing, multi-commodity flow, and quality of service provision, among others [15, 44, 22, 21, 34, 24, 26, 16, 47, 48, 30, 7, 43, 39]. A collection of papers covering a broad range of topics can be found in [6]. A brief survey of some of the literature is provided in [14]. Some standard references to game theory and microeconomics include [1, 17, 42, 45, 46].

Many of the earlier papers including some recent ones [16, 15, 26, 31, 43] have espounded a cooperative game theory framework to model user interactions and derive results based on Pareto optimality. Although fruitful to investigate due to the powerful tools available in cooperative game theory, a potential drawback of this approach is the assumption that users or applications behave *cooperatively* in networking contexts. For the long-term establishment of virtual circuits or the leasing of telephone lines, the cooperative user model may indeed be viable[3]. However, for best-effort applications that comprise much of today's Internet traffic, users are largely anonymous with respect to thousands of other users who concurrently share network resources at any given time, and a noncooperative framework where each user is assumed to optimize individual performance based on his or her limited available information about the network state is better suited.

The *noncooperative* framework can be traced as far back as '81 to a paper by Yemini [51] who has since been more strongly associated with the cooperative approach. The noncooperative network resource allocation approach has been actively pursued by Lazar and his co-workers beginning in the late '80s [20, 2] with more recent work carried out jointly with Korilis and Orda [21, 22, 23, 24, 34]. Their main work has revolved around an optimal flow control problem, and the development of techniques needed to show the existence of Nash equilibria [22]. Korilis et al. [23, 24] have also looked at the problem of using interventions by an impartial external entity—the network manager—to steer a system toward Nash equilibria that are system optimal. Of special interest is Orda et al.'s work on routing games [34] which is intimately related to the multi-class QoS provision model studied in this paper. This is further explicated below.

Another significant thrust in noncooperative network games is due to recent work by Shenker [44] where it is shown how choosing a packet scheduling discipline can influence the nature of the Nash equilibria attained. In the context of a congestion control model, it is shown that for a large class of packet scheduling disciplines, a configuration being Nash need not imply that it is Pareto optimal. A packet scheduling discipline called Fair Share is described and it is shown to lead to Nash equilibria with desirable properties including uniqueness

[3] It is also possible that intermediaries perform long-term leasing of network resources which are then packaged and made available as high-level services to the user. Aspects of such activities may be modeled as coalition behavior.

and reachability by a class of self-optimization procedures.

On the implementation side, the work of Waldspurger et al. [47] deserves attention since it is one of the few works that have built a nontrivial working system—CPU allocation and load balancing in workstation networks—and demonstrated that a system based on microeconomic principles can indeed work in practice. Other implementations worth noting include Wellman's work on multicommodity flow problems [48, 49].

QoS-Related Network Games Several papers have addressed the specific issue of multi-class QoS provision in high-speed networks using microeconomic models [7, 31, 43, 19]. In [31, 43], utility functions are defined with link bandwidth and switch buffers acting as substitutable resources. Pareto-optimal allocation of resources among service classes is affected either by the network exercising admission control [43] or by users performing purchasing decisions [31]. In both approaches, it is assumed that QoS guarantees are computable, given specific resource reservations. As stated earlier, an important goal of our approach is to shield the user from having to make complex computations to estimate service quality.

In [7], a general framework for investigating pricing in networks is proposed, with service discipline and pricing policy acting as design variables. Simulation results are shown that depict the existence of "desirable" price ranges related to system optimality. The simulations were carried out using a 2-service class packet scheduling algorithm where a shared FIFO queue was partitioned into two segments with high priority packets being queued at the front and low priority packets being queued at the back. Four types of applications with different QoS requirements were tested with priority settings set either to 1 or 2.

Our model is an n-application, m-service class, s-dimensional QoS vector quality service provision model, and emphasizes a different set of questions from that of [7] where the effect of pricing incentives are paramount. We apply noncooperative game-theoretic analysis to the *multi-dimensional* QoS vector model to understand under what conditions Nash equilibria exist and how they are related to Pareto and system optimality. We also investigate the problem of facilitating service classes with induced QoS levels that match application requirements under bursty traffic conditions.

Comparison with Congestion Control Models by Korilis et al. and Shenker The flow or congestion control models of Korilis et al. [22] and Shenker [44] represent a form of quality of service provision and it is important to explicate the differences between our model and theirs, given that all three follow the noncooperative framework. The main difference between the models by Korilis et al. and Shenker, and the model studied in the paper is that, indeed, theirs *is a flow/congestion control model*. Phrased in the language of the QoS provision model defined in Section 1.2 (a formal definition is given in Section 2.3), both [22] and [44] correspond to the situation where $n = m$, each player i is permanently assigned to the *fixed* service class i, and either $\lambda_{ii} \geq 0$ [44] or $0 \leq \lambda_{ii} \leq \lambda_i$ [22], but in both cases, $\lambda_{ij} = 0$ for $i \neq j$. That is, a player, being tied to a fixed service class, has the option of controlling how much traffic [44]—or using what time schedule [22]—to send his traffic but *not where*. Since delay or any other performance measure will deteriorate with increased traffic volume, but volume itself, keeping other things fixed, will generally increase utility, there is an optimum volume assignment—i.e., optimal flow or congestion control—that maximizes player i's utility.

In our model, there is no a priori fixed 1-1 correspondence of players to service classes. Indeed, the very *essence* of the QoS provision problem is to give each player $i \in [1, n]$ the freedom to choose *where* she wants to send her traffic, from service class 1 all the way to service class m. Hence, our QoS provision model is fundamentally different from the flow control models, being more complex and producing equilibria structures that are different from [22], [44]. Secondly, our model incorporates multi-dimensional QoS vectors whose consequences are then analyzed in both game-theoretic and network performance terms.

Comparison with Orda et al.'s Routing Model In [34], Orda et al. present a noncooperative routing game where a set of users with fixed throughput demands have a choice of assigning their flow to a set of *parallel links* or *routes*. Although motivated by different contexts, assuming *independence* between the parallel links—i.e., the performance characteristics (e.g., queueing delay) on some link or route depends only on the aggregate traffic volume assigned to it—a 1-1 correspondence can be established between Orda et al.'s routing model and the QoS provision model studied here.

Phrased in our language, the set of parallel links correspond to the service classes $j \in [1, m]$, and a user i's average throughput demand λ_i is assigned to the m routes given by the assignment vector $\Lambda_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{im})$. Orda et al. then define a cost function J_j^i which corresponds to our utility function $U_i(\lambda_{ij}, c_j)$. Both depend on the player i as well as the service class (or route) j. Since J_j^i is interpreted as a cost function, their's is a minimization problem. The similarities, however, end here.

