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

Kihong Park* Meera Sitharam† Shaogang Chen‡

CSD-TR 97-039
Dept. of Computer Sciences
Purdue University

Abstract

This paper studies the quality of service (QoS) provision problem in noncooperative networks where applications or users are selfish and routers implement a general class of packet scheduling disciplines which includes weighted fair queueing.

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 the first comprehensive analysis of the noncooperative multi-class QoS provision game, giving a complete characterization of Nash equilibria and their existence criteria, and we 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. Our conclusions stand in contrast to previous works on noncooperative network games including congestion control and routing games that depict an overly optimistic picture of the world stemming from restrictive assumptions and special case analysis. 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 quality of service provision 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 many cases, impossible for a service class to deliver quality of service superior in both mean- and burstiness-related QoS measures (e.g., packet loss rate and jitter) when weighted fair queueing is employed at routers. This somewhat surprising result, although general in its scope, has 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 satisfied.

Note: This manuscript includes an Appendix (of proofs) for the reader's reference.

*Dept. of Computer Sciences, Purdue University, West Lafayette, IN 47907; park@cs.purdue.edu.
†Dept. of Math. and Computer Science, Kent State University, Kent, OH 44242; sitharam@mcs.kent.edu.
‡School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907; shaogang@ecn.purdue.edu.

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, 11, 26, 32]. 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 individual utility functions. Users or applications[1] 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 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 [6, 7, 10, 11, 16, 27, 32, 34, 35, 8], 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 [9, 34]. Real-time constraints of multimedia traffic and the scale-invariant burstiness associated with self-similar network traffic [29, 39, 50, 36] 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, 36] 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 [24].

Recently, microeconomic/game-theoretic approaches to resource allocation have received significant interest with application domains spanning a number of different contexts [5, 13, 14, 18, 20, 23, 25, 30, 33, 38, 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 [5, 21, 31, 44, 43, 38]. 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 [5], and others represent flow/congestion control and routing models that only

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

partially address the quality of service problem [21, 33, 44].

Our approach differs from previous works in two significant ways. We devote significant effort to explicating the differences in the modeling assumptions and their relevance to network QoS provision, casting our new results in this light. First, we give a comprehensive noncooperative resource allocation model specifically 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 grounded in networking reality. The latter is worth emphasizing given that some models make concavity (convexity) assumptions that turn out to be contradictory in the network QoS provision context. Such assumptions allow overly optimistic conclusions to be drawn about the noncooperative QoS provision problem, depicting a "rosy" picture of the world that is unwarranted.

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 weighted fair queueing 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. We also make the further assumption that $c_j = c_j(q_j)$ which captures a form of isolatedness[2] (also called insularity or firewall property). 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 deprove 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

[2] 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.

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 any 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. Works that show existence using the concavity assumption on U_i must be interpreted with this caveat in mind. Specifically, the monotonicity conditions, unless concocted, do not give rise to U_i that are quasi-concave[3] (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. However, this is not too surprising given that it is for systems where resource contentions can arise that control algorithms are needed. We also show that for resource-plentiful systems a certain self-optimization procedure leads to quick, robust convergence to globally optimal Nash equilibria.

The previous results point to the need to shift away some of the focus of future research from traditional game-theoretic questions to distributed control and performance evaluation questions since, unless one is dealing with toy-like environments, little may be garnered from answering purely game-theoretic questions, be they in the positive or negative.

We also 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. [33] 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 and they crucially depend on the utility function being concave (convex in the case of their cost function). As mentioned above (and shown more rigorously in Sections 1.4 and 2.3), in the networking context, this assumption is problematic.

[3] 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.

1.3.2 Multi-Dimensional QoS Vectors and Burstiness

We investigate the problem of effectively facilitating quality of service in systems with multi-dimensional 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 weighted fair queueing. 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.

Since one of the goals of the multi-class QoS provision architecture is to provide service classes with stable QoS properties that match the diverse QoS requirements of heterogenous applications comprising the current network state, it is important to understand what QoS demands can be jointly satisfied by the same service class and which are conflicting. Our results show that these relations are nontrivial.

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 [13, 44, 21, 20, 33, 23, 25, 14, 47, 48, 30, 5, 43, 38]. A collection of papers covering a broad range of topics can be found in [4]. A brief survey of some of the literature is provided in [12]. Some standard references to game theory and microeconomics include [1, 15, 41, 45, 46].

Many of the earlier papers including some recent ones [14, 13, 25, 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[4]. 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

[4] 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.

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 [19, 2] with more recent work carried out jointly with Korilis and Orda [20, 21, 22, 23, 33]. Their main work has revolved around an optimal flow control problem, and the development of techniques needed to show the existence of Nash equilibria [21]. Korilis et al. [22, 23] 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 [33] 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 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 [5, 31, 43, 18]. 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 [5], 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 [5] 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. [21] 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 [21] 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 $0 \leq \lambda_{ii} \leq \lambda_i$ [21], 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 [21]—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 [21], [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 [33], 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^i_j 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^i_j 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^i_j (called type-A, type-B, and type-C). In type-B and type-C, the cost function J^i_j 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 [33] 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^i_j 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^i_j is convex is λ_{ij}. In fact, a simple consequence of the monotonicity assumption is that J^i_j 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, lead to consequences worlds apart.

More specifically, the assumption that J^i_j is convex in λ_{ij} is needed in [33] to invoke Rosen's theorem [40], 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^i_j is convex in λ_{ij}. If J^i_j 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 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 and they are allowed to choose both the service classes and traffic volumes assigned to them.

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. In particular, Section 2.3 discusses the differences between our model and the model of Orda et al. [33]. 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 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 weighted fair queueing (WFQ) 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 $\alpha_j \mu^{NR}$.

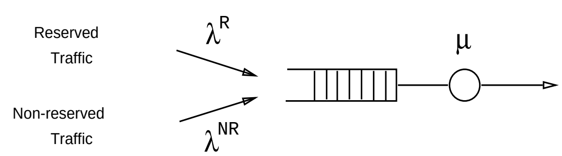

Figure 2.1: Dual traffic classification at output-buffered switch with shared priority queue implementing weighted fair queueing.

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 when performing simulations. We ignore effient 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.

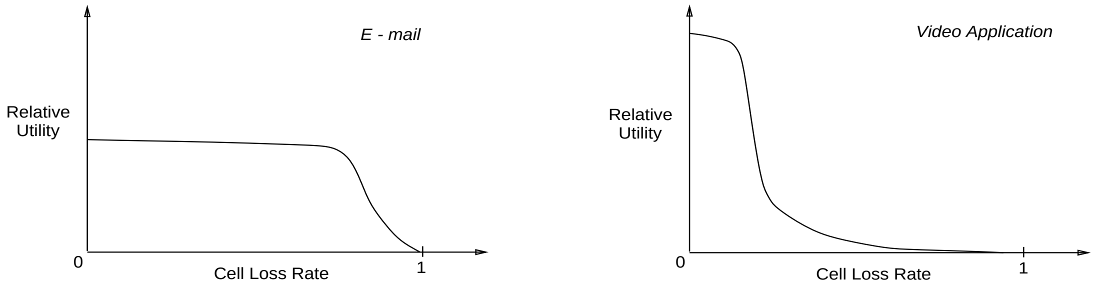

Figure 2.2: Utility functions. E-mail application (left) and video application (right).

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.

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. 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]. 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[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.

[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).

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

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

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,

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

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

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

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[7].

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

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

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

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

[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.

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.

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.

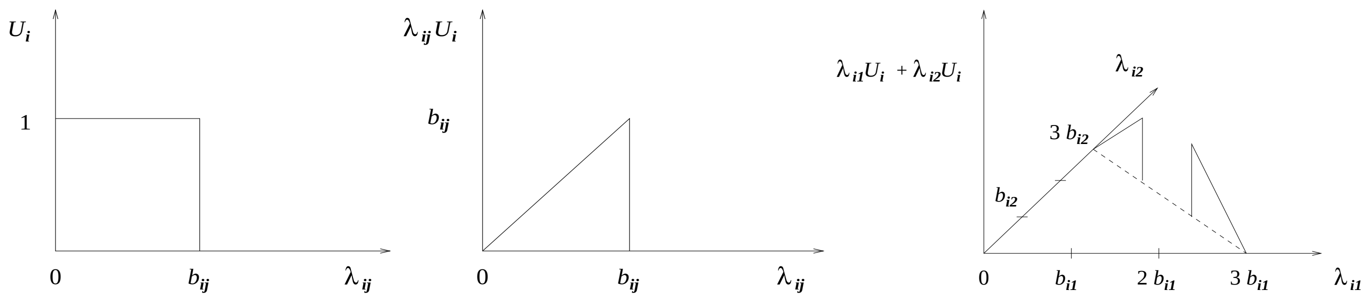

Figure 2.3: Left: U_i as a function of λ_{ij} (via q_j). Middle: $\lambda_{ij} U_i$ as a function of λ_{ij}. Right: $\bar{U}_i = \lambda_{i1} U_i + \lambda_{i2} U_i$ as a function of λ_{i1}, λ_{i2}.

