Pricing in Computer Networks: Motivation, Formulation, and Example

Ron Cocchi[1,2], Scott Shenker[3], Deborah Estrin[2], Lixia Zhang[3]

November 18, 1993

Abstract

We study the role of pricing policies in multiple service class networks. We first argue that some form of service-class sensitive pricing is required for *any* multiclass service discipline[4] to attain the desired level of performance. Borrowing heavily from the Nash implementation paradigm in economics, we then present an abstract formulation of service disciplines and pricing policies. This formulation allows us to describe more clearly the interplay between service disciplines and pricing policies in determining overall network performance. Effective multiclass service disciplines allow networks to focus resources on performance sensitive applications, while effective pricing policies allow us to spread the benefits of multiple service classes around to all users, rather than just having these benefits remain exclusively with the users of applications that are performance sensitive. Furthermore, service disciplines and pricing policies combine to form the incentive system facing a user; these incentives must be carefully tuned so that user self-interest leads to optimal overall network performance. Finally, we illustrate some of these concepts through simulation of several simple example networks. In our simulations, we find that it is possible to set the prices so that users of every application type are more satisfied with the combined cost and performance of a network with service-class sensitive prices. For some application types the performance penalty received for requesting a less-than-optimal service class is offset by the reduced price of the service. For the other application types the monetary penalty incurred by using the more expensive, higher-quality service classes is offset by the improved performance they receive.[5]

1 Introduction

Recent research on computer networks has been concerned almost exclusively with the hardware, software, and protocol standards needed to achieve better network performance. This research program has been an outstanding success. Today's computer networks link thousands of institutions and have become an indispensable part of the academic and industrial communication infrastructure. These networks support a wide variety of applications, including terminal connections, file

[1]Hughes Aircraft Company.

[2]Department of Computer Science, University of Southern California.

[3]Palo Alto Research Center, Xerox Corporation.

[4]Later in the paper we will present an abstract definition of a service discipline, but for now it is sufficient to equate the service discipline with the packet scheduling algorithm in network switches. A single class service discipline treats all packets as equivalent (e.g., FIFO), whereas a multiclass service discipline can treat the various classes of packets differently (e.g., different priority levels).

[5]This paper is an extensively revised version of Reference [2].

transfers, electronic mail, X-server connections, voice, and video. Furthermore, significantly faster and more sophisticated networks are currently being designed and prototyped; it is expected that these networks will spark a whole new generation of applications.

However, such technical progress is not the only important issue affecting network performance. Network performance, at least from the perspective of end users (applications), is not completely determined by the technical characteristics of the network. Network performance is also a function of the offered load. This is analogous to the fact that one's driving time is not just a function of the top speed of the vehicle or the speed limit of the road, but also depends on the level of traffic on the road. The aggregate traffic load on a network is the result of many users' individual decisions about whether and how to use the network, and these decisions are affected by the incentives these users encounter when using the network. Therefore, in addition to the technical specifications of the network, the issue of user incentives must be considered when discussing network performance from the perspective of end users. Note that these user incentives can take many forms: performance incentives, monetary incentives, administrative incentives, or social incentives, to name a few.

This paper represents an initial effort to grapple with user incentives in multiple service class networks. We restrict our focus to one particular aspect of user incentives; the intertwining of pricing policies, which produce monetary[6] incentives, and multiclass service disciplines, which produce performance incentives. Our goal is to articulate precisely some fundamental issues involved in the interplay between pricing policies and multiclass service disciplines and lay the groundwork for future research.

There are many other important issues related to pricing in computer networks. For instance, pricing will be affected by the market structure of network service, the regulatory environment, the cost structure of the various relevant technologies, and issues of capacity expansion and cost recovery. Pricing must also take into account the nature of the demand for network service, both in terms of its price elasticity and its variability, and the nature of the services offered by the network. We explore none of these issues here[7], since we are not attempting to address the full set of economic issues related to pricing in networks. To the contrary, we are only interested in pricing to the extent that it affects the effectiveness of multiclass service disciplines[8]. Consequently, for the sake of simplicity, we look at a rather narrow problem; we assume that users can only vary the class of service they request (i.e., the volume of their traffic is fixed) and consider normative criteria for evaluating pricing policies (instead of looking at the prices that would result from particular competitive or regulatory structures).

As suggested in the title, our work has three main components: motivation, formulation, and an example. In Section 2 we review certain technical and institutional characteristics of the current Internet, and then discuss how these are likely to change in near future. We believe that these predicted changes will bring the interaction between pricing and multiclass service

[6] *Pricing* can refer to forms of incentives other than money; for instance, one can price service in terms of administrative incentives such as quotas or a log. While our framework could be generalized to these other forms of pricing, for the sake of simplicity we will refer only to monetary incentives in this paper.

[7] See [9, 10, 12, 16, 22, 23, 24] for a discussion of these and other issues; References [9, 10, 23] provide a particularly useful complement to this paper.

[8] Similarly, we discuss multiclass service disciplines only to the extent that they interact with pricing issues; see [1] for a fuller description of the network issues.

disciplines to the forefront of the networking community's research agenda. We then present an abstract formulation of our problem in Section 3. This formulation allows us to articulate precisely the roles of service disciplines and pricing policies and then discuss their interaction. Lastly, in Section 4, we present an example of the interplay between pricing and service disciplines. We consider a network with a simple two-priority service discipline and several different application types (electronic mail, packetized voice, file transfer, and interactive terminal connection) running over standard transport-layer protocols. Using packet-level simulations of this network we then compare two different pricing policies: (1) flat pricing, where a uniform per-byte price is charged and is therefore service-class insensitive, and (2) priority pricing, where a higher per-byte price is charged for the high priority traffic and is therefore service-class sensitive. Keeping the overall level of revenue generated fixed and measuring user satisfaction as a function of both the cost and quality of service received, we find that in each of the several network configurations simulated one can always find priority prices such that (1) the users of every application type[9] are more satisfied with the service-class sensitive priority pricing scheme than they were with the service-class insensitive flat pricing scheme, and (2) these users, when maximizing their own satisfaction, choose priority settings that maximize the overall network efficiency. Thus, appropriate pricing policies allow us to both achieve maximal system efficiency and also spread the benefits of multiple service classes around to the users of all application types, rather than just having these benefits remain exclusively with the users of applications which are performance sensitive. The problem we address is similar to and the conclusions we reach are consistent with those of the *priority pricing* literature in economics, which discusses the supply of nonstorable goods like electrical power (see Reference [26]). Our discussion is quite different from the standard literature on externalities (as represented by [8]) and also quite different from the literature on telephony pricing (as represented by [14]). We delay a detailed discussion of related work and other relevant issues until Section 5, and then conclude in Section 6.

2 Motivation

Given the paucity of previously published papers on the topic, it is natural to ask: why is the question of pricing relevant to computer networks? We address this question by reviewing the characteristics of the current Internet, and then discussing how these characteristics are likely to change in the near future.

Today's Internet has four characteristics that are relevant to our discussion. First, the overall bandwidth is quite limited, as the backbone is currently comprised of T1 and T3 lines. This limited bandwidth prevents the widespread usage of certain bandwidth-intensive applications from utilizing the Internet; for instance, HDTV generates a 120Mbit/sec data stream under 8:1 encoding. A second characteristic of the Internet is restricted access; NSF's Acceptable Use Policy (AUP) allows only educational and research institutions to access the Internet[10]. These access restrictions, in addition to controlling the size of the user population, help preserve the cohesive

[9] As will be explained in Section 4, in our example we consider the set of users who are using a particular application type as a single entity for the purposes of the analysis.

[10] We should note that Internet service is now becoming available to the public through various private service providers such as PSI, ALTERnet, and Netcom.

nature of the user community. The shared history of creating the network, the large degree of control over technical decisions, the commonality of end system hardware and operating systems, and the preservation of the relatively small user community have produced widespread adherence to socially desirable behavioral norms. Third, the Internet offers a single class of service;[11] all packets are serviced on a best-effort, first-in-first-out (FIFO) basis. This single service class severely limits the nature of applications that can be adequately supported. Fourth, there are no *usage* fees; users are not charged on the basis of how many packets they send. That is not to say that the networks are free; most institutions are charged for access to a regional network. However, these fees are not based on the volume of traffic sent (although they often depend on the capacity of the connection to the regional network), and in many cases are not passed back to individual users[12].

Pricing has not been critical in today's Internet. However, the Internet of the future is likely to be quite different from today's Internet with respect to each one of these aforementioned characteristics; these changes will likely render the issue of pricing much more relevant than it is today. Below, we discuss each one of these changes in turn.