Orda et al. study the routing game under three successively more restrictive assumptions on the cost function J_j^i (called type-A, type-B, and type-C). In type-B and type-C, the cost function J_j^i takes on the form $\lambda_{ij} c_j(q_j)$ thus losing its dependence on i except for the weighting term λ_{ij}. As is formally defined in Section 2.3, in our QoS provision game, the utility function has the form $\lambda_{ij} U_i(c_j(q_j))$; thus the utility's dependence on heterogenous user preferences is preserved. Hence the results proved in [34] for type-B and type-C functions correspond to a population of users with homogenous preferences, and thus do not carry over to the more general and complex QoS provision game studied here.

As for type-A games where dependence on individual preferences is preserved, the assumption is made that J_j^i is convex (concave in our context) in λ_{ij}. However, as has been explicated in Section 1.2, the two monotonicity assumptions—c_j is increasing in q_j and U_i is decreasing in c_j—which are basic postulates applicable to most networking contexts of interest, are incompatible with the assumption that J_j^i is convex is λ_{ij}. In fact, a simple consequence of the monotonicity assumption is that J_j^i is *quasi-convex* in λ_{ij}. This is so since the composition of the two monotone functions again relates U_i monotonically (decreasing) to λ_{ij}, and monotone functions are trivially quasi-convex. Convexity and quasi-convexity, in the QoS provision context, however, lead to consequences worlds apart.

More specifically, the assumption that J_j^i is convex in λ_{ij} is needed in [34] to invoke Rosen's theorem [41], a common tool for exhibiting the existence of Nash equilibria. Rosen's theorem, in turn, uses Kakutani's fixed point theorem to establish existence. To apply Kakutani's fixed point theorem, a certain set arising from a point-set map must be convex, and this can be shown to hold if J_j^i is convex in λ_{ij}. If J_j^i is quasi-convex, however, all bets are off (this is formally discussed in Section 2.3). The non-applicability of Rosen's theorem, of course, does not imply that Nash equilibria do not exist; after all, existence may be shown by some other means. We settle the question by proving directly that for a large family of noncooperative multi-class QoS provision games, no Nash equilibria exist. From a technical perspective, our game-theoretic contributions complement the work by Orda et al. and constitute the first results that give a comprehensive analysis of the structure of the noncooperative multi-class QoS provision game where users possess heterogenous QoS preferences (leading to non-concave utilities) and they are allowed to choose both the service classes and traffic volumes assigned to them.

Many-Switch Systems In [4, 5], we describe an architecture for noncooperative multi-class QoS provision in *many-switch* systems[4]. Motivated by the analytical (and simulation) results and insights of this paper, we use the single-switch model as a building block in constructing a scalable architecture for multi-class QoS provision in WAN environments. We solve the end-to-end QoS provision problem in many-switch systems and the inter-switch couplings they introduce using a distributed control that shields the user from complex computations while preserving the basic premise of selfishness. We show that the network system is able to provide predictable, stratified service without resource reservations and is adaptive under stationary and non-stationary changes in network state.

The rest of the paper is organized as follows. In Section 2, we describe the overall set-up and formulate the network QoS provision model. Section 2.3 discusses the differences between our model and the model of Orda et al. [34], and the impact of heterogenous preferences in bringing about non-concave utilities. This is followed by Section 3 which gives a game-theoretic analysis of the QoS provision game structure. Section 4.1 extends the game-theoretic analysis to multi-dimensional QoS vectors and Section 4.2 studies the effect of burstiness on the characteristics of rendered QoS. We conclude with a discussion of our results and future work.

2 Noncooperative Network QoS Provision Game

2.1 Network Model

The network 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 packet scheduling 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 $\alpha_j \mu^{NR}$.

In keeping with the ATM framework, we assume fixed-size packets (i.e., *cells*) and we employ output-buffered switches. We implement a generic form of weighted fair queueing achieving perfect isolatedness and conservation of work. The latter come into effect

[4] Also called network of switches in [44].

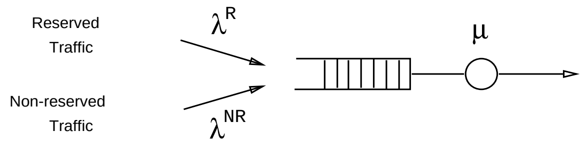


Figure 2.1: Dual traffic classification at output-buffered switch with shared priority queue implementing weighted fair queueing.

when performing simulations. We ignore efficient implementation considerations of WFQ, treating the processing cost at switches as fixed. The assumption of fixed-size packets simplifies the faithful rendering of service rates commensurate with the weights $\alpha_1, \ldots, \alpha_m$.

2.2 Application Model

Utility Function 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 $U : \mathbb{R}^s \rightarrow \mathbb{R}_+$, $s \geq 1$, 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.

The shapes of the utility functions indicate that nonurgent 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.

Selfishness Selfishness, in our context, will mean that each application $i \in [1, n]$ will try to take actions so as to maximize its individual utility U_i. The forms for U_i as well as user i's decision variables for the multi-class QoS provision problem are defined in the next section.

2.3 Definition of Network QoS Provision Game

QoS Provision Problem 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 (cf. Section 1.2), 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. This relation is only approximate for work conserving switches. The precise modeling of nonlinearities arising from work conservation, although interesting in its own right, is a general issue not specific to our context, and we will ignore its effect in this paper.

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[5]. We will also assume that $dU_i/dc \leq 0$. That is, making the packet loss rate smaller[6] 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. It turns out that the analysis of the multi-dimensional case reduces to the scalar case under certain conditions, and we will proceed with packet loss rate c as the sole QoS indicator.

The *weighted utility* of application i, given assignment Λ, is defined as

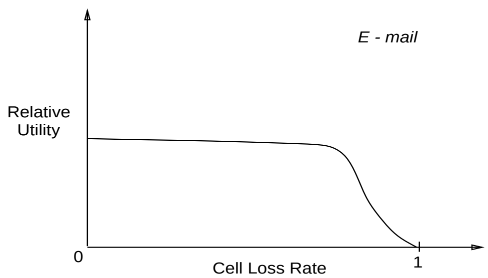
$$\bar{U}_i(\Lambda) = \sum_{j=1}^{m} \lambda_{ij} U_i(c_j).$$

Note that the utility function used in Section 1.2, $U_i(\lambda_{ij}, c_j)$, corresponds to $\lambda_{ij} U_i(c_j)$. Subject to the above constraints, the static optimization problem can be formulated as

$$\max_{\Lambda} \; \bar{U}(\Lambda) \; = \; \sum_{i=1}^{n} \bar{U}_i(\Lambda). \qquad (2.1)$$

[5] We sometimes use continuous notation for expositional purposes. Our results do *not* depend on c_j and U_i being smooth.

[6] An analogous assumption is made in the multi-dimensional QoS vector case (Section 4.1).