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 $q_j \leq b_{ij}$ $(j = 1, 2)$, and 0, otherwise. Furthermore, assume $b_{i1} = b_{i2}$ (i.e., there is not even a 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 $\{(\lambda_{i1}, \lambda_{i2}) : \bar{U}_i \geq b_{i1}/2\}$ is *not* convex: the projection onto $\mathcal{L}$ yields two separated line segments. This disables the application of Rosen's theorem [40], 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 [33], this is also the rigorous underpinning of the discussion given in Section 1.4 which showed why it is problematic to assume concavity (i.e., convexity for minimization in [33]) in their routing model.

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 one where they are used to compute resource reservations.

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

$$i \leq i' \Longleftrightarrow \theta_i \leq \theta_{i'}.$$

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 $\forall i \in [1, n-1]$, $\forall j \in [1, m]$, $b_{ij} \leq b_{i+1j}$.

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* $J_i^- \setminus \{j^*\} \subseteq I_i^0$.

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 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.

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.

[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.

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*

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

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 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:

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

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

4.2.1 Problem Statement

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 $\boldsymbol{\theta}^i = (\theta^i_1, \theta^i_2, \ldots, \theta^i_s)^T$.

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 weighted fair queueing 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 $X = \{ \mathbf{x}^j : j \in [1, m] \}$, $\mathbf{x}^j = (x^j_1, x^j_2, \ldots, x^j_s)^T$, is also linearly ordered? Of course, to maintain comparability, we will assume that the n input processes are i.i.d. Somewhat counter-intuitively, we will show that the answer is in the *negative*.

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,

$$(\theta^1_c, \theta^1_\sigma) < (\theta^2_c, \theta^2_\sigma) \implies (c_1, \sigma_1) < (c_2, \sigma_2), \tag{4.6}$$

where $\boldsymbol{\theta}^i = (\theta^i_c, \theta^i_\sigma)$ 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 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

$$(\theta^1_c, \theta^1_{\sigma^c}, \theta^1_d, \theta^1_{\sigma^d}) < (\theta^2_c, \theta^2_{\sigma^c}, \theta^2_d, \theta^2_{\sigma^d}) \stackrel{?}{\implies} (c_1, \sigma^c_1, d_1, \sigma^d_1) < (c_2, \sigma^c_2, d_2, \sigma^d_2) \tag{4.7}$$

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

[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.

4.2.2 Qualitative Analysis: Ordering due to Packet Loss

First, we give a qualitative analysis of the packet drop mean/variance ordering question, i.e., implication (4.6). The queueing set-up is the one shown in Figure 2.1 (Section 2.1) with our two applications comprising the nonreserved (i.e., best-effort) traffic.

Let $(\xi(t))_{t\in\mathbb{R}_+}$ denote the 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 instantaneous packet arrivals. We will model *reservedness* by assuming $\xi(t) \le \mu$ and

$$\eta(t) = [\,\mu - \xi(t)\,]^+ \tag{4.8}$$

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

$$c_j(t) = [\,1 - \alpha_j \eta(t)/q_j\,]^+. \tag{4.9}$$

Here, we have used the isolatedness property of WFQ. 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 (4.9). 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

$$\mathbf{V}(c_j) \;=\; \int_{\eta \le \frac{1}{\omega_j}} p(\eta)(1 - \omega_j \eta)^2 d\eta \;-\; \mathbf{E}(c_j)^2 \tag{4.10}$$

since for $\eta \le 1/\omega_j$, $c_j = 1 - \omega_j \eta$. By $\omega_1 \ge \omega_2$, the second moment term in (4.10) 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 (4.10) 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 $p(\eta)$.

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 $c_1(t) = 0$, $t \in \mathbb{R}_+$, it is certainly possible to have

$$\mathbf{E}(c_1) \le \mathbf{E}(c_2), \qquad \mathbf{V}(c_1) \le \mathbf{V}(c_2) \tag{4.11}$$

as desired in (4.6).

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

$$\mathbf{V}(c_j) = \omega_j^2 \mathbf{V}(\xi). \tag{4.12}$$

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

$$\mathbf{E}(c_1) < \mathbf{E}(c_2), \qquad \mathbf{V}(c_1) > \mathbf{V}(c_2). \tag{4.13}$$

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 (4.6) 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.

4.2.3 Numerical Estimation: Ordering of Packet Loss

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 will show the transition behavior, (4.11) $\mapsto$ (4.13), 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 4.1 (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 4.1 (left) shows that mean packet loss is ordered as $\mathbf{E}(c_1) < \mathbf{E}(c_2)$ as expected. In Figure 4.1 (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 4.2.4) 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 4.1 (middle) of the Poisson cross traffic case. That is, the overall contention level as determined by the average input

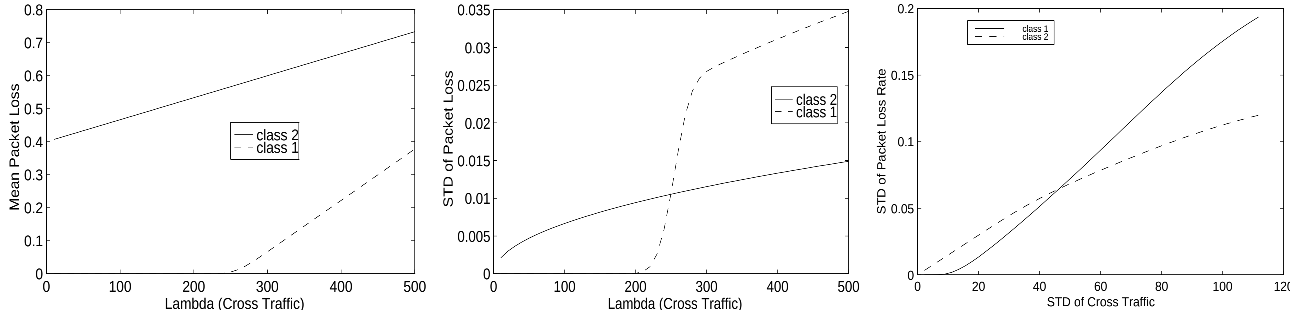

Figure 4.1: Left: Estimated mean packet loss of 2-service class system as a function of Poisson cross traffic parameter λ^R. Middle: Estimated standard deviation of packet loss rate of 2-service class system as a function λ^R. Right: Estimated standard deviation of packet loss rate as a function of standard deviation of white noise cross traffic.

rate is of import; this is discussed further in Section 4.2.4. 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 4.1 (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 (4.12).

4.2.4 Ordering of Packet Loss and Delay: Self-Similar Background Traffic

In this section, we study the ordering problem of the more comprehensive case, implication (4.7), 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 [28, 36] 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 weighted fair queueing (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, and the traces shown in Figures 4.2 and 4.3 depict 1 sec time-aggregations.

Impact of Network Contention Figure 4.2 (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 4.2 (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 4.2 (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*

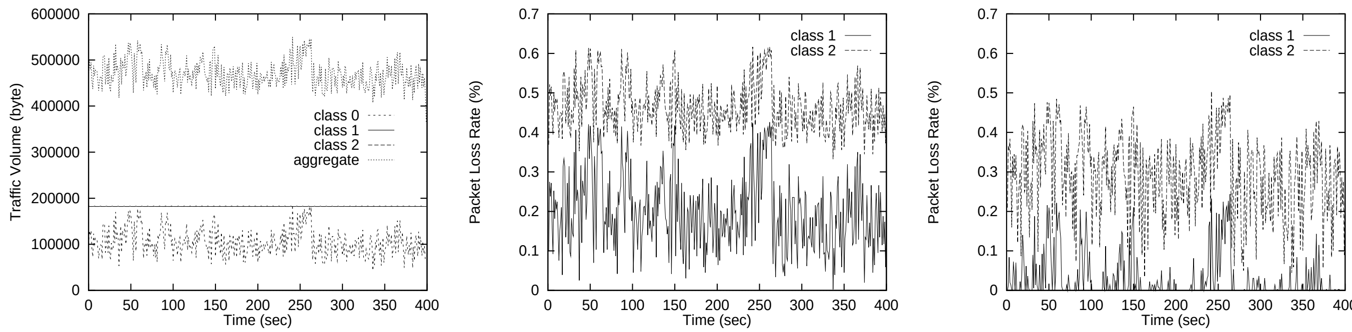

Figure 4.2: Left: Throughput trace of self-similar background traffic (service class 0) and constant application input traffic (service class 1 & 2). Middle: Packet drop trace at bottleneck router with link bandwidth 2.8 Mbps. Right: Packet drop trace at bottleneck router with same set-up except link bandwidth 3.3 Mbps.

