# A Study of Priority Pricing in Multiple Service Class Networks

Ron Cocchi Deborah Estrin
Computer Science Department
University of Southern California
Los Angeles, California 90089-0782
cocchi@usc.edu estrin@usc.edu

Scott Shenker Lixia Zhang
Palo Alto Research Center
Xerox Corporation
Palo Alto, Ca 94304
shenker@parc.xerox.com lixia@parc.xerox.com

**Abstract**

We study the role of pricing policies in multiple service class networks. We argue that some form of graduated prices are required in order for *any* multiclass service discipline to have the desired effect. Moreover, we demonstrate through simulation that it is possible to set the prices so that every user is more satisfied with the combined cost and performance of a network with graduated prices. For some users the performance penalty received for requesting a less-than-optimal service class is offset by the reduced price of the service. For the other users the monetary penalty incurred by using the more expensive, higher quality service classes is offset by the improved performance they receive. Thus, prices allow us to spread the benefits of multiple service classes around to all users, rather than just having these benefits remain exclusively with users who are performance sensitive.

## 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 transfers, 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. Just as the performance of the network cannot be derived solely from protocol specifications (as pointed out by Clark [1]), it is also true that from the perspective of end users (applications), network performance cannot be derived solely from implementation specifics. Network performance is also a function of the offered load, and the offered load is a function of the incentives individual users encounter when using the network. Thus, the issue of user incentives must be considered. Note that these incentives can take many forms: performance incentives, monetary incentives, administrative incentives, or social incentives, to name a few.

This paper represents one small effort to grapple with user incentives in multiple service class networks. We restrict our focus to one particular aspect of network incentives; the intertwining of pricing policies, which produce monetary[1] incentives, and multiclass service disciplines, which produce performance incentives. Our treatment is by no means comprehensive, and much of what we contend is not controversial. We view this work as an initial attempt to illustrate, through simulations, some of the issues involved in the interplay between pricing policies and multiclass service disciplines. We consider a very simple multiclass service discipline and compare two different pricing policies: (1) flat per-byte fees and (2) graduated fees for different priority classes. Measuring user satisfaction as a function of both the cost and quality of service received, and studying a simple network configuration with several different applications using standard transport-layer protocols, our study shows that every simulated user is more satisfied with the graduated pricing scheme.

In Section 2 we provide additional motivation for our study of multiple service class networks. Section 3 describes the multiclass service discipline and pricing schemes under consideration. The model presented in Section 4 characterizes, for each of four example applications, user satisfaction as a function of the service delivered by the network. We present our simulation results in Section 5, and discuss their significance in the context of related work in Section 6.

## 2 Motivation

Today's Internet has four characteristics that are relevant to our discussion here. First, the overall bandwidth is quite limited (the backbone is currently comprised of T1 lines), which restricts the aggregate traffic load and prevents certain bandwidth-intensive applications from utilizing the Internet. Second, access to the Internet is severely restricted; this is in part because of the limited bandwidth available. The relatively small community of Internet users, and the shared history of creating the network, has fostered widespread adherence to socially desirable behavioral norms. Third, the

[1] Actually, *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 logs. While our framework could be generalized to these other forms of pricing, for the sake of clarity we will refer only to monetary incentives in this paper.

Internet offers a single type of service (TOS);[2] all packets are serviced on a best-effort, first-in-first-out (FIFO) basis. This single TOS 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; typically institutions are charged for access to a regional network. However, these fees are not based on the volume of traffic sent, and are typically not passed back to individual users.

The Internet of the future is likely to be quite different in each one of these respects. First, the backbone links are being upgraded to 45 Mbps T3 lines; 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 new bandwidth-intensive applications to utilize the Internet. 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 personal computers, makes it likely that the future Internet, or networks like it, will be publicly accessible. No longer will the network have, by virtue of access restrictions, a small, technically knowledgeable, and mostly cooperative community. As with other public goods, informal enforcement of behavioral norms is unlikely to be sufficient to ensure socially desirable behavior.

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 TOS 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.). 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.

The first 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 minimal delay. Similarly, some applications are relatively insensitive to packet loss while others are not, and some applications can adjust to reduced bandwidth while others cannot. As the user community grows, so will the range of application types and the diversity of service requirements. Thus, it is crucial for the evolution of the Internet that means be found for meeting these increasingly varied service requirements.