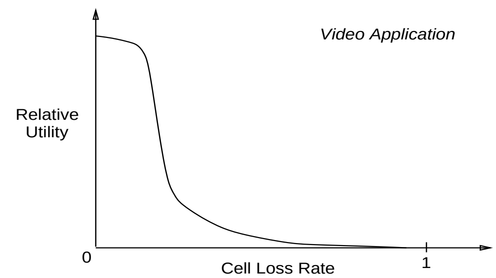


Figure 2.2: Utility functions. E-mail application (left) and video application (right).

This is a nonlinear programming problem with equality constraints.

Nash Equilibria, Pareto Optima, and System Optima Any Λ^* that satisfies (2.1) 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,

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

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

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

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 $\forall i \in [1, n]$, $\forall \Lambda \in \mathcal{L}_i(\Lambda^*)$,

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

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[7].

Simplifying Assumption To make the analysis tractable, we will work with (unit) *step utility functions* where for each player $i \in [1, n]$,

$$U_i(c) = \begin{cases} 1, & \text{if } c \le \theta_i, \\ 0, & \text{otherwise.} \end{cases}$$

Here $\theta_i \ge 0$ is a threshold that represents the i'th application's preference. Since $c_j = c_j(q_j)$, $j \in [1, m]$, there exist $b_{ij} \ge 0$ such that

$$U_i(c_j(q_j)) = \begin{cases} 1, & \text{if } q_j \le b_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

With a slight abuse of notation, we will sometimes write $U_i(q_j)$ for the composite function when the distinction is clear from the context.

Non-Concave Utilities The simplification is reasonable from two perspectives. First, from the technical side, we do not lose very much by sacrificing continuity of the utility function since Lemma 3.5—which shows the existence of 2-application/2-service class games with no Nash equilibria—can be shown to hold even when U_i is continuous and quasi-concave (but not concave) in each λ_{ij}. This also holds for Theorem 3.6 which generalizes Lemma 3.5 to n-application/m-service class games. The crucial factor in proving non-existence is the quasi-concavity property which allows U_i to be concave and convex over local segments and thus produce "holes" when forming convex combinations. In particular, even though U_i is quasi-concave in each λ_{ij}, $\bar{U}_i = \sum_j \lambda_{ij} U_i$ need not be quasi-concave.

[7] We note that in the ordinal (vs. cardinal) approach to modeling with utility functions, one refrains assigning values to utilities for the simple reason that doing so may be meaningless. Pareto optimality, then, becomes the central point of interest when considering optimality properties of Nash equilibria.

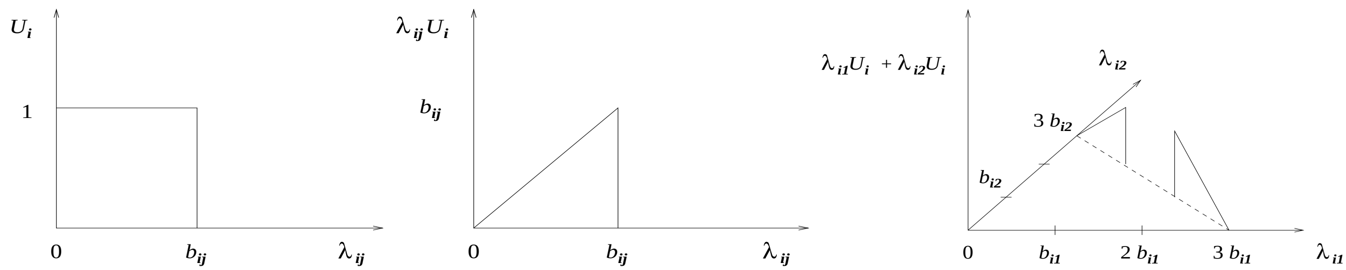


Figure 2.3: Left: U_i as a function of λ_{ij} (via q_j). Middle: $\lambda_{ij} U_i$ as a function of λ_{ij}. Right: $\bar{U}_i = \lambda_{i1} U_i + \lambda_{i2} U_i$ as a function of λ_{i1}, λ_{i2}.

Since the understanding of this point is important—this is where the heterogeneity of application QoS preference exerts its influence—let us illustrate it with a simple example. Assume we are given a 2-application/2-service class system where player i's utility function satisfies $U_i(q_j) = 1$ if $q_j \le b_{ij}$ ($j = 1, 2$), and 0, otherwise. Furthermore, assume $b_{i1} = b_{i2}$ (i.e., there is no need to make U_i depend on the service class j), and $\lambda_i = \lambda_{i1} + \lambda_{i2} = 3b_{ij}$. Figure 2.3 (left) and (middle) show the functions U_i, $\lambda_{ij} U_i$: clearly, both functions are quasi-concave in λ_{ij}. The total individual utility $\bar{U}_i = \lambda_{i1} U_i + \lambda_{i2} U_i$, however, while still being quasi-concave in each variable λ_{ij}, is itself *not* quasi-concave. This is depicted in Figure 2.3 (right) which shows $\bar{U}_i$ over the feasible region $\mathcal{L} = \{(\lambda_{i1}, \lambda_{i2}) : \lambda_{i1} + \lambda_{i2} = 3b_{ij}\}$. Even though $\mathcal{L}$ is convex, the set $\{(\lambda_{i1}, \lambda_{i2}) : \bar{U}_i \ge b_{i1}/2\}$ is *not* convex: the projection onto $\mathcal{L}$ yields two separated line segments. This disables the application of Rosen's theorem [41], which in our case—due to Lemma 3.5—is synonymous with non-existence of Nash equilibria. Relating back to Orda et al.'s routing game [34], this is also the rigorous underpinning of the discussion given in Section 1.4 which showed why assuming concavity (i.e., convexity for minimization in [34]) in the networking context can be problematic.

It is not difficult to see that discontinuity, caused by the step function assumption of U_i, did not play an essential role in the previous argument. Even if we "round off" U_i near θ_i and make it smooth (say in C^1), unless we concoct it to be concave—a necessary condition (unless U_i is constant) is to extend the support of U_i over at least $[0, \lambda_i]$ and make it "dome-shaped"—concavity will not be achieved. Indeed, constructing a concave utility function which captures the QoS requirement of a generic real-time application that states that only a small packet loss rate can be tolerated seems intrinsically difficult. Finally, even if for some application its utility U_i were concave, after composing it with the packet loss function c_j (or some other performance function in the multi-dimensional QoS vector case), U_i need not be concave anymore. Monotonicity (of c_j) does not preserve concavity under function composition.

Second, threshold or step utility functions have been implicitly applied in practical and analytical settings to model and encode/convey QoS preferences. For example, *hard* real-time systems, as defined in the real-time systems literature, have this "all or nothing" property. Furthermore, irrespective of whether the user of an application possesses a step utility preference or not, when interacting with a network system through an application, the user must ultimately code and convey her preference to the underlying system. *Bounds* on packet loss rate, delay, jitter, and other QoS measures have been used to encode application traffic QoS requirements in different contexts including ones where they are used to compute resource reservations and and in some commercial applications [32].