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 4.3 (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.

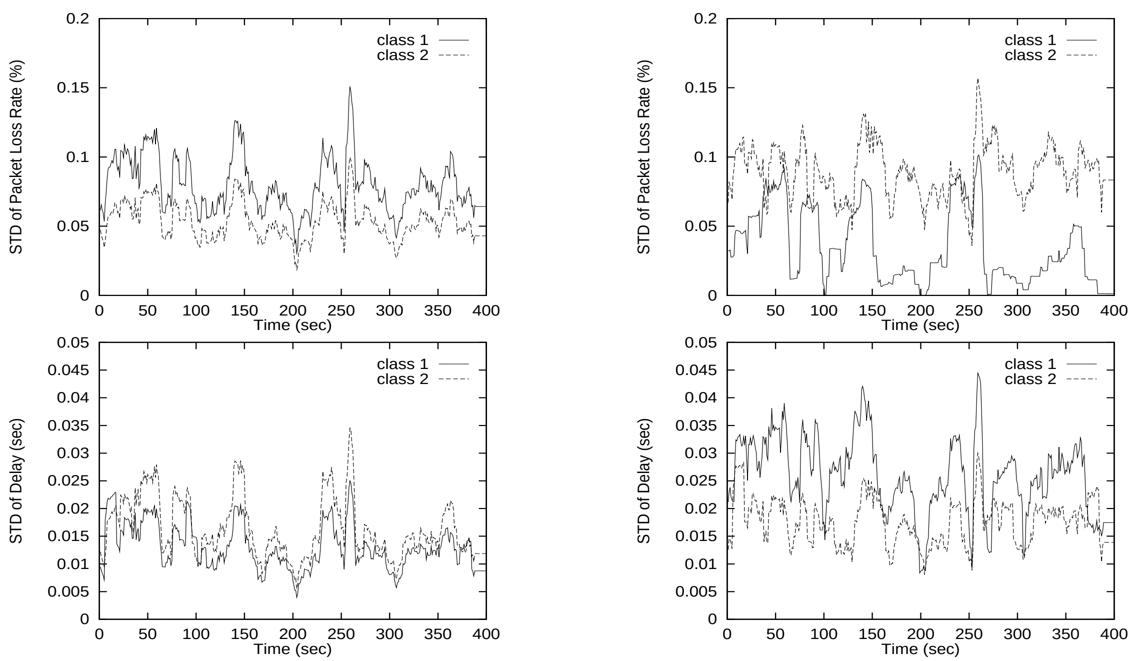

Figure 4.3: Top row: Traces of packet drop standard deviation for bottleneck bandwidth of 2.8 Mbps (left) and 3.3 Mbps (right). Bottom row: Corresponding traces for standard deviation of queueing delay at bottle switch for link bandwidth 2.8 Mbps (left) and 3.3 Mbps (right).

Figure 4.2 (right) and Figure 4.3 (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 4.3 (top-right).

The reasons underlying this phenomenon—dependence on bandwidth (more generally, contention level)—can be traced back to the analysis in Section 4.2.2. 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 (4.13) 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 (4.12) (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 4.1 (middle)) indicates that the transition point may be sharp and thus the transition region small.

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 4.3 (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.

Impact of Self-Similar Burstiness Self-similar traffic with long-range dependence possess a form of "scale-invariant burstiness" [36]. 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[11] values were in the range 0.55–0.95. With respect to the ordering relations (4.6), (4.7), we 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 4.1 (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 [17, 42].

[11] 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.

4.2.5 QoS Ordering: Simulation of 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. Due to space constraints, we give a cursory description of the set-up including the exact QoS provision architecture.

We implement the network set-up described in Sections 2.1 and 4.2.1 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 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

$$\boldsymbol{\chi}^j = \frac{1}{|A_j|} \sum_{i \in A_j} \boldsymbol{\theta}^i. \tag{4.14}$$

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 [44]) 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 will be described elsewhere.

Of import, in the present context, is that all the quantities—including (4.14)—can be easily computed because the information is available in the packet streams.

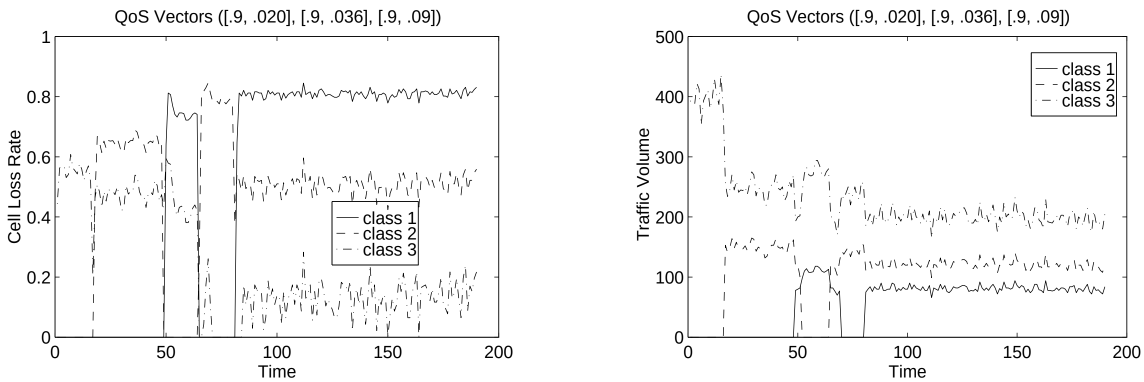

Figure 4.4: Same packet loss requirement but different variance requirements. Left: Cell loss trace shows inverted ordering where service class with least variance has highest cell loss rate. Right: Corresponding traffic volume trace q_1, q_2, q_3.

Effect of Burstiness Figure 4.4 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 4.4 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.

Degenerate Assignment Figure 4.5 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 4.5 (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.

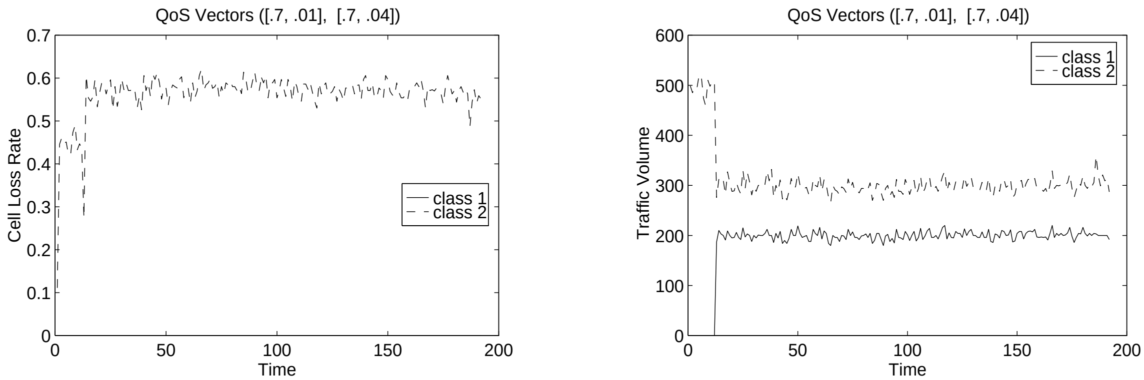

Figure 4.5: Degenerate case where QoS delivered obeys the same order as that required by constituent applications $(0.7, 0.01) < (0.7, 0.04)$. First component is packet loss rate and second component is variance. Left: Shows degenerate QoS rendered for service class 1 where packet loss rate and variance are both 0. Right: Corresponding traffic volume trace.

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 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, 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 studied 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. 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.

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. The main motivating factor, however, of this line of research is that the current structure of the Internet—which provides a single homogeneous clump of best-effort service (not counting reservation-based services)—must be shaped into an architecture which provides stratified services with desirable stability, fairness, optimality, and efficiency properties such the enormous burden to be placed on this enabling infrastructure can be effectively met.

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

[5] 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.

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

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] M. Grossglauser and J-C. Bolot. On the relevance of long-range dependence in network traffic. In *Proc. ACM SIGCOMM '96*, pages 15–24, 1996.