First, all of the 1.5 Mbps T1 backbone links are being upgraded to 45 Mbps T3 lines; further bandwidth increases are expected and gigabit lines, a thousand times the speed of the current T1 lines, are technically within reach. This dramatic increase in aggregate bandwidth will allow the widespread use of new bandwidth-intensive applications. Perhaps more importantly, it will also allow the Internet to service a much larger user community.

Second, this increase in bandwidth, combined with the widespread availability of both personal computers and residential ISDN, makes it likely that the future Internet, or networks like it, will be accessible to the public. No longer will the network have, by virtue of artificial access restrictions, a small, technically knowledgeable, and mostly cooperative user community. As with other widely used public facilities, informal enforcement of behavioral norms is unlikely to be sufficient to ensure socially desirable behavior. Thus, we expect that users will make service class selections which are in their own personal best interest, regardless of the effect on the overall functioning of the network.

Third, the future traffic control mechanisms, by which we mean the switch queueing algorithms as well as the host congestion control algorithms, are likely to be much more sophisticated than the single service class in the current Internet. There is active debate as to exactly what form these controls will take (reservation vs. best-effort, connections vs. datagrams, dumb hosts and smart switches vs. smart hosts and dumb switches, etc.). See Reference [1] for a brief overview of the literature and one specific proposal. However, the proposed mechanisms all have the same goal of supporting a wide variety of service classes. This agreement on the goal, despite the disagreement on the means, is due to widespread consensus on two points.

One point of consensus is that applications have very different service requirements. For instance some applications, like electronic mail, can tolerate significant delay without users experiencing discernible performance degradation, while other applications, such as packetized voice, degrade perceptibly with even extremely small delays. Similarly, some applications are relatively insensitive to packet loss while others are not, and some applications can adjust to reduced bandwidth while

[11] We will use the terms service class, quality-of-service (QOS), and type-of-service (TOS) interchangeably. A service discipline that has a TOS mechanism is one in which there are multiple service classes.

[12] In most cases, the cost of the Internet connection is considered institutional overhead.

others cannot. The range of applications, and the diversity of service requirements, is likely to grow rapidly in the near future. Thus, it is crucial for the evolution of the Internet that means be found for meeting these increasingly varied service requirements.

The other point of consensus is that networks can more efficiently meet these varied service requirements if the network offers multiple classes of service, so that a user can choose the class of service that is appropriate for her application.[13] The network can then, in periods of resource contention, focus its resources on the performance sensitive applications and avoid squandering them on applications that are not performance sensitive. Typically, not all service classes get better performance under this multiple service class scheme. For example, traffic in a lower quality service class at times will receive worse performance than it would in the current single class of service scheme. The purpose of multiple service classes is to degrade performance for those applications that are least sensitive in order to improve performance for those that are most sensitive[14].

Lastly, in contrast to the current Internet, we believe it is likely that portions, if not all, of the future Internet will implement usage fees regardless of how the cost of the network is financed.[15] This prediction is perhaps the most controversial of those made here, but we feel that the presence of multiple service classes makes this inevitable. Multiple service classes introduce the issue of performance incentives. Users naturally want good performance from their network. Once they are equipped with an action that influences the performance they receive, there is immediately an incentive to request the service class that maximizes their performance. In the absence of any other consideration, there is nothing to motivate a user to indicate that her application is less performance sensitive (which would thereby degrade the performance she receives under some network conditions). Perhaps in a small and cooperative user community the behavioral norm of requesting the appropriate service class can be enforced informally. But, in a public network with a large and relatively anonymous user community, we do not expect that such informal enforcement mechanisms will be sufficient. However, by pricing the service classes appropriately, one can offer monetary incentives for reducing the quality of service requested. We expect pricing of the various service classes to be a vehicle commonly used to encourage users to make reasonable choices[16].

---

[13]Note that this second point of consensus is not entirely trivial; one might contend that building a single class network with sufficient speed to meet the most stringent performance requirements is easier than building a slower network with multiple service classes. Also, we have used the term *appropriate* in two ways. *Appropriate* service classes are not those that maximize each individual user's performance, but are instead those that maximize overall system efficiency; this will be discussed more fully in Section 3. *Appropriate* pricing policies are, in the terminology of Section 3, adoptable and acceptable pricing policies.

[14]We hasten to add that network performance is measured along many different dimensions (delay, packet loss, delay jitter, bandwidth, etc.). Thus, it is a convenient, but occasionally misleading, simplification to talk about better or worse service.

[15]Again, these fees might not be monetary in nature, but rather quotas or some other administrative form. We will, for the sake of brevity, include all such mechanisms under the umbrella of monetary incentives in this paper. Furthermore, we should note that there is significant disagreement about how to best finance computer networks. Some argue for continued heavy subsidies from government, others argue for self-financing through usage fees. We are not entering this debate. We are merely assuming that, in order to make multiclass service disciplines viable, some service-class dependent usage fees are necessary as incentives to make users choose the appropriate service classes. Finally, let us clarify that these usage fees may be in addition to the traditional access fees, and that these usage fees may vary depending on the level of congestion in the network. Our point is that in contrast to the present situation, we expect that in the future *some* network charges will be assessed based on usage; this does not imply that all such charges will be, or that such usage charges will apply at all times.

[16]We should reiterate that there is another point of view on this issue. Some contend that capacity (in both

Thus, we believe that pricing will be an integral part of the future Internet, and that the design of appropriate pricing policies will be a crucial enabling technology. One key question is: can we set prices in such a way that the performance penalty received for requesting a less-than-optimal service class is offset by the reduced price of the service, while at the same time not making optimal service classes so expensive that even performance-sensitive users do not use them? If we cannot answer this question in the affirmative, then much of the technical work on multiclass service disciplines will be rendered ineffective because users will not employ the service classes in an appropriate way. The above question is posed in rather imprecise terms. The challenge, now, is to formulate this notion precisely. This is the task to which we turn in the next section.

3 Abstract Formulation

In this section, we present an abstract formulation of a service discipline and a pricing policy. With this formulation we are able to state an optimality criterion for service disciplines and then, for a given service discipline, an acceptability criterion for a pricing policy. Throughout this section, we consider a fixed network with a fixed set of n potential network users (who may or may not decide to use the network). The formulation borrows heavily from the field of economics, particularly in the use of utility functions, Nash equilibria, and Nash implementation (see References [7, 11, 15]).

Let s_i denote a characterization of the network service received by the i'th user (which may be no service at all). Let $V_i(s_i)$ denote the i'th user's level of satisfaction with a given network service s_i. The functional dependence of V_i on s_i reflects the particular nature of the application being run by user i. The unit of V_i is money (say, U. S. dollars); V_i reflects, up to an arbitrary additive constant, how much money user i would be willing to pay for a particular level of service. Thus, if a user is charged an amount c_i for that service, her overall level of satisfaction (up to the same arbitrary additive constant), which we denote by U_i, becomes: $U_i = V_i(s_i) - c_i$[17].

In our model of the service discipline, each user sends to the network a signal, or request, σ_i which lies in some space $\mathcal{S}$ of possible service requests. This term *request* should not be interpreted as necessarily involving an explicit call set-up; in this abstract formulation the signal could be anything, including the unreserved transmission of packets. The resulting network service s_i received by each user is a function of the vector of requests: $s_i(\vec{\sigma})$. The signal space $\mathcal{S}$ and the function $s_i(\vec{\sigma})$ completely specify the network's service discipline.

Let us denote by $\vec{\sigma}^{max}$ the vector of signals[18] that maximizes the sum of the $V's$:

$$\sum_{i=1}^{n} V_i(s_i(\vec{\sigma}^{max})) \geq \sum_{i=1}^{n} V_i(s_i(\vec{\sigma})) \quad for\ all \quad \vec{\sigma} \in \mathcal{S}^n$$

Let V^{max} denote the maximal sum: $V^{max} = \sum_{i=1}^{n} V_i(s_i(\vec{\sigma}^{max}))$. The network's resources are used

bandwidth and switching) will be so plentiful that congestion will rarely occur and so usage-based pricing is unnecessary. We do not believe that bandwidth will stay ahead of demand without some form of usage-based pricing, but there is no way to settle this argument definitively.

[17] We should note that this modeling choice contains the technical assumption that the preferences are quasilinear.