3 Properties of Noncooperative QoS Provision Game

3.1 Nash Equilibria and Existence Conditions

This section investigates the structure of Nash equilibria giving a complete characterization of Nash equilibria in the noncooperative multi-class QoS provision game as well as their existence conditions. First, let us impose a total order on the n players given by

$$i \le i' \iff \theta_i \le \theta_{i'}.$$

Unless otherwise stated, we will assume such a fixed order in the rest of the paper. Following is a simple but often used fact on the induced ordering of the traffic volume thresholds b_{ij}. It is a consequence of the total ordering of θ_i and the monotonicity of c_j.

Proposition 3.1 *$\forall i \in [1, n-1]$, $\forall j \in [1, m]$, $b_{ij} \le b_{i+1j}$.*

Next, we define certain subsets of service classes—parameterized by user i—that come into play when characterizing Nash equilibria. Let $I_i^+ = \{ j \in [1,m] : q_j > b_{ij},\ \lambda_{ij} > 0 \}$, $I_i^- = \{ j \in [1,m] : q_j < b_{ij} \}$, and $I_i^0 = \{ j \in [1,m] : q_j = b_{ij} \}$. That is, I_i^+ denotes the set of service class indices where player i has assigned a positive flow and the total traffic volume allocated exceeds player i's threshold. Thus user i attains 0 utility in these service classes. Conversely for I_i^- and I_i^0, however, it is not required that user i have a nonzero assignment in these classes. Let $q_j^i = \sum_{k \neq i} \lambda_{kj}$. That is, q_j^i is the traffic volume assigned to service class j not counting player i's contribution (if any). Hence $q_j = \lambda_{ij} + q_j^i$.

Let $J_i^+ = \{ j \in [1,m] : q_j^i \geq b_{ij} \}$ and $J_i^- = \{ j \in [1,m] : q_j^i < b_{ij} \}$. Hence J_i^+ is the set of service classes where, irrespective of player i's actions, player i cannot garner any utility. Let $J_i^* = \{ j \in [1,m] : b_{ij} - q_j^i = \min_{k \in J_i^-} b_{ik} - q_k^i \}$. J_i^* is the subset of service classes of J_i^- where the positive utility achievable by user i is minimal.

The next two results give uniform upper bounds on the individual utility of a fixed player where uniformity is with respect to all unilateral strategy changes by the player. Recall that the latter is denoted by $\mathcal{L}_i(\Lambda)$ where Λ is any configuration.

Proposition 3.2 *Given Λ, $i \in [1,n]$, let $v_i = \sum_{j \in J_i^-} b_{ij} - q_j^i$. Then*

$$\forall \Lambda' \in \mathcal{L}_i(\Lambda), \quad \bar{U}_i(\Lambda') \leq v_i.$$

Proposition 3.3 *Given Λ, $i \in [1,n]$, let $\lambda_i > v_i$ and $J_i^+ = \emptyset$. Then $\exists j^* \in J_i^*$ such that*

$$\forall \Lambda' \in \mathcal{L}_i(\Lambda), \quad \bar{U}_i(\Lambda') \leq v_i - (b_{ij^*} - q_{j^*}^i).$$

The two propositions are used in the proof of the following theorem which gives a complete characterization of Nash equilibria.

Theorem 3.4 (Nash Characterization) *Λ is a Nash equilibrium iff $\forall i \in [1,n]$ either*

(a) *$I_i^+ = \emptyset$, or*

(b) *$I_i^- = \emptyset$, $J_i^+ \neq \emptyset$, $J_i^- \subseteq I_i^0$, or*

(c) *$I_i^- = \emptyset$, $J_i^+ = \emptyset$, $\exists\, j^* \in J_i^*$ such that $J_i^- \backslash \{j^*\} \subseteq I_i^0$.*

In words, for each player i, one of three conditions must hold: a user either achieves full individual utility λ_i (part (a)), or partial utility $v_i = \sum_{j \in J_i^-} b_{ij} - q_j^i$ "dumping" the excess traffic $\lambda_i - v_i$ into one or more service classes belonging to J_i^+ (part (b)), or partial utility $v_i - (b_{ij^*} - q_{j^*}^i)$ with excess traffic being assigned to one of the service classes in J_i^* (part (c)). Service classes belonging to J_i^+ form the most natural dumping ground for channeling excess traffic since player i cannot derive utility from $j \in J_i^+$ no matter what. If $J_i^+ = \emptyset$, J_i^* takes on a surrogate role.

The next lemma gives a simple sufficiency condition for 2-application/2-service class games in which Nash equilibria do not exist.

Lemma 3.5 *Consider the family of 2-application/2-service class systems such that the thresholds b_{ij} on the total traffic volume of the service classes satisfy $b_{1j} < b_{2j}$, $j = 1, 2$ (i.e., the ordering of Proposition 3.1 is strict). Furthermore, assume the following inequalities hold:*

(a) $\lambda_2 < b_{11} + b_{12}$,

(b) $\lambda_2 + \lambda_1 > b_{21} + b_{22} > b_{11} + b_{12}$,

(c) $\lambda_2 > \max\{b_{11}, b_{12}\}$.

Then, for such choices of λ_i, b_{ij}, no Nash equilibrium exists.

Games satisfying the above conditions are easy to construct, and the reason that there are no Nash equilibria is because the game leads to a limit cycle. This type of behavior has also been observed in simulation studies. Next we generalize the "Nash Non-Existence condition" to n-application/m-service class games. The proof of Theorem 3.6 can be reduced to Lemma 3.5 and is a straightforward consequence.

Theorem 3.6 (Nash Non-Existence) *Consider a n-application/m-service class game where the ordering implied by Proposition 3.1 is strict. If there are players i' and i^* with $i^* > i'$ satisfying*

(a) $\sum_{i \neq i'} \lambda_i < \sum_j b_{i'j}$,

(b) $\sum_i \lambda_i > \sum_j b_{i^*j}$,

(c) $\sum_{i \leq i'} \lambda_i + \sum_{i > i^*} \lambda_i < \min_j b_{i^*j}$,

then no Nash equilibrium exists.

Whereas Lemma 3.5 and Theorem 3.6 constituted simple, easily constructable conditions for Nash non-existence, the next theorem gives a complete characterization of n-application/m-service class games for which Nash equilibria do exist.

Theorem 3.7 (Nash Existence) *Consider a n-application/m-service class game where the ordering implied by Proposition 3.1 is strict. Then a Nash equilibrium exists if and only if at least one of the following holds:*

(a) *Each player is "domitable;" i.e.,* $\forall i, \sum_{i' \neq i} \lambda_{i'} \geq \sum_j b_{ij}$.