[18] 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.

[19] 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.

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

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

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

[23] Y. Korilis, A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Trans. Networking*, 5(1):161–173, 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*, pages 101–114, 1994.

[25] 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.

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

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

[28] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193, 1993.

[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] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] K. Park, G. Kim, and M. Crovella. On the effect of traffic self-similarity on network performance. To appear in *Proc. SPIE International Conference on Performance and Control of Network Systems*, November 1997.

[38] 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.

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

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

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

[42] 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*, pages 3–14, 1996.

[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.

A Appendix

A.1 Proofs of Section 3.1

Proof of Proposition 3.1. Since $\theta_i \leq \theta_{i+1}$, $i \in [1, n-1]$, by monotonicity of c_j, $j \in [1, m]$,

$$c_j^{-1}(\theta_i) \leq c_j^{-1}(\theta_{i+1}).$$

Noting that $b_{ij} = c_j^{-1}(\theta_i)$ completes the proof. ■

Proof of Proposition 3.2. Since for all $j \in J_i^+$, $U_i(c_j(\Lambda')) = 0$, the upper bound v_i follows immediately. ■

Proof of Proposition 3.3 First, $J_i^* \neq \emptyset$ since $J_i^- \neq \emptyset$. Since $\lambda_i > v_i$ and $J_i^+ = \emptyset$, for at least one $j \in J_i^-$, $q_j > b_{ij}$. This implies that $U_i(c_j(\Lambda')) = 0$. It is easily checked that

$$\max_{\Lambda' \in \mathcal{L}_i(\Lambda)} \bar{U}_i(\Lambda')$$

is achieved by Λ' such that $\lambda'_{i\ell} = b_{i\ell} - q^i_\ell$ if $\ell \neq j^*$, and $\lambda'_{ij^*} = \lambda_i - \sum_{\ell \neq j^*} \lambda'_{i\ell}$, where j^* is some element in J_i^*. Hence, $\bar{U}_i(\Lambda') = \sum_{\ell \neq j^*} b_{i\ell} - q^i_\ell = v_i - (b_{ij^*} - q^i_{j^*})$. ■

Proof of Theorem 3.4 ($\Leftarrow$). Assume $I_i^+ = \emptyset$ (part (a)). Since $\forall j,\ \lambda_{ij} > 0 \Longrightarrow q_j \leq b_{ij}$, we have $\bar{U}_i(\Lambda) = \lambda_i$, the trivial upper bound on $\bar{U}_i$.

Assume (b) holds. By Proposition 3.2, $\bar{U}_i(\Lambda) \leq v_i$ where $v_i = \sum_{j \in J_i^-} b_{ij} - q^i_j$. $I_i^- = \emptyset$ and $J_i^- \subseteq I_i^0$ imply $\bar{U}_i(\Lambda) = v_i$, thus achieving the upper bound which holds for any $\Lambda' \in \mathcal{L}_i(\Lambda)$. Notice that J_i^+, J_i^- do not depend on the actions of player i.

Assume part (c). $I_i^- = \emptyset$ and $\exists\, j^* \in J_i^*$ such that $J_i^- \setminus \{j^*\} \subseteq I_i^0$ imply that $\bar{U}_i(\Lambda) \geq v_i - (b_{ij^*} - q^i_{j^*})$. If $J_i^* = \emptyset$, which holds iff $J_i^- = \emptyset$, then we are done. Assume $J_i^* \neq \emptyset$. Notice that the case $J_i^- \subseteq I_i^0$ is covered by part (b) or (a). Hence, we can assume $j^* \in I_i^+$. $I_i^- = \emptyset$ and $j^* \in I_i^+$ imply that $v_i < \lambda_i$. Thus, we can apply Proposition 3.3 which in conjunction with the lower bound on $\bar{U}_i(\Lambda)$ yields $\bar{U}_i(\Lambda) = v_i - (b_{ij^*} - q^i_{j^*})$.

($\Rightarrow$). We will prove the contrapositive. That is, assuming $\exists\, i \in [1, n]$, given Λ, such that

$$I_i^+ \neq \emptyset \quad \wedge \quad (\, I_i^- \neq \emptyset \ \vee \ J_i^+ = \emptyset \ \vee \ J_i^- \not\subseteq I_i^0\,)$$
$$\wedge \quad (\, I_i^- \neq \emptyset \ \vee \ J_i^+ \neq \emptyset \ \vee \ \forall j^* \in J_i^* : J_i^- \setminus \{j^*\} \not\subseteq I_i^0\,),$$

we will show that Λ is not Nash. There are nine clauses to be considered which are grouped into five cases (i)–(v).

(i) $(I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset)$, $(I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset \wedge J_i^+ = \emptyset)$, $(I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset \wedge J_i^- \not\subseteq I_i^0)$, $(I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset \wedge J_i^+ \neq \emptyset)$, $(I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset \wedge \forall j^* \in J_i^* : J_i^- \setminus \{j^*\} \not\subseteq I_i^0)$. They all have in common the conjunction $I_i^+ \neq \emptyset \wedge I_i^- \neq \emptyset$. The latter implies $\exists\, j,\ j',\ j \neq j'$, such that $\lambda_{ij} > 0$, $q_j > b_{ij}$, and $q_{j'} < b_{ij'}$.

We can construct an assignment $\Lambda' \in \mathcal{L}_i(\Lambda)$ such that $\lambda'_{i\ell} = \lambda_{i\ell}$, $\ell \in [1, m] \setminus \{j, j'\}$, and $\lambda'_{ij} = \lambda_{ij} - \epsilon$, $\lambda'_{ij'} = \lambda_{ij'} + \epsilon$, where $\epsilon = \min\{\lambda_{ij}, b_{ij'} - q_{j'}\}$. This yields

$$\bar{U}_i(\Lambda') - \bar{U}_i(\Lambda) \geq \epsilon$$

from which it follows that Λ is not a Nash equilibrium. It can be easily checked that the argument applies to the other four clauses.

(ii) $(I_i^+ \neq \emptyset \wedge J_i^+ = \emptyset \wedge J_i^+ \neq \emptyset) = \mathsf{F}$. The implication reduces to a tautology.

(iii) $(I_i^+ \neq \emptyset \wedge J_i^- \not\subseteq I_i^0 \wedge J_i^+ \neq \emptyset)$. $J_i^- \not\subseteq I_i^0$ implies that $J_i^- \neq \emptyset$. For $j \in J_i^- \setminus I_i^0$, either $q_j < b_{ij}$ or $q_j > b_{ij}$. If $q_j < b_{ij}$, then the argument from (i) can be applied. Assume $q_j > b_{ij}$. This implies that $U_i(c_j(\Lambda)) = 0$. Since $J_i^+ \neq \emptyset$, for all $j' \in J_i^+$, $j' \neq j$ and $U_i(c_{j'}(\Lambda)) = 0$.

We can construct $\Lambda' \in \mathcal{L}_i(\Lambda)$ such that $\lambda'_{i\ell} = \lambda_{i\ell}$, $\ell \in [1,m] \setminus \{j, j'\}$, and $\lambda'_{ij} = \lambda_{ij} - \epsilon$, $\lambda'_{ij'} = \lambda_{ij'} + \epsilon$, where $\epsilon = q_j - b_{ij}$. We still have $U_i(c_{j'}(\Lambda')) = 0$, however,

$$U_i(c_j(\Lambda')) = b_{ij} - q^i_j > 0$$

since $j \in J^-_i$ and $q'_j = b_{ij}$. Hence Λ is not Nash.

(iv) $(I^+_i \neq \emptyset \wedge J^+_i = \emptyset \wedge \forall j^* \in J^*_i : J^-_i \setminus \{j^*\} \not\subseteq I^0_i)$. $J^+_i = \emptyset$ implies $J^-_i \neq \emptyset$, $J^*_i \neq \emptyset$. In fact, $|J^-_i| \geq 2$. This follows from $\forall j^* \in J^*_i : J^-_i \setminus \{j^*\} \not\subseteq I^0_i$ since $J^*_i \subseteq J^-_i$, and assuming $|J^-_i| < 2$ would imply $J^-_i \setminus \{j^*\} = \emptyset$ which would violate $J^-_i \setminus \{j^*\} \not\subseteq I^0_i$.