[18] First, we assume that such a maximum exists. This would follow immediately if the space $\mathcal{S}$ were finite, or if the mappings $V_i(s_i(\vec{\sigma}))$ were continuous and the space $\mathcal{S}$ were compact. Also, there may be several such maximizing vectors. For convenience, in what follows we assume that there is only one.

most efficiently, i.e. produce the highest total satisfaction, when the users send the request vector $\vec{\sigma}^{max}$. Notice that here efficiency does not refer to any network-oriented measure (such as link utilization) but rather refers *only* to the aggregate level of user satisfaction that the network delivers. For a given network configuration, one service discipline is deemed superior to (i.e., more efficient than) another if it produces a larger value for V^{max}. For a given network configuration, an optimal service discipline is one which is not inferior to any other service discipline. We expect that multiclass service disciplines will have higher values for V^{max} than single-class service disciplines in most network configurations; this is merely a precise formulation of the observation in Section 2 that multiclass service disciplines could meet users' needs more efficiently than single-class service disciplines. Note that this condition does not say anything about the distribution of the total utility; the maximal value might be achieved when some users have very high utilities and others have very low utilities.

In the preceding paragraph we defined a service discipline as a function that takes a vector of requests $\vec{\sigma}$ as input and then assigns network service $s_i(\vec{\sigma})$ to each user. Similarly, a pricing policy (or pricing scheme) is a function that takes a vector of requests $\vec{\sigma}$ as input and then assigns costs (i.e., the fee charged for the service) $c_i(\vec{\sigma})$ to each user. Given a pricing scheme and a service discipline we can consider U_i to be a function of $\vec{\sigma}$: $U_i(\vec{\sigma}) = V_i(s_i(\vec{\sigma})) - c_i(\vec{\sigma})$.

Our assumption is that each user acts selfishly, and will request the service that maximizes her individual level of satisfaction. This assumption implies that the resulting request vector $\vec{\sigma}$ will have the property that no user, by unilaterally changing her own request σ_i, can increase her utility. In economics this is referred to as a *Nash equilibrium* (see References [7, 15] for basic definitions, and Reference [11] for a more comprehensive treatment). More formally, $\vec{\sigma}$ is a Nash equilibrium if for all i and all $\tilde{\sigma} \in \mathcal{S}$ we have $U_i(\vec{\sigma}) \geq U_i(\vec{\sigma}|^i\tilde{\sigma})$. Here we use the notation that $\vec{\sigma}|^i\tilde{\sigma}$ denotes the vector with the i'th component given by $\tilde{\sigma}$ and all other components j given by σ_j, i.e. the input vectors $\vec{\sigma}$ and $\vec{\sigma}|^i\tilde{\sigma}$ differ only in the i'th component.

Our goal is to have the network operate at peak efficiency, which happens only when the vector of requests is $\vec{\sigma}^{max}$. Given our assumption that the selfish behavior of individual users will result in $\vec{\sigma}$ being a Nash equilibrium, our goal of network efficiency requires that $\vec{\sigma}^{max}$ be a Nash equilibrium. Unfortunately, in the absence of a pricing scheme this is rarely the case. Recall that the criterion defining $\vec{\sigma}^{max}$ refers only to the sum; any individual user might receive a very low value for $V_i(s_i(\vec{\sigma}^{max}))$. Furthermore, without a pricing scheme (i.e., $c_i(\vec{\sigma}) = 0$ for all i and all $\vec{\sigma}$), the Nash equilibrium condition requires that $V_i(s_i(\vec{\sigma})) - V_i(s_i(\vec{\sigma}|^i\tilde{\sigma})) \geq 0$. This condition is unlikely to be met unless every user is requesting the highest quality service; however, the maximizing request vector $\vec{\sigma}^{max}$ is unlikely to be one in which every user is requesting the highest quality service. Thus, without a pricing policy it is improbable that the network resources will be used efficiently.

When we consider a nontrivial pricing policy, any Nash equilibrium $\vec{\sigma}$ obeys the following relation: $V_i(s_i(\vec{\sigma})) - V_i(s_i(\vec{\sigma}|^i\tilde{\sigma})) \geq c_i(\vec{\sigma}) - c_i(\vec{\sigma}|^i\tilde{\sigma})$. Thus, at a Nash equilibrium, any improvement in service quality obtained by submitting a different request is offset by the resulting increased cost; similarly, any decrease in cost obtained by submitting a different request is offset by the resulting decrease in service quality.

For a given network configuration, we say that a pricing scheme is *acceptable* if the unique Nash equilibrium for that pricing scheme is $\vec{\sigma}^{max}$. An acceptable pricing scheme is one in which the selfish behavior of users results in an efficient usage of the network's resources. In economics, this acceptability condition is called Nash implementation of maximal efficiency (see Reference [11] for a general description of Nash implementation of social choice functions; maximal efficiency is merely a particular example of a social choice function).

If there is a current status quo pricing scheme, we say that a new pricing scheme is *adoptable* if all users are at least as satisfied (and at least one user strictly more satisfied) with the new scheme[19]. This addresses the political process of adopting a new pricing policy. If no one would lose by the adoption of a new pricing policy, it is less likely to run into political opposition.

This abstract formulation allows us to precisely define the role of service disciplines and pricing policies. The role of an optimal service discipline is to maximize V^{max}, regardless of the resulting user incentives. The role of an acceptable pricing policy is to ensure that user self-interest will lead to the users choosing $\vec{\sigma}^{max}$ and thus will maximize the efficiency of the network. Furthermore, the role of an acceptable and adoptable pricing policy is to make every user at least as satisfied with the new pricing policy, so that the benefits of the increased efficiency are spread around to all users.

For any given service discipline $\vec{s}(\vec{\sigma})$ and set of V_i's, we can always find a set of charges $c_i(\vec{\sigma})$ that is both acceptable and adoptable. However, in practice, there are often restrictions on the network pricing structure. For instance, one might only be able to charge based on the signal σ_i (i.e., my cost is independent of what service other people request), and this cost (at least per unit) must be the same for all users (i.e., the network cannot discriminate between users based on their V_i's). Given such restrictions, it is not clear that one can always find a set of charges $c_i(\vec{\sigma})$ which is both acceptable and adoptable; we discuss this issue, and its relation to the externality and priority pricing literatures, in Section 5. In the next section, we will explore the existence of acceptable and adoptable pricing schemes in a simple example.

4 Example

We now have an abstract formulation of the interaction between pricing policies and service disciplines in computer networks. However, the most pressing challenge currently facing network designers is to develop the optimal service disciplines (that is, the ones which have the maximal V^{max}'s). While much basic research has been done in this area, no consensus has yet been reached, by either the marketplace or the research community, on the nature of these new service disciplines. Only when these new service disciplines are identified will it be possible to develop the associated acceptable and adoptable pricing policies. In the meantime, we present a qualitative framework for future pricing design. Our goal in this section is to illustrate some of the basic ideas of such pricing policies through the simulation of a simple example.

The example utilizes an extremely simple multiclass service discipline, one that involves only two service classes, so that the pricing issues are not obscured by the technical details of the service discipline; obviously, we expect that future networks will have substantially more complicated multiclass service disciplines. However, the network context is realistic, in that we consider a

[19] In standard economic terminology, a pricing scheme is adoptable if it produces an allocation which is Pareto superior to the status quo allocation. Thus, adoptability is merely an individual rationality condition relative to the status quo allocation.

standard TCP/IP ([20]) network and our simulations are done at the packet level. We focus on several simple network configurations (where a network configuration defines both the network properties and the user population). Each network user is represented by an instance of one of four different application types: electronic mail, real-time packetized voice, a file transfer service, and a remote login service. Associated with each application type is a function $V_{application}(s)$ which describes how the perceived performance of the application depends on the network service. Also, we associate with each application a particular model of traffic generation (specified by several user-specific parameters).

We outline the details of our example in the following subsections. We first describe the service discipline and pricing policies, and then discuss how the abstract formulation should be applied to this example. We next define the $V_{application}$'s for the various applications, and then, after reviewing some miscellaneous details of the simulation, describe the various network configurations considered. We conclude our treatment of this example with a presentation and discussion of the simulation results.