(b) *Let* $i^* = \min\{i : J_i^- \neq \emptyset\}$. *There is a configuration* Λ *such that* $\forall i > i^*$, $I_i^+ = \emptyset$, *and one of the three conditions of Theorem* 3.4 *holds for player* i^*.

The above characterization has several interesting features. First, the theorem states that if any Nash equilibrium exists at all, then, in fact, a Nash equilibrium exists (possibly different) satisfying conditions that are much more restrictive than those of Theorem 3.4. Second, removing the *existential quantifier* in part (b) of the theorem is not possible[8] without replacing it by another existential quantifier of similar scope. This is due to the fact that the problem of checking if a Nash equilibrium exists—given the parameters of a game—is *NP*-complete[9]. The proof of hardness relies on the hardness of checking whether there is a configuration satisfying constraint (b) in the theorem. The latter, in turn, is proved using a reduction from minimum cost multicommodity network flow with step cost functions.

The relevance of these remarks is that, even though it is possible to completely characterize QoS provision games for which a Nash equilibrium exists, it is not possible to give an *effective* characterization in the sense of feasible computability. Thus control algorithms, even if privy to information about the network state, cannot, in general, accurately determine whether a network system with given resources and user demands is prone to instability in the Nash sense.

Let us consider a retricted QoS provision game where each user must channel his entire traffic into a single service class. That is, traffic is *unsplittable*. When viewed in the routing context, this would correspond to a circuit-switched system where a connection, once assigned a route, must follow the path during the entire lifetime of the connection. In our model, this corresponds to placing the *further* restriction that $\lambda_{ij} \in \{0, \lambda_i\}$ for all users i and service classes j. Interestingly, for this restricted game, we can show that a Nash equilibrium always exists.

Theorem 3.8 (Unsplittable Games) *Any unsplittable game has a Nash equilibrium.*

Relating back to the issue of concavity and Nash existence, for unsplittable games, the problem of having to consider function values over convex combinations when utility is quasi-concave does not arise since the domain is discrete. Existence, however, does not mean that a Nash equilibrium is always reached starting from any initial configuration. In Section 3.3, Theorem 3.15, we show that for certain "resource-plentiful" systems, there is robust convergence to Nash equilibria from any initial state.

[8] More precisely, "highly unlikely" since our argument depends on the $P \neq NP$ conjecture.

[9] This result, and the machinery to prove it, have been omitted due to space constraints.

3.2 Relationship to Pareto and System Optimality

In this section, we characterize the relationship between Nash equilibria, Pareto optimal, and system optima for the multi-class QoS provision game. First, we state a useful lemma that can be used to relate Pareto optimality of a configuration to system optimality.

For a configuration Λ, an equivalent assignment Λ' can be found with the same total utility so that the players are partitioned into two sets around a unique, dividing player $i_{\Lambda'}$. The first set consists of players with indices larger than $i_{\Lambda'}$ with respect to the ordering induced by Proposition 3.1, with all players having full utility. The second set consists of players with smaller indices than $i_{\Lambda'}$, all of them having zero utility. The third set is the singleton set $\{i_{\Lambda'}\}$ consisting of the dividing player who has partial utility. We will call such an assignment Λ' a *normal form* of Λ.

Lemma 3.9 (Normal Form) *Let* Λ *be a configuration with* $\bar{U}(\Lambda) < \sum_{i=1}^n \lambda_i$. *Let* $i_\Lambda \equiv \max\{i : \bar{U}_i(\Lambda) < \lambda_i\}$. *Then* $\exists\, \Lambda'$ *with* $\bar{U}(\Lambda') = \bar{U}(\Lambda)$ *such that*

(a) $\forall i < i_{\Lambda'}$, $\bar{U}_i(\Lambda') = 0$, *and*

(b) $\forall i > i_{\Lambda'}$, $\bar{U}_i(\Lambda') = \lambda_i$.

The usefulness of the normal form of a configuration (including Nash) comes into play when checking for system optimality of a Nash assignment. This is so since, as we shall see, it is sufficient to check Pareto optimality of the normal form to establish system optimality of the original configuration. Moreover, a normal form is easy to obtain from the original Nash configuration (construction in the proof of Lemma 3.9) and checking for Pareto optimality is generally easier than checking for system optimality.

Theorem 3.10 (Pareto & System Optimal) *Given a configuration* Λ, *let* Λ' *be its normal form. Then* Λ *is system optimal iff* Λ' *is Pareto optimal.*

An immediate corollary of the theorem is that a Nash equilibrium is system optimal iff its normal form is Pareto optimal. Although Theorem 3.10 gives an interesting relationship between Pareto optimality and system optimality and is useful for reasoning about Nash equilibria in other contexts, it falls short of further exploiting potential structure specific to Nash equilibria. It is an open question whether there is some "independence" relation between Nash equilibria and system optima for the general multi-class QoS provision game.

Given the form of Theorem 3.10, one may wonder whether all assignments that are Nash and Pareto optimal are also system optimal. The next result gives a counterexample which shows that Theorem 3.10 is "tight" in the sense that, when conditioned with Nash equilibria, there are assignments that are both Nash and Pareto but not system optimal.

Proposition 3.11 *There exist Nash equilibria that are Pareto optimal but not system optimal.*

Next, we characterize those Nash equilibria that *are* Pareto optimal. First, consider a *modified game*, parameterized by some assignment Λ, defined as follows. The thresholds for the players remain the same as in the original game. However, for each player i, the mean arrival rates are taken to be $\gamma_i \equiv \bar{U}_i(\Lambda)$. Moreover, there is an additional player 0 whose thresholds b_{0j} are all 0, but whose traffic demand is $\gamma_0 = \sum_i \lambda_i - \sum_i \gamma_i$. Note that the configurations Λ' in the original game for which $\forall i : \bar{U}_i(\Lambda') \geq \bar{U}_i(\Lambda)$ correspond (many-to-one) to system optimal configurations M for the modified game. Let $i_j := \min_{i \neq 0}\{\gamma_{ij} > 0\}$.

Theorem 3.12 (Nash-Pareto Characterization) *Let Λ be a Nash equilibrium and let i^* be the player such that $\forall i > i^*$, $\bar{U}_i(\Lambda) = \lambda_i$; i.e., i^* is the largest player with incomplete utility. Then Λ is a Pareto optimum if and only if the following hold:*

(a) $\forall i \leq i^*$, $I_i^+ \subseteq \{j : q_j > b_{i^*j}\}$.