The second point of consensus is that networks can more efficiently meet these varied service requirements if the network offers multiple classes of service, so that users can choose the class of service that is most appropriate for their application.[3] The network can then, in periods of resource contention, allocate its resources so that resources are focused on the performance sensitive applications and not squandered on applications that are not performance sensitive. Notice that not all service classes necessarily get better performance under this multiple service class scheme. For example, traffic in a lower quality service class will typically 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 to it.[4]

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.[5] 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 TOS that maximizes their performance. In the absence of any other consideration, there is nothing to motivate the user to indicate that their application is less performance sensitive (which would thereby degrade the performance they receive, under some network conditions). Perhaps in a small and cooperative user community the behavioral norm of requesting the appropriate TOS 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. 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.

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 different TOS options viable, some service-class dependent usage fees are necessary as incentives for proper user behavior.

Our focus in this paper is precisely on the interplay between the monetary incentives provided by prices and the performance incentives provided by the service classes. Our initial goal is to examine these issues in an extremely simple network model. In this model, we assume users are sensitive to both the quality of service they receive and the price they have to pay for that service. Following the traditional

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

[3] 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.

[4] 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.

[5] 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.

practice in economics, we model this sensitivity through a combined "utility function" which describes a user's level of satisfaction with the combined network performance and cost. We construct simple utility functions representing users of several different applications: voice, Telnet, FTP, and Email. The simulated network's priority-service discipline (described in Section 3.1) represents a toy model of the sophisticated TOS mechanisms we expect to find in future networks. We then compare two different pricing schemes. One pricing scheme is a flat per-byte usage fee. In this case, we assume users have no incentive to select less than optimal service classes, and so the service mechanism reverts back to a single class of service discipline. The other pricing scheme is a graduated set of prices with the lower quality TOS's being cheaper. Here, there are incentives for requesting less than optimal service. We find, for the network configurations studied here, that it is possible to set the prices so that every user is more satisfied with the combined cost and performance of this network with graduated prices. Thus, in answer to the key question posed above, it was possible to set prices in our simulation in such a way that (1) for some users the performance penalty they received for requesting a less-than-optimal service class was offset by the reduced price of the service, and (2) for the other users the monetary penalty incurred by using the more expensive higher quality service classes was offset by the improved performance they receive. Thus, prices allowed us to spread the benefits of multiple service classes around to all users, rather than having these benefits remain exclusively with users who were performance-sensitive.

## 3 Network Model

In this section we present our network model, describing in turn the service discipline, pricing schemes, and network topology. Our goal is to provide a simple example that illustrates our point. Thus, our emphasis is not on detailed and complicated models, but on extreme simplicity.

### 3.1 Multiclass Service Discipline

At an abstract level, there are only two decisions faced by a gateway. When the line is free and there are packets in the queue, the gateway must select the next packet to transmit. When a packet arrives and there is no room in the queue, the gateway must decide which packet to discard. The simplest multiclass service discipline is to have, for each of these decisions, two priority levels. Thus, on each packet there is a priority service flag and a priority no-drop flag. There are then four service classes, corresponding to the four possible combinations of flag settings. In Section 4 we will discuss how to set these priorities, but for now we just discuss their role in determining the gateway's handling of the packets.

The gateway gives preference in service to those packets with the priority flags on. Within a priority class, service is provided in a FIFO order. Thus, when the line is idle, the earliest arrived packet that has the service priority flag on is serviced. If there are no such packets, then the earliest arrived packet without the service priority flag is serviced. We discard packets in the following order: both priority flags off, service priority flag on and no-drop flag off, service priority flag off and no-drop flag on, and finally both priority flags on.

### 3.2 Pricing Schemes

We consider two different pricing schemes. The first is a flat per-byte price applied to all packets traversing a link, call it $p_{flat}$. We refer to the second pricing scheme as priority pricing, based on a theoretical discussion provided in [10]. In this approach, we modify the per-byte prices based on the priority flags set in the packet. Let us denote these four per-byte prices by $p_{0,0}$, $p_{1,0}$, $p_{0,1}$, $p_{1,1}$, where the first bit in the subscript indicates whether or not the service priority flag is on, and the second bit indicates the status of the no-drop flag. Clearly the prices charged for packets with either or both of the priority flags turned on should be higher than those with neither flag turned on, so that $p_{0,0} < \{p_{0,1}, p_{1,0}\} < p_{1,1}$.

Our objective, as stated in the Introduction, is to demonstrate that there was at least one set of priority prices that every user prefers to flat pricing. In fact, this condition may be met by a wide variety of priority prices. We have chosen to present only the simplest of these schemes. We set a base price of $p_{priority}$ for each packet, and charge an additional $p_{priority}$ for each priority flag set. Thus, $p_{0,0} = p_{priority}$, $p_{0,1} = p_{1,0} = 2p_{priority}$, and $p_{1,1} = 3p_{priority}$.