Let $j \in J^-_i$, $j' \in J^*_i$, with $j \neq j'$. If $q_j < b_{ij}$, then the argument from (i) applies and we are done. Similarly for j'. Let $q_j > b_{ij}$. If $|I^+_i| \geq 2$, then we can choose $j'' \in I^+_i$ with $j \neq j''$ and apply the argument in (iii) with I^+_i in place of J^+_i. Assume $|I^+_i| = 1$, i.e., $I^+_i = \{j\}$. We need only consider the case $q_{j'} = b_{ij'}$. Notice that $J^-_i \setminus J^*_i \neq \emptyset$ since, if $J^-_i = J^*_i$ then $J^-_i \setminus \{j\} \subseteq I^0_i$ by $|I^+_i| = 1$, which would contradict the assumption $\forall j^* \in J^*_i : J^-_i \setminus \{j^*\} \not\subseteq I^0_i$.

Construct the assignment $\Lambda' \in \mathcal{L}_i(\Lambda)$ such that $\lambda'_{i\ell} = \lambda_{i\ell}$, $\ell \in [1,m] \setminus \{j, j'\}$, and $\lambda'_{ij} = \lambda_{ij} - \epsilon$, $\lambda'_{ij'} = \lambda_{ij'} + \epsilon$, where $\epsilon = q_j - b_{ij}$. Now, $U_i(c_{j'}(\Lambda')) = 0$ but $U_i(c_j(\Lambda')) = b_{ij} - q^i_j$. Since $j \in J^-_i \setminus J^*_i$ and $j' \in J^*_i$,

$$(b_{ij} - q^i_j) - (b_{ij'} - q^i_{j'}) > 0$$

which implies $\bar{U}_i(\Lambda') - \bar{U}_i(\Lambda) > 0$.

(v) $(I^+_i \neq \emptyset \wedge J^-_i \not\subseteq I^0_i \wedge \forall j^* \in J^*_i : J^-_i \setminus \{j^*\} \not\subseteq I^0_i)$. In the proof of (iv), $J^+_i = \emptyset$ was only needed to establish $J^-_i \neq \emptyset$ which we can get from $J^-_i \not\subseteq I^0_i$. Hence the argument of (iv) carries over unchanged. ■

Proof of Lemma 3.5. To the contrary, assume Λ is a Nash equilibrium for the example described in the proposition. Due to the first inequality satisfied by the λ_i's and the b_{ij}'s, it follows that there is a service class j_1 for which $\lambda_{2j_1} = q^1_{j_1} < b_{1j_1}$. Using this observation and applying the Nash characterization from Theorem *3.4* to the player 1, we obtain (without loss of generality, by the choice of j_1),

$$q_{j_1} \leq b_{1j_1}. \tag{A.1}$$

Now, due to the second inequality (b) in the proposition, it follows that service class $j_2 \neq j_1$ has assigned traffic volume

$$q_{j_2} > b_{2j_2}. \tag{A.2}$$

Furthermore, using $(A.1)$ and the third inequality in the proposition,

$$\lambda_{2j_2} \neq 0. \tag{A.3}$$

Moreover, since $b_{1j} < b_{2j}$, for all j, we know from $(A.1)$ that $\lambda_{1j_1} \leq q_{j_1} \leq b_{1j_1} < b_{2j_1}$. Thus we get

$$\lambda_{1j_1} = q^2_{j_1} < b_{2j_1}. \tag{A.4}$$

Using $(A.2)$, $(A.3)$, and $(A.4)$, and applying the Nash characterization from Theorem *3.4* to player 2, we get $q_{j_1} \geq b_{2j_1}$ which contradicts $(A.1)$ since $b_{1j} < b_{2j}$, for all j. ■

Proof of Theorem 3.7. ($\Leftarrow$). First notice that (a) implies the existence of a Nash equilibrium. This follows by observing that since each player is domitable—i.e., the n equations $\sum_{i' \neq i} \lambda_{i'} + \sum_j b_{ij} + a_i$; are satisfied (the a_i act as positive slack constants)—one can always find a configuration Λ where each player is *dominated in each class*. In other words, there is a choice of the $2nm$ assignment variables λ_{ij} and slack variables s_{ij} which will satisfy the nm constraints: $\forall i\ \forall j \sum_{i' \neq i} \lambda_{i'j} = b_{ij} + s_{ij}$ (which is straightforward), which *in addition* satisfy the $2n$ constraints: $\forall i \sum_j \lambda_{ij} = \lambda_i$ and $\forall i \sum_j s_{ij} = a_i$. Next, notice that (b) implies the existence of a

Nash equilibrium because, if Λ satisfies the conditions in (b), then each of the players satisfies one of the three conditions of Theorem 3.4.

($\Rightarrow$). Now we show that the negations of (a) and (b) together imply that every configuration Λ is not Nash. The negation of (a) implies that for each configuration Λ, some player is not dominated in some class. This, together with the negation of (b) implies that for each configuration Λ there is a smallest player i^* which is not dominated in some class, and *either* there is a player $i > i^*$ which does not have complete utility in Λ *or* none of the three Nash conditions holds for the player i^*. In the latter case, clearly Λ is not Nash. In the former case (assuming one of the three Nash conditions holds for the player i^*), it follows that there is some class j^* where $q_{j^*} \leq b_{i^*j^*}$. However, since some player $i > i^*$ does not have full utility in Λ, in order for Λ to be Nash, $q_j \geq b_{ij} > b_{i^*j}$ must hold for every class j due to the strict ordering of thresholds imposed by the statement of the Theorem. Hence it follows that Λ is not Nash. ■

Proof of Theorem 3.8. The proof uses a dynamic update process described in Section 3.3 which turns out to be a useful reasoning tool in this context. Some of the terminology used here are defined (out of sequence) in Section 3.3.

Notice that in the unsplittable case, since players do not move unnecessarily, if a player i moves at some stage in the update process, it attains full utility in a single move (by shifting all of its traffic from one class to another class). Moreover, by Proposition 3.1, once player i moves (and attains full utility) moves by players $i' \leq i$ will not affect the utility of i. In general, once players $n, n-1, \ldots, n-k$ have moved—in that order—the the subsequent moves of the lower players $1, \ldots, n-k-1$ do not affect the (full) utility of the higher players $n, n-1, \ldots, n-k$. Hence the latter players never move again. It follows that a Nash equilibrium Λ is attained by this process, starting from any initial assignment, as soon as the sequence of players (i.e., moves) includes the subsequence $n, n-1, \ldots, 1$. ■

A.2 Proofs of Section 3.2

Proof of Lemma 3.9. Let $S_{i_\Lambda} = \{\, i \in [1,n] : i < i_\Lambda, \bar{U}_i(\Lambda) \neq 0 \,\}$. By the definition of i_Λ, for all $i > i_\Lambda$, $\bar{U}_i(\Lambda) = \lambda_i$, which gives (b). If $S_{i_\Lambda} = \emptyset$, then we are done.

Assume $S_{i_\Lambda} \neq \emptyset$. We will construct an assignment Λ' from Λ such that it satisfies property (a) while preserving (b). Notice that by Theorem 3.4 and $\lambda_{i_\Lambda} > \bar{U}_{i_\Lambda}(\Lambda)$, $q_j \geq b_{i_\Lambda j}$ for all $j \in [1,m]$. Also, by Proposition 3.1, $b_{i_\Lambda j} \geq b_{ij}$ for all $i \in S_{i_\Lambda}$. Let

$$\nu = \lambda_{i_\Lambda} - \bar{U}_{i_\Lambda}(\Lambda), \qquad \pi = \sum_{i < i_\Lambda} \bar{U}_i(\Lambda).$$

To achieve (a), we will distribute the excess utility π into service classes j with $q_j > b_{i_\Lambda j}$ thus nullifying their contribution. To avoid otherwise disturbing the utility assignment, we will move a commensurate amount from ν, *exactly* filling the gap left by π. That is, $q'_j = q_j$, $j \in [1,m]$, in the modified assignment Λ'. If $\nu > \pi$, the reassignment can be achieved in one round. If $\nu \leq \pi$, a refined construction is used that iteratively shrinks the violating player set S_{i_Λ} until it becomes empty. Following is a formal description of the construction.

Case (i). Assume $\nu > \pi$. Let $K^- = \{\, j \in [1,m] : q_j = b_{ij}, \lambda_{ij} > 0, i \in S_{i_\Lambda} \,\}$, $K^+ = \{\, j \in [1,m] : q_j > b_{i_\Lambda j}, \lambda_{i_\Lambda j} > 0 \,\}$. We construct Λ' as follows. For $i \in S_{i_\Lambda}$, $j \in K^-$,

$$\lambda'_{ij} = 0, \qquad \lambda'_{i_\Lambda j} = \lambda_{i_\Lambda j} + \sum_{k \in S_{i_\Lambda}} \lambda_{kj}.$$

For $i \in S_{i_\Lambda}$, $j \in K^+$,

$$\lambda'_{ij} = \lambda_{ij} + \epsilon_{ij}, \qquad \lambda'_{i_\Lambda j} = \lambda_{i_\Lambda j} - \sum_{k \in S_{i_\Lambda}} \epsilon_{kj},$$

where $\epsilon_{ij} \geq 0$, $\sum_{k \in S_{i_\Lambda}} \epsilon_{kj} \leq \lambda_{i_\Lambda j}$, and $\sum_{i \in S_{i_\Lambda}, j \in K^+} \epsilon_{ij} = \pi$. For all other i and j, $\lambda'_{ij} = \lambda_{ij}$.