(b) $\forall j\ [q_j \leq b_{i^*j} \Rightarrow \forall i\ j \notin I_i^+]$. *Notice since Λ is Nash, it follows from the hypothesis above and Theorem 3.4 that $q_j = b_{i^*j}$.*

(c) *The two sets of players $S_1 \equiv \{i > i^* : \exists j\ \lambda_{ij} > 0,\ \exists i' \leq i^*\ j \in I_{i'}^+\}$ and $S_2 \equiv \{i > i^* : \exists j\ \lambda_{ij} > 0, q_j \leq b_{i^*j}\}$ are disjoint.*

(d) *For any system optimum configuration M of the modified game, i.e., $\bar{U}(M) \geq \sum_{i=1}^n \gamma_i$, one of the following holds for each service class j:*

  (d1) $\sum_{i=0}^{n} \gamma_{ij} = b_{i_j j}$ *when i_j is defined,*

  (d2) $\sum_{i \neq 0} \gamma_{ij} \geq b_{i^*j}$,

  (d3) $\gamma_0 > b_{i^*j} - \sum_{i \neq 0} \gamma_{ij} + \sum_{j' \neq j} b_{i_{j'}j} - \sum_i \sum_{j' \neq j} \gamma_{ij}$.

Note that in part (c) of Theorem 3.12, an even stronger statement is true: Consider the directed graph G whose vertices are the players $i > i^*$ and whose edges are defined as follows. An edge (i_1, i_2) exists in G if and only if $\{j : \lambda_{i_1 j} > 0,\ \lambda_{i_2 j} > 0\} \neq \emptyset$ or $\exists\, j_1, j_2$ with $\lambda_{i_1 j_1} > 0$, $\lambda_{i_2 j_2} > 0$, and $q_{j_2} \leq b_{i_1 j_2}$. Then there is no path from any vertex in S_2 to any vertex in S_1 in the graph G. In other words, for all players $i > i^*$, there is a path from S_2 to i, or from i to S_1, or neither, but not both.

There are several interesting points to note in the above characterization. First, parts (a) and (b) depend on the combination of facts that Λ is both Nash and Pareto. Parts (c) and (d), however, depend only on the fact that Λ is Pareto. Second, removing the universal quantifier in (d) ("For *any* configuration M ...") is impossible for reasons similar to removing the existential quantifier in the statement of Theorem 3.7. The problem of deciding whether a configuration is *not* Pareto is NP-complete as long as the thresholds of each player are allowed to vary arbitrarily across the classes. Third, the optimization problems that correspond to the above decision problems possess convex feasible regions but the objective functions are highly nonlinear and even discontinuous.

On the other hand, the feasible region can be naturally partitioned into convex subregions over each of which the objective function is, in fact, linear. In each such region, the traffic volume q_j of each class lies between an adjacent pair of threshold values $b_{i_j j}$ and $b_{i_j+1 j}$. The properties (a) to (c) in the above theorem, and, in fact, most of the structural results in this paper, rely on the behavior of objective functions whose level sets are convex *within* the subregions where they are linear. However, as encountered earlier in the context of inapplicability of Kakutani's fixed point theorem, the level sets of these objective functions are nonconvex and consist of an intractably large number of disconnected components once we move outside the boundaries of these subregions. Therefore, searches for optima across boundaries of these subregions rapidly result in combinatorial explosion. The monotonicity properties of Proposition 3.1 do not seem to control this explosion.

In general, a simple consequence of the above discussion is that many Nash equilibria exist which are not Pareto optimal. In fact, the normal form of a Nash assignment Λ obtained from the construction in the proof of Lemma 3.9 is typically itself Nash, and can be used to exhibit assignments that are Nash but not Pareto optimal. Thus, in general, gaps exist in all the important relations between configurations that are Nash equilibria, Pareto optimal, or system optimal.

3.3 Resource-Plentiful Systems and Dynamical Behavior

In this section, we show that for certain "resource-plentiful" systems Nash equilibria always exist, and furthermore, they are always Pareto and system optimal. We also show that starting from any initial configuration robust convergence to a Nash equilibrium is achieved.

We define a dynamic game via the *dynamic update process* $\mathcal{P}$ as follows. We assume that the players move asynchronously, and at each step t, a single player i_t unilaterally and selfishly reassigns its λ_{i_t} so that the new assignment Λ_t maximizes its individual utility $\bar{U}_{i_t}(\Lambda)$. We further assume that no player moves *unnecessarily*—i.e., a player only makes changes to its assignment if it thereby strictly increases its individual utility. Moreover, for each user i there is an infinite sequence of time steps $t_1^i < t_2^i < \cdots$ where i is allowed to perform an update (including a "no move" update).

Theorem 3.13 (Resource-Plentiful System) *For all $i \in [1, n]$, let*

$$\sum_{j=1}^{m} q_j \leq \sum_{j=1}^{m} b_{ij}. \tag{3.14}$$

Then Λ is a Nash equilibrium if and only if Λ is a system optimum if and only if Λ is a Pareto optimum. Moreover the optimum value achieved is $\bar{U}(\Lambda) = \sum_j q_j = \sum_i \lambda_i$.

First, note that $\lambda = \sum_j q_j$. Resource plentifulness manifests itself via $\sum_{j=1}^{m} b_{ij}$. Since $b_{ij} = c_j^{-1}(\theta_i)$ where c_j is the packet loss function and θ_i is user i's utility threshold (cf. Proposition 3.1), the more resources there are available in the system (e.g., bandwidth), the less pronounced c_j will be and the larger b_{ij} (keeping θ_i fixed). Condition (3.14) then states that there are sufficient resources available to *potentially* accommodate each user's requirements, and Theorem 3.13 shows that this is indeed the case even when users are selfish. The next theorem shows that such desirable configurations can be realized in a noncooperative manner starting from any initial configuration.

Theorem 3.15 (Convergence) *Assume the supposition of Theorem 3.13 holds. Then, starting from any initial configuration Λ_0, the dynamic update process $\mathcal{P}$ converges to a Nash equilibrium Λ. Moreover, Λ is attained as soon as the sequence of players (i.e., moves) in the process $\mathcal{P}$ includes the subsequence $n, n-1, \ldots, 1$.*

4 Multi-Dimensional QoS Vectors

This section generalizes the QoS provision model to non-scalar QoS vectors. We seek to answer two questions which arise as a result of the extension. First, do the game theoretic results of Section 3 carry over in the multi-dimensional QoS vector case? Second, what is the effect of system variability—caused by fluctuating background and source traffic—on the rendered QoS of the service classes when multiple QoS measures are present?

4.1 Extension of Game-Theoretic Analysis

In Section 2, we formulated a noncooperative QoS provision game based on singleton QoS vectors, $\mathbf{x} = (c)$, where c was a bound on packet loss rate. Here, we will extend the model to multi-dimensional QoS vectors $\mathbf{x} \in \mathbb{R}^s$, $s \geq 1$, and show that the singleton vector analysis carries over unchanged.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_s)^T$, and let $\mathbf{x}^j = (x_1^j, x_2^j, \ldots, x_s^j)^T$ denote the quality of service rendered to service class $j \in [1, m]$. As before, we make the monotonicity assumption $dx_r^j / dq_j \geq 0$, $r \in [1, s]$, $j \in [1, m]$, which is satisfied by most packet scheduling policies of interest including weighted fair queueing. Each player's utility function $U_i(\mathbf{x})$, $i \in [1, n]$, has the form

$$U_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \forall r \in [1, s],\ x_r \leq \theta_r^i, \\ 0, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\theta}^i = (\theta_1^i, \theta_2^i, \ldots, \theta_s^i)^T \geq \mathbf{0}$ is the multi-dimensional threshold vector that represents the i'th application's preference.