4.1 Service Discipline and Pricing Policies

At an abstract level and ignoring routing, there are only two decisions faced by a switch. When the transmission line is free and there are packets in the queue, the switch must select the next packet to transmit. When a packet arrives and there is no room in the queue, the switch must decide which packet to discard. The traditional FIFO service discipline, which provides only a single service class, is to queue the packets in order of increasing time-of-arrival and then transmit the packet at the head of the queue and, when necessary, discard the packet at the tail of the queue. In our example, we use the simplest multiclass service discipline, which is merely to extend the FIFO service to have two service classes: high priority and low priority. The switch then (logically) keeps a queue with the high priority packets arranged in increasing time-of-arrival, followed by the low priority packets arranged in increasing time-of-arrival. The switch transmits the packet at the head of the queue and, when necessary, discards the packet at the tail of the queue. We consider two different pricing schemes. The first is a flat per-byte price applied to all packets traversing a link, call it p_{flat}. In the second pricing scheme, called priority pricing (based on a theoretical analysis of a similar problem in [26]), we charge different per-byte prices for the two priority classes. Let us denote these two per-byte prices by p_{high} and p_{low}; clearly we should set $p_{high} \geq p_{low}$ so that the monetary incentives encourage the use of the low priority service class.

In order to facilitate direct comparison between the two pricing schemes in the simulation study presented below, we require that, in equilibrium, both pricing schemes recover the same net revenue $R = \sum_{i=1}^{n} c_i(\vec{\sigma})$. This means choosing p_{flat} and p_{high} and p_{low} so that the total revenue is equal to R, and thus the absolute values of the prices in the two schemes will depend on the offered load.

4.2 Definition of $V_{application}$'s

To study user satisfaction in a network with numerous traffic sources and distinct types of service, we have chosen a set of applications with diverse service requirements. The following applications will be considered in this study: electronic mail (Email), a file transfer service (FTP), a remote login service (Telnet), and real-time packetized voice (Voice). We discuss each one of them in turn, giving a general description and then presenting our performance evaluation criteria, $V_{application}$, that roughly models the application's service requirements. These functions oversimplify the true relationship between application performance and network service, but our purpose is only to capture the essence of the relationship. We defer the description of the detailed traffic generation characteristics of these applications to Section 4.4. However, it is relevant to our current discussion to note that each of the applications is invoked many times during the period of evaluation. Accordingly, the user's perception of the application performance reflects the average performance throughout this interval; thus, each $V_{application}$ is a function of average quantities, such as average delay. These average quantities are the service descriptors s_i used in the abstract formulation. Also, recall that $V_{application}$ is defined up to an arbitrary additive constant. We have chosen, for three of our applications (Email, Telnet, and Voice), to have $V_{application} < 0$ and let $\|V_{application}\|$ reflect the level of perceived performance degradation of the application. For the fourth application (FTP), we have $V_{application} > 0$ and let $\|V_{application}\|$ reflect the level of satisfaction. We should also note that the constants appearing in the definitions for $V_{application}$ are chosen so that the dynamic ranges have similar magnitudes.

Email is used for multi-user, asynchronous communication. Since instantaneous delivery is not expected, we assume users care mostly about their mail arriving within some delay on the scale of minutes. For messages delivered within this bound, we assume the user has only a slight preference for reduced delays. We model these service requirements through:

$$\begin{aligned} V_{Email} = & -0.1(\text{average message delay, in seconds}) \\ & -(\text{\% of messages not delivered in loose delay bound of five minutes}) \end{aligned}$$

In our model, FTP is a single user pseudo-interactive application. Since FTP users often await completion of the application before proceeding with other tasks, we assume they want the network to deliver relatively prompt service. This expectation must be tempered by the physical limits of the underlying network. The ideal transfer time of a file is the time it would take the file to be transferred if there was no other traffic on the network (and the flow control algorithm in the underlying transport layer allowed for full utilization of the available bandwidth). Defining the normalized throughput of a particular file transfer to be the ratio of the ideal transfer time to the actual transfer time, we model the FTP service requirements by:

$$V_{FTP} = 100(\text{average normalized throughput})$$

Telnet exemplifies a truly interactive application. A user conducts a Telnet session as a primary task and expects real time responses. We assume Telnet users are sensitive to packet delays that exceed a few hundredths of a second. Since remote echoing is used in most Telnet connections, the relevant delay is the roundtrip time from the transmission of a packet to its acknowledgment. These requirements are expressed by:

$$V_{Telnet} = -(\text{average packet round trip time, in milliseconds})/10$$

Voice is a real time application which is extremely sensitive to delay; voice applications cannot tolerate absolute delays much above 100 milliseconds [5][20]. The improvement in performance when the delays are reduced well below this limit is slight. At the same time, voice differs from the other applications considered in that the requirement for 100% reliability is removed. Human speech includes enough redundancy to allow correct interpretation in the presence of some data loss. This allows the voice application to trade reliability for reduced delivery delay when it cannot get both. We consider a model of a voice application in which packets that do not make a tight delay bound (set at 100ms) are discarded; from the perspective of the application dropped packets are no different from delayed ones. Letting d denote the average one-way delay of the voice packets (measured in ms), our assumed application performance function is:

$$V_{Voice} = -(\% \text{ of packets not obeying the tight delay bound of 100ms}) - \mathrm{d}/100$$

4.3 Applying the Abstract Formulation

We have not attempted to capture overall demand elasticity in this model, i.e. where users could choose to adjust their traffic generation pattern (or even cease to use the network) if the prices were too high[21]. This enhancement to the model is discussed in Section 5 where we discuss related work, and is also covered, in a somewhat different context, in Reference [17]. In our current study we make the simplifying assumption that the total traffic generated by a user is independent of the price; *we model only the users' service class decisions.* Therefore, in this example, the *requests* σ_i are merely the priority settings on the packets. The only action we allow users is to select their priority settings on their packets. The packet generation pattern is a function of the application type (defined in more detail in Section 4.4) and is independent of this priority selection.

In the abstract formulation, each user could individually choose their request σ_i. However, in our example we have chosen to have each instance of a particular application (Email, Voice, Telnet, and FTP) use the same priority settings. We make this choice for two reasons. First, we assume that it will be the application that invokes the underlying network transport protocols and thus determines the priority settings; the typical user, we assume, will not be intervening manually in this process. Second, two users of the same application will want to set their priority levels differently only if they know that the network conditions along their respective transmission paths are different. We assume that the underlying network will be relatively invisible to the typical user, so that this information will not be available[22].

This modeling choice implies that in applying the abstract formulation to this model, we should assume that it is the application type that is choosing the request σ_i and that the V of the application type is the sum of the V_i's of the users of that application. The Nash equilibrium conditions, as well as the acceptability and adoptability conditions, must be modified in the same way.

Under the flat pricing scheme, it is clear that each application type will choose to request high priority service. This is because there is no monetary incentive to request low priority service and,

[20] Voice is also sensitive to the variance in delays, often referred to as jitter, but we do not model that dependence here.

[21] Such a model would have to include in V_i the dependence of user satisfaction on the traffic generated.

[22] This assumption, while widely held, is a debatable normative statement about future networks.

if there is any congestion in the network, there is a performance incentive to request high priority service. We denote by V^{flat} the value of $\sum_{i=1}^{n} V_i$ when all applications request high priority service.

Under the priority pricing scheme, the situation is less obvious. There is a performance incentive to request high priority service and a monetary incentive to request low priority service. The service request that maximizes a user's utility will depend on the values of p_{high} and p_{low}, as well as on the traffic load in the network. Thus, the Nash equilibrium priority requests will depend in detail on the network configuration.

Our objective is now to find, for each network configuration, prices p_{high} and p_{low} such that priority pricing is acceptable, i.e. that the selfish Nash equilibrium results in priority settings for each application type that maximizes the overall level of satisfaction. Furthermore, we want to find adoptable priority prices, in that every application type is at least as well off with priority prices as they were under the flat pricing scheme.

4.4 Simulation Details

The applications are built upon two transport protocols. Email, FTP, and Telnet use TCP [20] whereas Voice uses UDP [19]. UDP was chosen for Voice because, given the strict delay constraints of that application, retransmissions of dropped packets are not useful.

As mentioned in Section 4.2, each user repeatedly requests service from her application, and the application's performance is averaged over all such instances. Each request can be characterized by a size, s, and the time interval, t, from the last invocation of the application. We have modeled this user behavior by a random process, with both the request size and the time interval being exponentially distributed random variables with means $\bar{s}$ and $\bar{t}$ respectively.

For Email and FTP, the size of a request refers to the size of the message or file to be transmitted. These messages and files are transmitted using a maximum packet size of 500 bytes. For Telnet, the size of a request is the number of characters generated in a burst; each character is transmitted and echoed separately, using 50 byte packets[23]. A voice request is a conversation; the size of the request is the duration of the conversation. Each conversation is made up of several talk spurts; during a talk spurt, 180 byte packets are transmitted in packet trains supporting a 64 Kbps data rate. The total simulation time was 90 minutes per run, not including an initial warmup period of two minutes.