By construction, $q'_j = q_j$ for $j \in [1, m]$, and since the excess utility π has been transferred into service classes belonging to K^+, we have $\bar{U}_i(\Lambda') = 0$ for $i \in S_{i_\Lambda}$. Hence, $i_{\Lambda'} = i_\Lambda$. Also, notice that

$$\bar{U}_{i_\Lambda}(\Lambda') = \bar{U}_{i_\Lambda}(\Lambda) + \pi$$

since player i_Λ's unutilized traffic volume has been tranferred to service classes in K^- where, by Proposition 3.1, they now count.

Case (ii). Assume $\nu \leq \pi$. We will perform a similar switch as in case (i), however, over (possibly) several rounds each time monotonically shrinking S_{i_Λ} and obtaining a new estimate for $i_{\Lambda'}$ by decrementing the previous estimate.

In the first round, we transfer a traffic volume of ν from players $i \in S_{i_\Lambda}$ with assignments in K^- to service classes belonging to K^+. To preserve, $q'_j = q_j$, $j \in [1, m]$, we transfer an equal amount from player i_Λ's assignments in K^+ to K^-. This is possible since $\nu \leq \pi$. This yields

$$\bar{U}_{i_\Lambda}(\Lambda') = \bar{U}_{i_\Lambda}(\Lambda) + \nu = \lambda_{i_\Lambda}.$$

Thus, $i_{\Lambda'} \leq i_\Lambda - 1$.

If $S_{i_{\Lambda'}} = \emptyset$ then we are done. If $S_{i_{\Lambda'}} \neq \emptyset$, we recursively repeat the switching process with $i_{\Lambda'}$ in place of i_Λ until $S_{i_{\Lambda'}} = \emptyset$. Since the dividing player's index monotonically decreases by at least one in each round, the process terminates in at most $i_\Lambda - 1$ rounds. ■

Proof of Theorem 3.10. Let Λ' be the normal form constructed in the proof of Lemma 3.9. We will prove the following statement from which the theorem follows immediately: Λ' is not system optimal iff there is a Λ^* with $\bar{U}(\Lambda^*) > \bar{U}(\Lambda')$ such that

(a) $\forall i \in [1, n]$, $\bar{U}_i(\Lambda^*) \geq \bar{U}_i(\Lambda')$, and

(b) $\exists\, i \leq i_{\Lambda'}$ such that $\bar{U}_i(\Lambda^*) > \bar{U}_i(\Lambda')$.

That is, Λ' is not Pareto optimal. Note that $\bar{U}(\Lambda') = \bar{U}(\Lambda)$ by the definition of Λ'.

The '$\Leftarrow$' direction of the statement above is trivial. To show the '$\Rightarrow$' direction, we start with a $\tilde{\Lambda}$ with $\bar{U}(\tilde{\Lambda}) > \bar{U}(\Lambda')$, which exists since Λ' is not system optimal. For all $i > i_{\Lambda'}$, $\bar{U}_i(\Lambda') = \lambda_i$, hence any increase in the utility $\bar{U}(\tilde{\Lambda})$ over $\bar{U}(\Lambda')$ must come from one or more $i \leq i_{\Lambda'}$ for which $\bar{U}_i(\tilde{\Lambda}) > \bar{U}_i(\Lambda')$. Indeed, $\bar{U}_i(\Lambda') = 0$ for $i < i_{\Lambda'}$, hence (b) and part of condition (a), i.e., $\forall i < i_{\Lambda'}$, $\bar{U}_i(\tilde{\Lambda}) \geq \bar{U}_i(\Lambda')$, are already satisfied. We will construct Λ^* from $\tilde{\Lambda}$ such that the remaining part of (a), i.e., $\forall i \geq i_{\Lambda'}$, $\bar{U}_i(\tilde{\Lambda}) \geq \bar{U}_i(\Lambda')$, is satisfied as well. Let

$$L^- = \{\, i \leq i_{\Lambda'} : \bar{U}_i(\tilde{\Lambda}) > \bar{U}_i(\Lambda') \,\}, \qquad L^+ = \{\, i \geq i_{\Lambda'} : \bar{U}_i(\tilde{\Lambda}) < \bar{U}_i(\Lambda') \,\}.$$

Clearly, $L^- \cap L^+ = \emptyset$. Moreover, $i_{\Lambda'}$ need not be an element of either L^- or L^+. Let

$$\pi = \sum_{i \in L^-} \bar{U}_i(\tilde{\Lambda}) - \bar{U}_i(\Lambda'),$$
$$\nu = \sum_{i \in L^+} \bar{U}_i(\Lambda') - \bar{U}_i(\tilde{\Lambda}).$$

By $\bar{U}(\tilde{\Lambda}) > \bar{U}(\Lambda')$, we have $\pi - \nu > 0$. We can perform a switch in assignments between players in L^- and L^+, similar to the proof of Lemma 3.9, obtaining an assignment Λ^* which preserves $q^*_j = \tilde{q}_j$, $j \in [1, m]$, and which satisfies $\forall i \in L^+$, $\bar{U}_i(\Lambda^*) = \bar{U}_i(\Lambda')$, $\forall i \in L^-$, $\bar{U}_i(\Lambda^*) \geq \bar{U}_i(\Lambda')$, and for at least one element $i \in L^-$, $\bar{U}_i(\Lambda^*) > \bar{U}_i(\Lambda')$.

Pick any two players $i_- \in L^-$, $i_+ \in L^+$. Then, $\exists j_-, j_+ \in [1, m]$, $j_- \neq j_+$, such that

$$\lambda_{i_- j_-} > 0, \quad b_{i_- j_-} \geq q_{j_-} \quad \text{and} \quad \lambda_{i_+ j_+} > 0, \quad b_{i_+ j_+} < q_{j_+}.$$

The inequalities follow from Lemma 3.9. $j_- \neq j_+$ follows from the inequalities and the fact that if $j_- = j_+$,

$$b_{i_-j_-} \geq q_{j_-} = q_{j_+} > b_{i_+j_+} = b_{i_+j_-},$$

which leads to a contradiction due to the threshold ordering implied by Proposition 3.1.

Let $\epsilon = \min\{\lambda_{i_-j_-}, \lambda_{i_+j_+}\}$. We can move an ϵ amount of i_-'s assignment from j_- to j_+, and an equal amount of i_+'s assignment from j_+ to j_-. By Proposition 3.1, player i_+'s utility strictly increases by ϵ whereas player i_-'s utility strictly decreases by the same amount. The other players' utilities remain undisturbed since the total volume assignment to each service class was held invariant.

Since $\pi - \nu > 0$, this reassignment process can be repeated until a total traffic volume of ν has been shifted from players in L^- to players in L^+ and vice versa. Since $\forall i > i_{\Lambda'}$, $\bar{U}_i(\Lambda') = \lambda_i$, by the definition of ν, we have that $\forall i > i_{\Lambda'}$, $\bar{U}_i(\Lambda^*) = \lambda_i$, and thus $\forall i > i_{\Lambda'}$, $\bar{U}_i(\Lambda^*) \geq \bar{U}_i(\Lambda')$. For players $i < i_{\Lambda'}$, $\bar{U}_i(\Lambda^*) \geq \bar{U}_i(\Lambda')$ remains satisfied since $\bar{U}_i(\Lambda') = 0$.

The only consideration left is player $i_{\Lambda'}$. If $i_{\Lambda'} \notin L_- \cup L_+$, then we are done. If $i_{\Lambda'} \in L_-$, then after the switch operation, either $\bar{U}_{i_{\Lambda'}}(\Lambda^*) \geq \bar{U}_{i_{\Lambda'}}(\Lambda')$—in which case we are done—or $\bar{U}_{i_{\Lambda'}}(\Lambda^*) < \bar{U}_{i_{\Lambda'}}(\Lambda')$. In the latter, we may perform a further switch between player $i_{\Lambda'}$ and players $i < i_{\Lambda'}$ until $i_{\Lambda'}$'s utility has been suffiently increased vis-à-vis $\bar{U}_{i_{\Lambda'}}(\Lambda')$. This is possible since $\pi - \nu > 0$. If $i_{\Lambda'} \in L_+$, and after the switch we still have $\bar{U}_{i_{\Lambda'}}(\Lambda^*) < \bar{U}_{i_{\Lambda'}}(\Lambda')$, then the same process as with $i_{\Lambda'} \in L_-$ can be done yielding the desired ordering result. ■

Proof of Proposition 3.11. The following describes a counter example consisting of a system of 3 players and 3 service classes and an assignment Λ which is Nash and Pareto but not system optimal. As usual, using Proposition 3.1, for each service class j, we can assume that $b_{1j} \leq b_{2j} \leq b_{3j}$.