In order to deal with the multi-dimensional QoS vectors and thresholds uniformly, we henceforth make one of two uniformity assumptions: either assume that the thresholds θ_r^i can be ordered such that the ordering is uniform over r, i.e.,

$$\forall r \in [1, s], \forall i \in [1, n]: \ \theta_r^i \leq \theta_r^{i+1}, \tag{4.1}$$

or we assume that the functional forms x_r^j are uniform over r for each j, i.e.,

$$\forall j \in [1, m]: \ x_1^j = x_2^j = \cdots = x_s^j. \tag{4.2}$$

By isolatedness, $x_r^j = x_r^j(q_j)$, $r \in [1, s]$, $j \in [1, m]$, and just as in Proposition 3.1, the condition $x_r^j(q_j) \leq \theta_r^i$ can now be stated as $q_j \leq b_{ij}^r$ using the definition

$$b_{ij}^r = (x_r^j)^{-1}(\theta_r^i).$$

Let b_{ij} be the minimum over r, i.e., $b_{ij} = \min_{r \in [1,s]} b_{ij}^r$.

We can now rephrase $U_i(\mathbf{x}^j)$ as

$$U_i(\mathbf{x}^j) = \begin{cases} 1, & q_j \leq b_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, under the assumption that the functional forms x_r^j are uniform over r for each j where x_*^j satisfies $\forall j \in [1,m]$, $\forall r \in [1,s]$, $x_r^j = x_*^j$, and using the monotonicity of x_*^j, it can be observed that the following identity holds:

$$b_{ij} = \min_{r\in[1,s]} (x_*^j)^{-1}(\theta_r^i) = (x_*^j)^{-1}(\min_{r\in[1,s]} \theta_r^i). \qquad (4.3)$$

That is, the min operator commutes with $(x_*^j)^{-1}$.

Now we are ready to state a total ordering on b_{ij} for fixed j corresponding to its counterpart Proposition 3.1.

Proposition 4.4 *For the multi-dimensional QoS vector model with assumption* (4.1) *or* (4.2)*, there exists an ordering of the players* $i \in [1,n]$ *such that* $\forall i \in [1,n-1]$, $\forall j \in [1,m]$,

$$b_{ij} \leq b_{i+1j}.$$

Proposition 4.5 *The game-theoretic results of Section* 3 *hold for the multi-dimensional QoS vector model with assumption* (4.1) *or* (4.2).

The proof structure of our game-theoretic results rely on Proposition 3.1 to order application QoS preferences. The QoS vectors (i.e., scalar packet loss indicator) and their functions affect the proof only through Proposition 3.1. Thus, under either of the uniformity assumptions, and with Proposition 4.4 in hand, it is straightforward to check that the proofs carry over unchanged giving Proposition 4.5.

4.2 Effect of Burstiness on QoS

A consequence of generalizing the QoS provision model to multi-dimensional QoS vectors *without* using either of the uniformity assumptions of the previous section is that there may no longer be a total order on the set of application QoS requirements. That is, whereas in the scalar QoS case (e.g., take packet loss rate), applications could be linearly ordered by the bounds on their packet loss rate, $i \leq i' \iff \theta_i \leq \theta_{i'}$, in the vector QoS case, this is no longer the case and only a partial order can be imposed on the set of QoS requirements $\Theta = \{\, \boldsymbol{\theta}^i : i \in [1,n] \,\}$ where $\boldsymbol{\theta}^i = (\theta_1^i, \theta_2^i, \ldots, \theta_s^i)^T$.

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 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* uniformity holds and 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?

Of course, 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,

$$(\theta_c^1, \theta_\sigma^1) < (\theta_c^2, \theta_\sigma^2) \implies (c_1, \sigma_1) < (c_2, \sigma_2),$$

where $\boldsymbol{\theta}^i = (\theta_c^i, \theta_\sigma^i)$ is the QoS requirement of user $i \in \{1,2\}$ and $\mathbf{x}^j = (c_j, \sigma_j)$ is the QoS rendered at service class $j \in \{1,2\}$.

As a second reference point that is more comprehensive, we are 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

$$(\theta_c^1, \theta_{\sigma^c}^1, \theta_d^1, \theta_{\sigma^d}^1) < (\theta_c^2, \theta_{\sigma^c}^2, \theta_d^2, \theta_{\sigma^d}^2) \implies (c_1, \sigma_1^c, d_1, \sigma_1^d) < (c_2, \sigma_2^c, d_2, \sigma_2^d)$$

where the first two components are as before and the last two components represent mean delay and delay variance[10], respectively.

The answers to both questions turn out to be in the *negative*. We omit the analysis and simulation results including simulation of single-switch noncooperative QoS provision games due to space constraints. We refer the reader to [38] for the full paper.

5 Conclusion and Discussion

We have presented a study of the quality of service provision problem in noncooperative multi-class network environments where applications or users are assumed to be selfish. Users are endowed with heterogenous QoS preferences, and they are allowed to choose both *where* and *how much* of their traffic to send. Our framework and its conclusions are best suited—but not exclusively

[10] 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.

so—for best-effort traffic environments where the network is not required to provide stringent QoS guarantees which can only be accomplished, currently, by employing conservative resource reservations. Rather, service classes with differentiated QoS levels matching the needs of constituent applications are induced by the latter's selfish interactions, providing reasonably stable and predictable QoS levels as a function of network state.

We have formulated a noncooperative multi-class QoS provision model and given a comprehensive analysis of its properties. We have shown that Nash equilibria—which correspond to stable fixed points in noncooperative games—need not be Pareto nor system optimal; in fact, Nash equilibria need not even exist. We have given a complete characterization of Nash equilibria and their existence conditions, and we have studied the game-theoretic structure relating Nash equilibria to Pareto optima and system optima. In general, gaps exist between the classes at all levels, producing a picture of the world that is nontrivial and complex. Much of this is due to the presence of applications with *diverse* QoS requirements, the fact that they are allowed to choose *where* to send their traffic, and the *basic axioms* underlying network systems. For "resource-plentiful" systems, however, we have shown that Nash, Pareto, and system optima all coincide, and moreover, convergence is monotone and fast if a form of asynchronous self-optimization is used.