As stated before, when comparing various pricing schemes, we hold fixed the total revenue. For this example, we set the total revenue R to be 100.

4.5 Network Configurations

We study the interaction between the service discipline and the pricing scheme on seven simple network configurations. A configuration is specified by the network topology and the user population. The seven configurations we studied were based on three different network topologies, which

[23]We model only the traffic generated by the Telnet client.

are labeled A, B, and C and are depicted in Figures 1-3. The figures depict the transmission lines, as well as the location of the various switches and network hosts. The links that connect two switches (as opposed to connecting a host to a switch) are called *backbone* links and are depicted in the figures with thick lines.

The user population of the seven configurations investigated in our simulation study are described in Tables 1-7. The final column of these tables lists the source host and destination host (labeled by the host number depicted on the associated network topology diagram) of each instance of a given application type and given traffic generation parameter description $(\bar{s}, \bar{t})$. Furthermore, Table 8 describes the relative composition of the traffic load in the seven configurations, listing the percentages of the number of packets generated and the number of bytes generated by each application type.

All links which connect two switches in a given network topology have the same speed, which we refer to as the backbone speed, and have a constant latency of 10msec. All links connecting hosts to switches are always 10Mbps regardless of the backbone speed, and have a latency of 1msec. For each network configuration, we performed simulations with six different backbone speeds.

The seven configurations were chosen to represent a wide range of topologies and traffic patterns. The first configuration was designed to provide some baseline data on a very simple network topology and traffic pattern. Configuration 1 uses network topology A, which has only a single backbone link, and has only one-way traffic (i.e., all data packets traveled in the same direction on the backbone link). Configurations 2 and 3 also use the same simple network topology A and have only one-way traffic, but have different application mixtures than Configuration 1; Configuration 2 has more Email traffic and Configuration 3 has more Voice traffic. Configuration 4 uses the same network topology A, but has two-way traffic (i.e., there were data packets traveling in both directions on the backbone link[24]). Configuration 5 uses the slightly more complicated network topology B which has two backbone links; the traffic load has only one-way traffic. Configurations 6 and 7 use the much more complicated network topology C, which has seven backbone links. Configuration 6 has only one-way traffic, and Configuration 7 has two-way traffic. Thus, these seven configurations allow us to explore the effects of different application mixtures, one-way versus two-way traffic, and varied network topologies.

4.6 Results

Tables 9-15 contain the simulation results for the various configurations. For each configuration, we studied six different backbone speeds (recall that in the multi-link networks, all links connecting two switches have the same speed). For each configuration and backbone speed, the Tables 9-15 contain the following information: average link utilization, the maximizing priority settings $\vec{\sigma}^{max}$ (displayed as a string of digits $\sigma_{Telnet}, \sigma_{Voice}, \sigma_{FTP}, \sigma_{Email}$ where 2 indicates high priority and 1 indicates low priority), V^{max}, V^{flat}, and three price ranges. Note that given a priority setting and p_{high}, the value of p_{low} is determined by the constant revenue condition and the offered load (which is independent of the priority settings). Thus, a pricing scheme for a particular configuration is completely specified by a priority setting and a value for p_{high}. The first price range is the feasible

[24] Reference [28] observes that the network dynamics with two-way traffic can be very different than the dynamics with only one-way traffic.

set of values for p_{high} such that at the maximizing priority setting we have $0 \leq p_{low} \leq p_{high}$. At the low end of this range for p_{high}, we have $p_{high} = p_{low}$ and thus there is no monetary incentive to choose low priority service; at the high end of this range we have $p_{low} = 0$ and all of the revenue is generated by the high priority service class. The second price range is the set of values for p_{high} such that the pricing scheme is acceptable; for each price in this range, the maximizing priority vector $\vec{\sigma}^{max}$ (which is listed in the third column) is a Nash equilibrium. The third price range is the set of values for p_{high} such that the pricing scheme is adoptable; for each price in this range, all application types have values of U under the priority pricing scheme that are higher than their values of U under the flat pricing scheme.

Our objective was to find values for p_{high} and p_{low} such that the priority pricing scheme was both acceptable and adoptable. The most important result of our simulations is that, for each configuration and at each backbone speed (making a total of 42 separate instances), there is a set of prices that is both acceptable and adoptable. Thus, in each of these configurations, we can always find values for p_{high} and p_{low} such that all users are more satisfied with the priority pricing than with the flat pricing scheme. Furthermore, this priority pricing scheme will induce each user to, out of her own self-interest, ask for the service class that maximizes the network's overall efficiency. The existence of acceptable and adoptable priority pricing schemes over a wide range of network conditions is encouraging evidence that pricing will indeed be an effective method of inducing overall network efficiency.

There are also several other interesting aspects to the data. As expected, the overall level of satisfaction is always higher under the maximizing priority setting than it is when all applications request high priority service (which occurs when the prices are flat). The difference between V^{max} and V^{flat} decreases as the backbone speed increases (which, because the offered load is kept constant, reduces the utilization) because the scheduling algorithm becomes less important as the network becomes less congested. Thus, the use of a pricing scheme to maximize network efficiency is less important when the network is lightly loaded.

A related effect is that the range of acceptable prices increases as the utilization increases. When the network is heavily loaded, almost any pricing scheme that charges more for the high priority service will induce applications to make the appropriate service class decisions. When the network is only lightly loaded, the range of acceptable prices is often extremely small. However, since the overall efficiency is relatively independent of the priority choices in this case (as indicated by the closeness of V^{max} and V^{flat}), there is no significant efficiency loss if the prices are not within this range.

One would expect that increasing the backbone speed would always increase the overall level of satisfaction. However, notice that in each one of our configurations this expectation is violated; as the backbone speed increases, V^{max} first increases as expected but then decreases rather dramatically. This surprising decrease in V^{max} is due to dynamics of the window-based flow control used in TCP, and has nothing to do with the issues addressed in this paper. We clarify this phenomenon as follows. Recall that V_{FTP} compares the actual transfer time to the ideal transfer time, and this ideal transfer time is computed assuming that the flow control algorithm allows the source to fully utilize the available bandwidth. When the backbone speeds are low, the transfer time of an FTP application is limited by congestion in the backbone links; at these low speeds, any increase in the backbone speed decreases the congestion and therefore increases V_{FTP} and thus increases V^{max}.

At high backbone speeds and the same offered load, there is little congestion (as indicated by the utilization data) and the achieved throughput rate of an FTP application is limited only by the available bandwidth and the flow control algorithm's ability to utilize that bandwidth. The window flow control used in TCP, which in our simulations has a maximum window size of 5000 bytes, results in the transfer time reaching a minimum value that is essentially independent of any further increases in the backbone speed. Thus, any further increase in the backbone speed merely decreases the ideal transfer time but does not change the actual transfer time, thereby decreasing V_{FTP} and thus decreasing V^{max}.

The Telnet and Voice applications are both real-time applications, with performance sensitivities to delays of tenths of seconds. The Email and FTP are much less sensitive to delays. Thus, one might have predicted that the maximizing priority setting would always be when the Telnet and Voice applications request high priority service and the FTP and Email applications request low priority service; in our notation, this is the 2211 priority setting. At the slowest backbone speeds (the highest network utilizations), this is indeed the maximizing priority setting in all of our network configurations. However, as the backbone speed is increased, the maximizing priority setting changes from 2211 to 2221 in every configuration. At the setting 2221, the FTP application is requesting high priority service along with the Voice and Telnet applications; this priority setting is not maximizing at the slowest backbone speeds because, when the utilization is extremely high, the Telnet and Voice applications cannot tolerate additional high priority traffic without suffering significant performance degradation. In six of the seven configurations, at the highest backbone speed the maximizing priority setting had Voice requesting low priority service[25]; this is because the utilization was low enough that no packets violated the Voice delay bound. Also, note that in no case was it maximizing for Email to request high priority service; this reflects the relative delay-insensitivity of this application.

The data shows that the range of acceptable and adoptable priority prices depends on the network topology, the backbone speeds, and the offered load. In order for our paradigm, in which prices are used to elicit efficient network utilization, to be applicable to real computer networks, we must assume that the administrative body that sets the prices can make, on the basis of historical evidence (and perhaps market research), sufficiently accurate predictions of the offered load. There is a precedent for this; in the telephone network, the offered load is reasonably well characterized (see [14]). We expect that in the future Internet, the aggregate traffic load will be so large that a statistical characterization will yield reasonably accurate predictions. Of course, these load patterns will vary depending on time-of-day and other factors, and so the pricing scheme will take those factors into account.