For service class 1, take $b_{11} = b_{21}$, and $b_{31} = b_{11} + 1$. For service class 2, take $b_{12} = b_{22} = b_{32} = e$ where e is a very small positive quantity. For service class 3, take $b_{23} = b_{33}$ and $b_{13} = s$. Also, let $b_{32} < b_{31} < b_{33}$.

The assignment Λ is defined as follows. The assignments to service class 1 are: $q_1 = \lambda_{11} = \lambda_1 = b_{11}$, and $\lambda_{21} = \lambda_{31} = 0$. The assignments to service class 2 are: $q_2 = \lambda_{22} = \lambda_2 = b_{22} + E$, where E is a very large quantity and $\lambda_{12} = \lambda_{32} = 0$. The assignments to service class 3 are: $q_3 = \lambda_{33} = b_{33}$ and $\lambda_{13} = \lambda_{23} = 0$. This assignment Λ is clearly a Nash equilibrium: $\lambda_{22} = \lambda_2$ is unutilized, but player 2 cannot unilaterally reassign its share to improve its utility. Players 1 and 3 have full utility. Hence the total utility for assignment Λ is $\lambda_3 + \lambda_1$.

This assignment Λ, however, is not system optimal. The total utility can be increased using the following changes to the assignment: the quantity λ_1 can be moved to service class 2 from service class 1 so that the new λ_{11} is now 0, but the new λ_{21} is now equal to λ_1. A part of λ_2 equivalent to the quantity $\lambda_1 + 1$ is moved into service class 3 so that service class 2 now has total volume q_2 that is one less than its previous value. Therefore λ_2 is now partitioned into $\lambda_{23} = \lambda_1 + 1$, with the remainder of λ_2 assigned to λ_{23} while λ_{21} remains 0. Finally, a part of λ_3 equivalent to the quantity $\lambda_1 + 1$ is moved to service class 1 so the volume of service class 1 increases overall by 1 unit, and service class 3 retains the same volume as before. Now λ_3 is partitioned into $\lambda_{31} = \lambda_1 + 1$, with the remainder of λ_3 assigned to λ_{33} while λ_{32} remains 0.

The utility of player 3 remains the same as before, i.e., it has full utility λ_3. The utility of player 1 has decreased from λ_1 to 0 and the utility of player 2 has increased from 0 to $\lambda_1 + 1$. Hence the total utility after completion of the above reassignment is $\lambda_1 + \lambda_3 + 1$ and hence it has increased by 1 overall which shows that the assignment Λ is not system optimal. It is not hard to see that Λ is, in fact, Pareto optimal; i.e., for any assignment Λ' that has higher total utility, there must be at least one player, in particular, player 1, whose individual utility in Λ' is less than that in Λ. ■

Proof of Theorem 3.12. Before we give the proof, we first define a concept that is used often. A *k-flip* is a map from one configuration Λ to another Λ', denoted by a sequence $(i_1, j_1, i_2, j_2, \ldots, i_k, j_k)$ with $\min\{\lambda_{i_1,j_1}, \ldots, \lambda_{i_k,j_k}\} = \nu > 0$ called the *flip value*. The map is defined as follows. The new assignments λ'_{ij}

remain the same as λ_{ij} except in the following cases: for each l with $1 \leq l \leq k$,

$$\lambda'_{i_l, j_{(l+1) \pmod k}} = \lambda_{i_l, j_{(l+1) \pmod k}} + \nu, \quad \lambda'_{i_l, j_l} = \lambda_{i_l, j_l} - \nu.$$

Notice that a flip leaves total volumes unchanged in all classes. Also, player i_l's utility does not decrease if it holds that:

$$q_{j_l} \leq b_{i_l j_l} \Rightarrow q_{j_{(l+1) \pmod k}} \leq b_{i_l j_{(l+1) \pmod k}}.$$

In fact, player i_l's utility strictly increases if $q_{j_l} > b_{i_l j_l}$, whereas $q_{j_{(l+1) \pmod k}} \leq b_{i_l j_{(l+1) \pmod k}}$. Notice that 2-flips have already been used extensively in earlier proofs.

($\Rightarrow$). To show (a), assume to the contrary, i.e., $\exists i < i^* \; \exists j \in I_i^+ \setminus \{j : q_j > b_{i^*j}\}$, in particular, $j \in I_i^+ \setminus I_{i^*}^+$. Since i^* has incomplete utility, we know that $I_{i^*}^+ \neq \emptyset$, and by Theorem 3.4, we know that $q_j = b_{i^*j}$. Let $j^* \in I_{i^*}^+$. Now, we obtain Λ' from Λ by performing the 2-flip (i, j, i^*, j^*), which ensures that the individual utilities of all players except i^* remain unchanged and i^*'s utility increases by the flip value. This contradicts that Λ is Pareto.

Now (b) follows from (a) and the fact that Λ is Nash (the conditions of Theorem 3.4 applied to i^*), and $i > i^* \Rightarrow I_i^+ = \emptyset$.

To show (c), assume to the contrary that there is a path from $i_b \in S_2$ to $i_a \in S_1$ in G.

Case 1: ($i_a = i_b = i$). Consider a class $j_a \in I_{i'}^+$ for some $i' \leq i^*$, such that $\lambda_{i,j_a} > 0$. The class j_a causes i to be in S_1. Consider also a class j_b with $\lambda_{ij_b} > 0$ and $q_{j_b} \leq b_{i^*j_b}$ which causes i to be in S_2.

Case 1 (i): ($i' = i^*$). That is, $j_a \in I_{i^*}^+$. Now, we obtain Λ' from Λ by performing the 2-flip (i, j_b, i^*, j_a), which, using the definition of j_a and j_b, ensures that the individual utilities of all players except i^* remain unchanged, and i^*'s utility increases by the flip value. This contradicts that Λ is Pareto.

Case 1 (ii): ($i' \neq i^*$). We have $j_a \notin I_{i^*}^+$. Pick a class $j_c \in I_{i^*}^+$. First obtain Λ'' using the 2-flip (i^*, j_c, i', j_a), which ensures (using part (a)) that all players in Λ'' have the same utility as in Λ, and therefore, Λ'' is Pareto if Λ is Pareto. Now, in fact, in Λ'', it holds that $j_a \in I_{i^*}^+$, and thus the proof of Case 1 (i) can be directly employed to contradict the fact that Λ'' is Pareto thereby contradicting the fact that Λ is Pareto.

Case 2: ($i_a \neq i_b$). Consider the class j_b that causes i_b to be in S_2 and the class j_a that causes i_a to be in S_1.

Case 2 (i): ($i' = i^*$). That is, $j_a \in I_{i^*}^+$. Since there is a path from i_b to i_a in G, say $i_b = i_1, i_2, \ldots, i_k = i_a$, we use the definition of the edges of G to construct a flip sequence as follows. The existence of the edges (i_l, i_{l+1}) for $1 \leq l < k$ implies the existence of classes $j_b = j_1, j_2, \ldots, j_k = j_a$, such that the flip sequence $(i_b = i_1, j_b = j_1, i_2, j_2, \ldots, i_k = i_a, j_k = j_a)$ has non-zero flip value. Moreover, by the definition of the edges of G, and using the definition of j_a and j_b, we obtain Λ' from Λ by performing this k-flip which ensures that the individual utilities of all players except i^* remain unchanged, and i^*'s utility increases by the flip value. This contradicts the fact that Λ is Pareto.

Case 2 (ii): ($i' \neq i^*$). A preprocessing is performed exactly like Case 1 (ii), and thereafter, the proof of Case 2 (i) is applied.