We have extended the analysis to systems with multi-dimensional QoS vectors containing both mean- and variance-related QoS measures. We have shown that the game-theoretic results carry over if a uniformity assumption is placed either on application preference thresholds or on QoS vector functions. We have stated a subtle but important effect introduced by considering multiple QoS measures—namely, the ordering characteristics of QoS rendered at service classes when weighted fair queueing is employed. Our results show 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 can be shown to hold under self-similar traffic conditions with varying degrees of long-range dependence.

Many interesting and challenging problems remain, some of a mostly technical nature, and others motivated by performance evaluation and practical issues arising out of implementation-related considerations. Current work is directed in two main avenues, one, in the extension of the game-theoretic analysis to arbitrary monotone utility functions and the incorporation of pricing which requires further development of analytical tools and techniques, and two, in the study of *many-switch systems*—a prime target being the realization of such QoS provision architectures in wide area network environments including the Internet. In the latter, the interaction among switches or routers introduces couplings that give rise to new complexities and a slew of challenging distributed control problems. An architecture for noncooperative multi-class QoS provision in many-switch systems can be found in [4, 5].

Acknowledgments The proofs of our results in the paper have been omitted due to space constraints. All proofs can be found in [38]. They are also available on-line as `http://www.cs.purdue.edu/homes/park/noncoop_qos_tech_new.ps.Z`.

References

[1] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, second edition, 1995.

[2] A. Bovopoulous and A. Lazar. Decentralized algorithms for optimal flow control. In *Proc. 25th Allerton Conference on Communication, Control and Computing*, 1987.

[3] A. Campbell, C. Aurrecoechea, and L. Hauw. A review of QoS architectures. To appear in *ACM Multimedia Systems Journal*, 1996.

[4] S. Chen and K. Park. An architecture for noncooperative QoS provision in many-switch systems. Technical Report CSD-TR-98-013, Department of Computer Sciences, Purdue University, 1998.

[5] 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.

[6] Scott Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation, Scott Clearwater*. World Scientific, Hong Kong, 1996.

[7] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in computer networks: motivation, formulation, and example. *IEEE/ACM Trans. Networking*, 1(6):614–627, 1993.

[8] R. L. Cruz. A calculus for network delay, part I: network elements in isolation. *IEEE Trans. Inform. Theory*, 37(1):114–131, 1991.

[9] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE J. Select. Areas Commun.*, 13(6):1048–1056, 1995.

[10] 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):1081–1090, 1995.

[11] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking: Res. Exper.*, 1:3–26, 1990.

[12] 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.

[13] A. Elwalid and D. Mitra. Analysis, approximations and admission control of a multi-service multiplexing system with priorities. In *Proc. IEEE INFOCOM '95*, pages 463–472, 1995.

[14] D. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic models for allocating resources in computer systems. In Scott Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation, Scott Clearwater*. World Scientific, Hong Kong, 1996.

[15] D. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. In *Proc. IEEE INFOCOM '89*, pages 110–118, 1989.

[16] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic algorithms for load balancing in distributed computer systems. In *Proc. 8th International Conference on Distributed Computing Systems*, pages 491–499, 1988.

[17] J. W. Friedman. *Game Theory with Applications to Economics*. Oxford University Press, 1986.

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

[19] A. Heddaya and K. Park. Parallel computing on high-speed wide-area networks: a pricing policy for its communication needs. In *Proc. 3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems*, pages 188–191, 1995.

[20] Man-Tung T. Hsiao and Aurel A. Lazar. Optimal flow control of multi-class queueing networks with decentralized information. In *Proc. IEEE INFOCOM '87*, pages 652–661, 1987.

[21] Y. Korilis and A. Lazar. Why is flow control hard: Optimality, fairness, partial and delayed information. In *Proc. 2nd ORSA Telecommunications Conference*, March 1992.

[22] Y. Korilis and A. Lazar. On the existence of equilibria in noncooperative optimal flow control. *Journal of the ACM*, 42(3):584–613, 1995.

[23] Y. Korilis, A. Lazar, and A. Orda. Architecting noncooperative networks. *IEEE J. Select. Areas Commun.*, 13(7):1241–1251, 1995.

[24] Y. Korilis, A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Trans. Networking*, 5(1):161–173, 1997.

[25] 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*, pages 101–114, 1994.

[26] J. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Trans. on Computers*, 38(5):705–717, 1989.

[27] Jim Kurose. Open issues and challenges in providing quality of service guarantees in high-speed networks. *Computer Communication Review*, 23(1):6–15, 1993.

[28] A. Lazar and G. Pacifici. Control of resources in broadband networks with quality of service guarantees. *IEEE Network Magazine*, pages 66–73, October 1991.

[29] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

[30] S. Low and P. Varaiya. A new approach to service provisioning in ATM networks. *IEEE/ACM Trans. Networking*, 1(5):547–553, 1993.

[31] S. Low and P. Varaiya. An algorithm for optimal service provisioning using resource pricing. In *Proc. IEEE INFOCOM '94*, pages 368–373, 1994.

[32] John-Francis Mergen. Personal communication.

[33] R. Nagarajan and J. Kurose. On defining, computing and guaranteeing quality-of-service in high-speed networks. In *Proc. IEEE INFOCOM '90*, pages 2016–2025, 1992.

[34] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Trans. Networking*, 1(5):510–521, 1993.

[35] 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):344–357, 1993.

[36] 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):137–150, 1994.

[37] 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*, pages 171–180, October 1996.

[38] K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness. Technical Report CSD-TR-97-039A, Department of Computer Sciences, Purdue University, 1998.

[39] Kihong Park. Self-organized multi-class QoS provision for ABR traffic in ATM networks. In *Proc. 15th IEEE International Phoenix Conference on Computers and Communications*, pages 446–453, 1996.

[40] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94*, pages 257–268, 1994.

[41] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

[42] J. Rosenmüller. *The Theory of Games and Markets*. North-Holland, 1981.

[43] 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*, pages 1111–1119, 1995.

[44] Scott Shenker. Making greed work in networks: a game-theoretic analysis of switch service disciplines. In *Proc. ACM SIGCOMM '94*, pages 47–57, 1994.

[45] H. R. Varian. *Microeconomic Analysis*. Norton Press, 1993.

[46] N. N. Vorob'ev. *Game Theory*. Springer-Verlag, 1977.

[47] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta. Spawn: a distributed computational economy. *IEEE Trans. Software Engineering*, 18(2):103–117, 1992.

[48] Michael P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[49] Michael P. Wellman. Market-oriented programming: some early lessons. In S. H. Clearwater, editor, *Market-based control: a paradigm for distributed resource allocation*, chapter 4. World Scientific, 1995.

[50] 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*, pages 100–113, 1995.

[51] Yechiam Yemini. Selfish optimization in computer networks. In *Proc. IEEE Conference on Decision and Control*, pages 281–285, 1981.