5 Related Work and Relevant Issues

This paper represents an initial attempt to study pricing policies in multiple service class computer networks. We first presented a framework for understanding the interaction between pricing and service disciplines, and identified two normative criteria for pricing schemes: acceptability and adoptability. We then described a simple model in which to explore this interaction. Our

[25]We assume that this would also have occurred in the remaining configuration (configuration 2) if we had tested a higher backbone speed.

model combines several disparate issues, such as service disciplines, application performance, user behavior, congestion externalities, and incentives. In order to make the model tractable we have, in each case, considered these issues in an oversimplified way. In the course of reviewing the related work, we now revisit some of these issues and discuss the limitations of our current model.

There is a large and rapidly growing body of work on the design of multiclass service disciplines for high-speed data networks that support a wide range of service requirements (see, for example, References [1, 6, 27]). There are many different approaches to meeting these service requirements, and they differ in some profound ways (such as reservation vs. best-effort) that will have significant implications for network performance and the associated user incentives. In particular, admission control would allow the network to prevent users from using the network if the traffic load is too high; the network can then choose to deny service to some users rather than have the overall level of service delivered to all users reach unacceptable levels. Also, using service disciplines which provide service guarantees (as in Reference [1]) would mean that for certain service classes the quality of service delivered to a user is independent of the traffic conditions.

In this paper we have chosen to consider only the simplest form of a multiclass service discipline, retaining the best-effort paradigm of the current Internet protocols and merely adding two priority classes. We do not expect that this simple approach could, in reality, support a sufficiently wide class of service requirements. However, we do expect that the fundamental pricing issues raised by this simple approach will resemble those of more complicated service disciplines.

There is a surprisingly small literature on the relationship between network performance and application performance. Those studies that have been done have focused on voice [18]. The other application performance measures $V_{application}$ we described are probably indicative of the primitive state-of-the-art in this regard. One of the points of our work is to apply these V's to network performance under different loads and service disciplines in order to measure relative user satisfaction. Of course, there is no fundamental way to compare the V's of different applications, and so we have made rather arbitrary choices about the relative scales of the various V's. Our results are somewhat robust against changes in these modeling choices, in that we were still able to find acceptable and adoptable pricing schemes with different definitions of the V's.

A central assumption in this paper is that users respond according to the incentives they face. We expect that in the future Internet, with its large user population, users will not restrict their offered load, or the service class requested, without some incentive to do so. However, we considered a very simple set of responses in this first modeling effort, addressing only the issue of service class requests. We did not consider a broader set of user actions such as demand elasticity, in which users reduce offered load in response to increased price, and substitution, in which users switch from one application to another (e.g., using Email instead of voice). Such critical extensions to our work will require a much more detailed model of user preferences. We should also note that in considering a fixed load we totally ignored the natural variations in load that occur (even without incentives). Obviously, any real pricing scheme must take time-of-day and other factors into account.

A critical feature of our problem, which renders it nontrivial, is that agents affect each other by their actions. The effect that one user has on another in a network can be modeled in the traditional economics framework as an externality. There is a well developed theory of externalities in the economics literature (see, for example, [7] or [8]). Modulo various mathematical technicalities (convexity conditions, etc.), if one charges each user an individualized price for their consumption one can find prices such that the Nash equilibrium yields the efficient allocation; these prices are just set equal to the total marginal disutility of all other agents (which, for an efficient allocation, is exactly equal to the marginal utility of the consuming agent). This result depends crucially on the prices being different for each user since the externality they cause is different[26].

In our formulation, we have assumed that the price-per-unit of a certain service signal is uniform across all agents[27]. Thus, we do not have the personalized prices required to invoke the standard externality result. In fact, our model is more specifically related to the priority pricing literature, as represented by [26], than to the standard externality literature. This literature considers a simple model of allocation of a nonstorable good with fluctuating supply and constant demand; Reference [26] shows that there are certain priority pricing schemes that can make everyone better off, when compared to a flat pricing scheme. Our purpose, in this paper, was to apply this insight to a realistic network setting. The model considered in [26] is inherently one-dimensional (the quality of service delivered can be described by a single variable). The network problem we consider is many-dimensional, since quality of service involves many parameters. A related theoretical question is whether or not the rigorous results of priority pricing can be generalized to the multidimensional case of sophisticated network service models. Thus, while the results from our simple example are similar to, and inspired by, those in [26], they are not implied by them.

There is also a sizable literature on congestible facilities. The standard models, as represented by [25], invoke *ad hoc* formulae to describe the effect of congestion on users. The structural models, as represented by [3], derive these congestion effects directly from the dynamics of the congested facility itself. In that sense, this structural approach is similar to ours; however, these structural models of congestion, with their emphasis on peak and off-peak usage, are rather different from the network model we have considered here.

In the future, network designs must provide adequate mechanisms to implement these monetary and performance incentives. Recent discussion of resource usage feedback mechanisms has identified several somewhat independent design choices such as feedback channel (e.g., monetary, administrative), feedback policy (e.g., based on quality of service delivered or priority level requested), and granularity (e.g., charge back to end users or to institutions for aggregate traffic) [4]. We have not addressed the many implementation and protocol support issues that arise in the context of any usage sensitive pricing systems. In particular, we have not dealt with the issues of accounting and authentication, which must be confronted before pricing can become a reality.

There are other approaches to studying incentive issues in networks. In particular, user responses to incentives in computer networks have also been treated in the game theory literature. While only simple queueing network models are considered (see [13, 21] and references therein), the incentive issues addressed are quite similar to those discussed in this paper.

Reference [17] also addresses the issue of pricing in computer networks, but its focus is rather different from that of this paper. First, Reference [17] considers a reservation-oriented service

[26]There are some specialized situations, such as when the externality is homogeneous, where personalized prices are not needed.

[27]We consider this to be a realistic modelling choice, in that network operators will likely either be forbidden to charge personalized prices by common carriage laws, or will find such personalized prices to be impractical due to the impossibility of accurately identifying the externality effects.

discipline as opposed to a best-effort service discipline; congestion therefore effects only the rate of call blocking and does not effect the quality of service delivered. Second, Reference [17] does not explicitly deal with user incentives but rather considers the constraints imposed by finite budgets (in that each user has a finite budget and will purchase the maximal amount of network service they can afford).

References [9, 10] provide a good complement to this paper. They discuss the economics of the Internet, giving data on the cost structure and trends. They also propose a pricing scheme for the Internet which involves "bidding" on each packet; while this proposal does not address multiple service classes, it does use monetary incentives to control network use. Another related work is Reference [24] which discusses various economic issues in the pricing of broadband services.

In this paper, we have focused on a very narrow question: how does pricing interact with multiple qualities of service? While we did not discuss them here, we should note that there are many other issues that are relevant to pricing in computer networks. The nature of network demand and the the relative supply of bandwidth will play a crucial role in determining the pricing structure. Also, different forms of network suppliers (such as cable TV, wireless networks, telephone networks, etc.) will have very different service offerings and very different cost structures; this too will have an impact on pricing. Finally, the regulatory framework of the future telecommunications infrastructure, and in particular the issue of common carriage (see [16]), will have a major impact on pricing. However, the basic point we have focused on here remains valid in the presence of these other considerations; multiple service classes will be an effective way of building efficient networks only to the extent that they are supported by appropriate pricing policies.

6 Conclusion

In this paper we have studied the role of pricing policies in multiple service class networks. We have argued that for *any* multiclass service discipline to have the desired effect of maximizing network performance, some form of service-class sensitive pricing is required. We then presented an abstract formulation of service disciplines and pricing policies. This formulation allowed us to more clearly describe the interplay between service disciplines and pricing policies. Effective multiclass service disciplines allow networks to focus resources on the performance sensitive applications, while effective pricing policies allow us to spread the benefits of multiple service classes around to all users, rather than just having these benefits remain exclusively with the users of applications which are performance sensitive. Finally, we illustrated some of these concepts through simulation of several simple example networks. In our simulations, we found that it is possible to set the prices so that users of every application type are more satisfied with the combined cost and performance of a network with service-class sensitive prices. For some application types the performance penalty received for requesting a less-than-optimal service class is offset by the reduced price of the service. For the other application types the monetary penalty incurred by using the more expensive, higher quality service classes is offset by the improved performance they receive.