To show (d), assume to the contrary that there is a system optimum configuration M of the modified game as well as a service class j^* for which the negations of (d1), (d2), and (d3) hold:

- $\sum_{i=0}^{n} \gamma_{ij^*} \neq b_{i_{j^*}j^*}$ when $i_{j^*} \geq 1$ is defined; this implies $\gamma_{ij^*} < b_{i_{j^*}j^*}$ since $\bar{U}_i(M) = \gamma_i$ for $i \neq 0$.
- $\sum_{i \neq 0} \gamma_{ij^*} < b_{i^*j^*}$
- $\gamma_0 \leq b_{i^*j^*} - \sum_{i \neq 0} \gamma_{ij^*} + \sum_{j' \neq j^*} b_{i_{j'}j} - \sum_i \sum_{j' \neq j^*} \gamma_{ij}$.

We can now create a configuration Λ' from Λ (in fact, from M)—where the utility of no player decreases and that of i^* increases—as follows. Beginning with the service class j^* and the player i^*, we assign $\lambda'_{i^*j^*} \equiv$

$\gamma_{i^*j^*} + \min\{\lambda_{i^*} - \gamma_{i^*}, b_{i^*j^*} - \sum_{i\neq 0}\gamma_{ij^*}\}$. The remaining unallocated volumes (of all players) are now allocated to the classes in any manner that satisfies:

(i) $\forall i\ \forall j\ \lambda'_{ij} \geq \gamma_{ij}$,

(ii) $\forall j \neq j^* \sum_i \lambda'_{ij} \leq b_{i_j j}$,

(iii) $\sum_i \lambda'_{i^*j^*} \leq b_{i^*j}$.

It is clear that such an allocation is always possible since the γ_{ij} and $b_{i_j j}$ satisfy the negations of (d1), (d2) and (d3) listed above. Now, because of (i), (ii) and (iii), it follows that $\forall j \neq j^*$, the amount that each player i contributes to its utility $\bar{U}_i(\Lambda')$ *through* the class j is at least γ_{ij}, and in fact the player i^* contributes strictly more than $\gamma_{i^*j^*}$ *through* class j^*. Since M was chosen so that $\forall i\, \gamma_i = \bar{U}_i(\Lambda)$, we have now exhibited a Λ' which shows that Λ is not Pareto.

($\Leftarrow$). We assume Λ is not Pareto and derive a contradiction to part (d). If Λ is not Pareto, without loss of generality, there is a Λ' where the individual utilities of all players are at least as large as in Λ, and in fact, the utility of the player i^* strictly increases in going from Λ to Λ'. But each such configuration Λ' corresponds to a configuration M' of the modified game (*based on* Λ') with

$$\bar{U}(M') = \sum_{i=1}^{n} \gamma'_i = \sum_i \bar{U}_i(\Lambda'). \tag{A.5}$$

Clearly, each such configuration M' embeds a configuration M of the modified game (*based on* Λ) with $\bar{U}(M) = \sum_{i=1}^{n} \gamma_i = \sum_i \bar{U}_i(\Lambda)$. By "embed" we mean that

$$\forall j : \sum_{i=0}^{n} \gamma'_{ij} = \sum_{i=0}^{n} \gamma_{ij}, \qquad \forall i\ \forall j : \gamma_{ij} \leq \gamma'_{ij}, \qquad \text{and} \quad \exists j^* : \gamma_{i^*j^*} < \gamma'_{i^*j^*}.$$

Now consider the class j^* where $\gamma_{i^*j^*} < \gamma'_{i^*j^*}$. For this j^*, clearly (d1) does not hold: otherwise, $\sum_{i=0}^{n} \gamma'_{ij^*}$ exceeds $b_{i_{j^*}j^*}$, which means that $\gamma'_{i_{j^*}} \geq \gamma_{i_{j^*}} > 0$ would not contribute to the utility of M', contradicting equation (A.5).

Clearly, (d2) does not hold either: otherwise, $\sum_{i\neq 0} \gamma'_{ij^*} > b_{i^*j^*}$, which means that $\gamma'_{i^*j^*}(> \gamma_{i^*j^*} \geq 0)$ would not contribute to the utility of M', again contradicting equation (A.5).

Finally, (d3) does not hold: otherwise, since (by the fact that (d2) does not hold) $\sum_{i\neq 0} \gamma'_{ij^*} \leq b_{i^*j^*}$, it would follow that for some class j', $\sum_{i=0}^{n} \gamma'_{ij'} > b_{i_{j'}j'}$. But this would result in $\gamma'_{i_{j^*}} > 0$ not contributing to the utility of M', again contradicting equation (A.5). ■

A.3 Proofs of Section 3.3

Proof of Theorem 3.13. It is sufficient to show that every Nash equilibrium Λ is system optimal with utility $\bar{U}(\Lambda) = \sum_i \lambda_i$. The equivalence of Nash, Pareto, and system optima follows immediately.

Due to the inequality in (3.14), for an assignment Λ, each player can always unilaterally reassign its λ_{ij}'s and strictly increase its own utility unless the following holds:

$$\forall i \forall j : \ \lambda_{ij} \neq 0 \implies q_j \leq b_{ij}. \tag{A.6}$$

Thus Λ is a Nash equilibrium (i.e., such a reassignment is impossible) only if (A.6) holds. But (A.6) is equivalent to

$$\forall i \forall j : \ q_j > b_{ij} \implies \lambda_{ij} = 0,$$

which, in turn, implies that Λ is system optimal.

Note that if (A.6) holds for Λ, then clearly no player contributes to any service class where the contribution would be unutilized—i.e., every player has complete utility and thus $\bar{U}(\Lambda) = \sum_i \lambda_i$. Hence Nash, Pareto, and system optima are all equivalent. ∎

Proof of Theorem 3.15. To show that the process $\mathcal{P}$ converges to a Nash equilibrium starting from any initial configuration, notice that

(1) When it is player i's turn to move, if $\bar{U}_i < \lambda_i$—the player has less than full utility—then it can always unilaterally reassign its λ_{ij}'s and achieve full utility. In other words, it can achieve the status described in (A.6). Otherwise, if player i has full utility, it does not move at all, i.e., it keeps its current assignment.

(2) Once player i has moved, the subsequent moves of players with indices $k < i$ will not affect i's (full) utility. This is due to the inequality in Proposition *3.1*, and because of the observation in (1): the move of such a player k does not *newly* cause the traffic volume q_j of any service class to cross the threshold $b_{kj} \leq b_{ij}$.

Thus, once player n has moved, it achieves full utility, and subsequent moves of the other players does not affect its utility; hence player n never moves again. In general, once players $n, n-1, \ldots, n-k$ have moved, in that order, the subsequent moves of the lower players $1, \ldots, n-k-1$ do not affect the (full) utility of the higher players $n, n-1, \ldots, n-k$, and hence they never move again. It follows that a Nash equilibrium Λ is attained by the process $\mathcal{P}$, starting from any initial assignment, as soon as the sequence of players (i.e., moves) includes the subsequence $n, n-1, \ldots, 1$. ∎

A.4 Proofs of Section 4.1

Proof of Proposition 4.4. We will consider both uniformity assumptions on the multi-dimensional QoS vectors and thresholds simultaneously.

First, we consider the uniformity assumption (4.1) which states that the thresholds θ_r^i can be ordered such that the ordering is uniform over $r \in [1, s]$. Using this ordering and the monotonicity of x_r^j for each $j \in [1, m]$ and $r \in [1, s]$, by the definition of the b_{ij}^r, we can conclude that

$$\forall r \in [1, s], \forall j \in [1, m], \forall i \in [1, n-1], \quad b_{ij}^r \leq b_{i+1j}^r.$$

Now for any fixed i, j, let r' satisfy $\min_{r\in[1,s]} b_{ij}^r$ and let r'' satisfy $\min_{r\in[1,s]} b_{i+1j}^r$. Clearly, $b_{ij}^{r''} \leq b_{i+1j}^{r''}$, Furthermore, $b_{ij}^{r'} \leq b_{ij}^{r''}$, since b_{ij}^r is minimized at $r = r'$. It therefore follows that the same ordering on i also satisfies

$$\min_{r\in[1,s]} b_{ij}^r \leq \min_{r\in[1,s]} b_{i+1j}^r$$

from which the proposition follows immediately.

Next, we consider the uniformity assumption (4.2) which states that the functional forms x_r^j in the QoS vector $\boldsymbol{x}^j$ are uniform over $r \in [1, s]$ for each $j \in [1, m]$. In this case, we can define a natural ordering on i induced by

$$\min_{r\in[1,s]} \theta_r^i \leq \min_{r\in[1,s]} \theta_r^{i+1}.$$

Since the x_*^j's are all monotone, and as observed previously, $b_{ij} = (x_*^j)^{-1}(\min_{r\in[1,s]} \theta_r^i)$, this ordering yields the required

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

which holds for all $j \in [1, m]$ and $i \in [1, n-1]$. ∎