On one level our conclusions are hardly surprising. Offering multiple service classes and charging differently for them is an obvious idea, and it is certainly not new with us. However, it is a crucial idea that needs to be more fully explored. We expect that with service-class insensitive pricing, user behavior will render the network equivalent to a single service class network. We then think one of two outcomes is likely. One possibility is that the quality of service will be quite low, thereby limiting the ability of the network to support demanding applications like real-time video or voice; in this case, the only viable applications will be like those on today's Internet. The other likely outcome is that, by over-engineering the network, the quality of service will be quite high, but so will the costs, and only the most quality conscious users will consider the cost worthwhile. In both cases, the technical achievement of integrating applications with different qualities of service requirements in one network may be undone by the economic forces that segment the market.

References

[1] David D. Clark, Scott Shenker, and Lixia Zhang. Support for real-time applications in an integrated services packet network: architecture and mechanism. In *Proceedings of SIGCOMM '92*, September 1992.

[2] Ron Cocchi, Deborah Estrin, Scott Shenker, and Lixia Zhang. A study of priority pricing in multiclass networks. In *Proceedings of SIGCOMM '91*, September 1991.

[3] André De Palma and Richard Arnott. The temporal use of a telephone line. *Information Economics and Policy*, 4:155–174, 1989.

[4] Deborah Estrin and Lixia Zhang. Design considerations for usage accounting and feedback in internetworks. *ACM Computer Communication Review*, 20(5):56–66, October 1990.

[5] Domenico Ferrari. Client requirements for real-time communication services. *Communications Magazine*, pages 65–72, November 1990.

[6] Domenico Ferrari and Dinesh C. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[7] David Kreps. *A Course in Microeconomic Theory*, Princeton University Press, 1990.

[8] Jean-Jacques Laffont. *Fundamentals of Public Economics*, The MIT Press, 1988.

[9] Jeffrey MacKie-Mason and Hal Varian. Some economics of the internet forthcoming in *Networks, Infrastructure and the New Task of Regulation*, Harvard University, 1993.

[10] Jeffrey MacKie-Mason and Hal Varian. Pricing the internet to appear in *Proceedings of "Public Access to the Internet"*, Harvard University, 1993.

[11] Eric Maskin Theory of implementation in nash equilibrium. In *Social Goals and Social Organization*, edited by Hurwicz, Schmeidler, and Sonnenschein, pages 173-204, 1985.

[12] Richard McLean and William Sharkey. An approach to the pricing of broadband telecommunication services, preprint, 1993.

[13] Haim Mendelson and Seng-Jin Whang. Optimal Incentive-Sompatible Priority Pricing for the M/M/1 Queue. *Operations Research*, 38:870–883, 1990.

[14] Bridger Mitchell and Ingo Vogelsang. *Telecommunications Pricing*, Cambridge University Press, 1991.

[15] Walter Nicholson. *Microeconomic Theory Basic Principles and Extensions*, Dryden Press, 1989.

[16] Eli Noam. The impending doom of common carriage, preprint, 1993.

[17] Colin Parris, Srinivasan Keshav, and Domenico Ferrari. A framework for the study of pricing in integrated networks. Technical Report TR-92-016, International Computer Science Institute, Berkeley, March 1992.

[18] David Petr, Luiz DaSiliva, and Victor Frost. Priority discarding of speech in integrated packet network. *IEEE Journal on Selected Areas in Communications*, 7(5):644–656, June 1989.

[19] Jon Postel. User datagram protocol. Request For Comments 768, Information Sciences Institute, University of Southern California, August 1980.

[20] Jon Postel. Transmission control protocol. Request For Comments 793, Information Sciences Institute, University of Southern California, September 1981.

[21] Scott Shenker. Efficient network allocation with selfish users. In P. J. B. King, I. Mitrani, and R. J. Poole, editors, *Proceedings of Performance '90*, pages 279–285, September 12-14, 1990. Edinburgh, Scotland. North-Holland.

[22] Scott Shenker. Service models and pricing policies for an integrated services internet. to appear in *Proceedings of "Public Access to the Internet"*, Harvard University, 1993.

[23] Marvin Sirbu. Telecommunications technology and infrastructure. In *A National Information Network*, Institute for Information Studies, 1992.

[24] Padmanabhan Srinagesh. Economic issues in the pricing of broadband services preprint, 1992.

[25] L. Taylor. *Telecommunications Demand: A survey and critique*, Ballinger, 1979.

[26] Robert Wilson. Efficient and competitive rationing. *Econometrica*, 57(1):1–40, January 1989.

[27] Lixia Zhang. Virtual clock: A new traffic control algorithm for packet switching networks. In *Proceedings of SIGCOMM '90*, September 1990.

[28] Lixia Zhang, Scott Shenker, and David D. Clark. Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. In *Proceedings of SIGCOMM '91*, pp 133-147, 1991.

Figure 3: Network Topology C

application type	$\bar{s}$	$\bar{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,2)
	6 pkts	2 sec	(1,2)
Voice	70 sec	90 sec	(1,2), (1,2), (1,2), (1,2)
FTP	100 KB	16 sec	(1,2)
	500 KB	31 sec	(1,2)
	1 MB	87 sec	(1,2)
	1 MB	98 sec	(1,2)
E-Mail	5 KB	6 sec	(1,2)
	10 KB	8 sec	(1,2)
	15 KB	8 sec	(1,2)
	50 KB	13 sec	(1,2)
	100 KB	19 sec	(1,2)

Table 1: Configuration 1 on Network Topology A.

application type	$\bar{s}$	$\bar{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,2)
	6 pkts	2 sec	(1,2), (1,2)
Voice	70 sec	90 sec	(1,2), (1,2), (1,2), (1,2)
FTP	100 KB	16 sec	(1,2)
	500 KB	50 sec	(1,2)
	1 MB	98 sec	(1,2)
E-Mail	5 KB	6 sec	(1,2)
	10 KB	8 sec	(1,2)
	15 KB	8 sec	(1,2), (1,2)
	50 KB	13 sec	(1,2), (1,2), (1,2)
	100 KB	19 sec	(1,2), (1,2), (1,2)

Table 2: Configuration 2 on Network Topology A.

application type	$\overline{s}$	$\overline{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,2)
	6 pkts	2 sec	(1,2), (1,2)
Voice	70 sec	90 sec	(1,2), (1,2), (1,2), (1,2), (1,2), (1,2), (1,2), (1,2), (1,2)
FTP	100 KB	16 sec	(1,2)
	500 KB	50 sec	(1,2)
	1 MB	98 sec	(1,2)
E-Mail	5 KB	6 sec	(1,2)
	10 KB	8 sec	(1,2)
	15 KB	8 sec	(1,2)
	50 KB	13 sec	(1,2), (1,2)
	100 KB	19 sec	(1,2), (1,2)

Table 3: Configuration 3 on Network Topology A.

application type	$\overline{s}$	$\overline{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,2)
	6 pkts	2 sec	(1,2), (2,1)
Voice	70 sec	90 sec	(1,2), (1,2), (2,1), (2,1)
FTP	100 KB	16 sec	(1,2)
	500 KB	31 sec	(2,1)
	1 MB	87 sec	(1,2)
	1 MB	98 sec	(2,1)
E-Mail	5 KB	6 sec	(2,1)
	10 KB	8 sec	(1,2)
	15 KB	8 sec	(1,2)
	50 KB	13 sec	(1,2)
	100 KB	19 sec	(2,1)

Table 4: Configuration 4 on Network Topology A.

application type	$\overline{s}$	$\overline{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,3)
	6 pkts	2 sec	(1,2), (2,3)
Voice	70 sec	90 sec	(1,3), (1,3), (1,2), (1,2), (2,3), (2,3)
FTP	100 KB	16 sec	(1,3), (2,3)
	500 KB	31 sec	(1,3)
	1 MB	87 sec	(1,2)
	1 MB	98 sec	(2,3)
E-Mail	5 KB	6 sec	(1,2)
	10 KB	8 sec	(1,3)
	15 KB	8 sec	(1,3), (2,3)
	50 KB	13 sec	(1,3), (2,3)
	100 KB	19 sec	(1,2)

Table 5: Configuration 5 on Network Topology B.

application type	$\overline{s}$	$\overline{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,6), (4,2), (5,6)
	6 pkts	2 sec	(1,2), (4,3), (5,3)
Voice	70 sec	90 sec	(1,2), (1,2), (1,6), (1,6), (4,2), (4,2), (4,3), (4,3), (5,3), (5,3), (5,6), (5,6)
FTP	1 MB	87 sec	(1,2), (4,3), (5,6)
	1 MB	98 sec	(1,6), (4,2), (5,3)
E-Mail	50 KB	13 sec	(1,2), (1,6), (4,2), (4,3), (5,3), (5,6)
	100 KB	19 sec	(1,2), (1,6), (4,2), (4,3), (5,3), (5,6)

Table 6: Configuration 6 on Network Topology C.

application type	$\overline{s}$	$\overline{t}$	(src host, dst host)
Telnet	9 pkts	4 sec	(1,6), (2,4), (6,5), (6,3), (7,3)
	6 pkts	2 sec	(2,1), (4,3), (5,3), (6,3)
Voice	70 sec	90 sec	(1,2), (2,1), (1,6), (6,1), (2,4), (4,2), (4,3), (3,4), (5,3), (3,5), (5,6), (6,5), (4,3), (3,4), (5,3), (3,5), (5,6), (6,5)
FTP	1 MB	87 sec	(1,2), (3,4), (5,6), (6,3)
	1 MB	98 sec	(6,1), (2,4), (5,3), (7,6), (7,3)
E-Mail	50 KB	13 sec	(1,2), (1,6), (4,2), (4,3), (5,3), (6,5), (7,6), (6,3), (3,7)
	100 KB	19 sec	(2,1), (6,1), (2,4), (3,4), (3,5), (5,6), (6,7), (3,6), (7,3)

Table 7: Configuration 7 on Network Topology C.

	Telnet			Voice			FTP			Email		
Configuration	num apps	% pkts	% bytes	num apps	% pkts	% bytes	num apps	% pkts	% bytes	num apps	% pkts	% bytes
1	2	4.6	0.6	4	24.0	10.7	4	54.9	68.2	5	16.6	20.6
2	3	6.2	0.8	5	27.7	13.0	3	29.6	38.6	10	36.6	47.6
3	3	5.9	0.9	5	42.3	22.5	3	29.3	43.3	7	22.5	33.3
4	3	7.1	1.7	4	22.8	9.5	4	51.8	65.8	5	18.2	23.1
5	3	5.5	0.7	6	27.6	12.8	5	48.6	62.8	7	18.3	23.6
6	6	5.8	0.8	12	29.8	14.2	6	35.5	46.8	12	29.0	38.2
7	9	6.0	1.5	18	31.7	14.0	9	32.3	43.7	18	30.1	40.8

Table 8: Aggregate Traffic Characteristics

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
570	88.58	2211	109.38	-188.30	(0.2681, 2.4802)	(0.5501, 1.0417)	(0.2757, 1.1307)
670	75.35	2211	163.50	-26.70	(0.2681, 2.4802)	(0.4792, 0.6429)	(0.2798, 0.6868)
1070	47.35	2221	277.65	233.23	(0.2681, 0.3381)	(0.2724, 0.2964)	(0.2736, 0.2987)
1270	39.42	2221	288.30	260.54	(0.2681, 0.3381)	(0.2705, 0.2898)	(0.2711, 0.2831)
2000	25.24	2221	232.34	224.90	(0.2681, 0.3381)	(0.2682, 0.2742)	(0.2683, 0.2686)
5000	10.10	2121	104.89	104.82	(0.2681, 0.3856)	(0.2681, 0.2682)	(0.2681, 0.2682)

Table 9: Simulation Results for Configuration 1.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
670	90.22	2211	62.30	-334.29	(0.2380, 1.7856)	(0.7789, 1.2343)	(0.2332, 1.3435)
870	70.38	2221	195.18	-4.81	(0.2239, 1.7856)	(0.2395, 0.4033)	(0.2526, 0.4059)
1070	56.87	2221	214.20	119.29	(0.2235, 0.4059)	(0.2297, 0.3501)	(0.2297, 0.3501)
1270	47.81	2221	220.70	171.81	(0.2235, 0.4059)	(0.2271, 0.3214)	(0.2271, 0.3214)
2000	30.28	2221	174.64	161.69	(0.2235, 0.4059)	(0.2239, 0.2543)	(0.2239, 0.2543)
4000	15.14	2221	96.19	95.77	(0.2235, 0.4059)	(0.2235, 0.2238)	(0.2235, 0.2238)

Table 10: Simulation Results for Configuration 2.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
670	81.39	2211	78.33	-468.95	(0.2503, 1.1212)	(0.5916, 0.8178)	(0.2615, 0.9753)
870	63.12	2221	154.14	-29.18	(0.2503, 0.3777)	(0.2651, 0.3505)	(0.2727, 0.3775)
1070	51.15	2221	198.26	109.58	(0.2503, 0.3777)	(0.2563, 0.3320)	(0.2594, 0.3016)
1270	43.03	2221	208.49	166.16	(0.2503, 0.3777)	(0.2537, 0.3111)	(0.2554, 0.2738)
2000	27.29	2221	171.86	163.29	(0.2503, 0.3777)	(0.2505, 0.2669)	(0.2507, 0.2510)
4000	13.64	2121	95.19	94.84	(0.2503, 0.5494)	(0.2503, 0.2505)	(0.2503, 0.2504)

Table 11: Simulation Results for Configuration 3.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
300	87.93	2211	35.81	-270.20	(0.2816, 2.5253)	(0.6576, 2.4664)	(0.3763, 2.5253)
450	60.24	2211	189.11	75.39	(0.2733, 2.5253)	(0.4775, 0.8111)	(0.3189, 0.8712)
650	42.05	2221	254.84	212.08	(0.2732, 0.3511)	(0.2904, 0.3029)	(0.2953, 0.3434)
1050	25.91	2221	317.55	296.04	(0.2731, 0.3511)	(0.2760, 0.2923)	(0.2768, 0.2873)
2000	13.58	2221	241.97	237.11	(0.2729, 0.3511)	(0.2730, 0.2773)	(0.2730, 0.2731)
4000	6.79	1121	131.06	130.94	(0.2728, 0.4072)	(0.2728, 0.2729)	(0.2728, 0.2730)

Table 12: Simulation Results for Configuration 4.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
570	78.01	2211	84.48	-337.52	(0.2094, 1.6068)	(0.4722, 0.9920)	(0.2240, 1.1045)
670	66.37	2211	145.10	-134.76	(0.2048, 1.6068)	(0.4187, 0.5685)	(0.2203, 0.6193)
1070	41.56	2221	254.40	211.89	(0.2045, 0.2662)	(0.2091, 0.2241)	(0.2106, 0.2341)
1270	35.01	2221	257.94	234.61	(0.2045, 0.2662)	(0.2067, 0.2191)	(0.2074, 0.2201)
2000	22.23	2221	205.98	201.44	(0.2045, 0.2662)	(0.2046, 0.2078)	(0.2047, 0.2049)
5000	8.89	2121	90.68	90.60	(0.2044, 0.3129)	(0.2044, 0.2045)	(0.2044, 0.2045)

Table 13: Simulation Results for Configuration 5.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
570	68.37	2211	225.62	-357.65	(0.1175, 0.8182)	(0.4097, 0.5087)	(0.1260, 0.5696)
670	58.17	2221	297.01	-76.71	(0.1175, 0.1912)	(0.1359, 0.1912)	(0.1474, 0.1912)
1070	36.42	2221	378.91	338.60	(0.1175, 0.1912)	(0.1191, 0.1518)	(0.1201, 0.1315)
1270	30.69	2221	378.91	340.50	(0.1175, 0.1912)	(0.1191, 0.1518)	(0.1201, 0.1302)
2000	19.81	2221	285.80	278.01	(0.1174, 0.1912)	(0.1175, 0.1250)	(0.1176, 0.1177)
5000	7.67	2121	120.26	120.19	(0.1174, 0.2420)	(0.1174, 0.1175)	(0.1174, 0.1175)

Table 14: Simulation Results for Configuration 6.

link speed (kbps)	link util (%)	$\vec{\sigma}^{max}$	V^{max}	V^{flat}	feasible price range	acceptable price range	adoptable price range
300	70.98	2211	229.55	-804.87	(0.0824, 0.5312)	(0.4926, 0.5312)	(0.1112, 0.5312)
400	53.23	2211	395.99	-209.54	(0.0804, 0.5312)	(0.3871, 0.5312)	(0.1002, 0.5312)
650	32.76	2221	654.67	454.43	(0.0799, 0.1323)	(0.0911, 0.1323)	(0.0984, 0.1323)
1000	21.29	2221	654.02	594.02	(0.0799, 0.1323)	(0.0818, 0.1208)	(0.0830, 0.0964)
2000	10.65	2221	449.10	441.77	(0.0799, 0.1323)	(0.0800, 0.0851)	(0.0800, 0.0810)
3000	7.10	2121	310.69	310.20	(0.0799, 0.1709)	(0.0799, 0.0802)	(0.0799, 0.0802)

Table 15: Simulation Results for Configuration 7.