In order to facilitate direct comparison between the two pricing schemes in the simulation study presented below, we require that both pricing schemes recover the same net revenue, which we refer to as $D$. Thus, the absolute values of the prices in the two schemes will depend on the offered load. This means choosing $p_{flat}$ and $p_{priority}$ so that the total revenue is equal to $D$.

### 3.3 Network Configuration

We study the interaction between the service discipline and the pricing scheme on a very simple network topology. This network, as shown in Figure 1, consists of two hosts connected to two gateways through two 10 Mbps Ethernets, respectively; the two gateways in turn are connected by a single bottleneck link. We assume that there are a number of users on host-1 running a variety of applications which require the transfer of data to host-2. In the next section we discuss our user model.

## 4 User Model

In this section we describe how we model users in our simple network. Each user is represented by a network application which sends data from host-1 to host-2 in Figure 1. Users care about the cost of running an application, which we will denote by $C$. Users also care about the performance of their application; this application performance is a function of the network performance. Let us denote by $V$ the application performance *degradation* apparent to the user due to network performance (the higher $V$ is, the worse the application is performing). For each application type we define a different $V_{application}$ to measure that particular application's performance sensitivities.

In what follows we describe these applications and how their perceived performance $V_{application}$ depends on the network behavior. We then discuss how to combine a user's perceived service and cost into a single evaluation (utility) function.

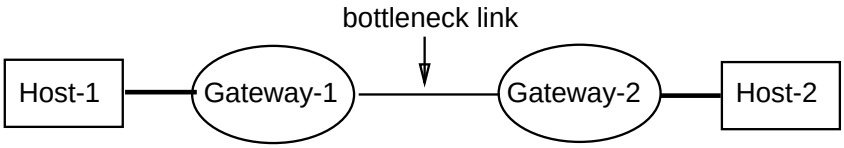


Figure 1: Simulation environment

### 4.1 Applications

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 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 merely to capture the essence of that relationship. We defer the description of the detailed traffic generation characteristics of these applications to Section 5.1. 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.

Email is used for multi-user, asynchronous communication. Since instantaneous delivery is not anticipated, 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:

$$V_{Email} = 0.1(\text{average message delay, in seconds}) + (\% \text{ of messages not delivered in loose delay bound})$$

In our model, FTP is a single user pseudo-interactive application. Since FTP users await completion of the application before proceeding with other tasks, we assume they want the network to deliver relatively prompt service. Accordingly, we model the FTP service requirements by:

$$V_{FTP} = (\text{average file transfer time, in seconds})$$

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 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:

$$V_{Telnet} = (\text{average packet round trip time, in seconds})$$

Voice is a real time application which is extremely sensitive to delay; voice applications cannot tolerate absolute delays above 200 milliseconds [3][6]. Voice differs from the other applications considered above 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 off 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 are discarded; from the perspective of the application dropped packets are no different than delayed ones. Our assumed application performance function is:

$$V_{Voice} = (\% \text{ of packets not obeying tight delay bound})$$

### 4.2 Utility Functions

A user's overall level of satisfaction is a function of both the application performance and the cost of running that application. We model this combined dependence through a utility function $U$ which depends on $V$ and $C$. A higher value of $U$ indicates a higher level of satisfaction. Since we assume that users always prefer lower costs and higher performance, this function $U$ could be any strictly monotonically decreasing function of these two variables. We have chosen, for simplicity, to set $U = -V - C$ for this initial study.[7]

Before proceeding, we should clarify what these utility functions are. Utility functions have played an integral role in standard economic theory since the mid nineteenth-century [5]. Utility functions are merely a way of representing an individual's preference ordering among a set of alternatives. Comparison of values generated by an individual's utility function yields an individual's preference of one alternative over another. The magnitude of the difference is irrelevant, only the sign is important. Since the utility function represents only an ordering, it does not make sense to compare absolute utility function values between different individuals. Similarly, we do not intend that the functions $V$ above should be used to compare the relative performance of two different applications. Utility functions merely reflect the preference ranking of network conditions by a single application under different network conditions.

### 4.3 User Behavior

In this section we discuss how users react to the various incentives. It is important to note that we have not attempted to capture overall demand elasticity in this model. In our current study we make the assumption that the total traffic generated by a user is independent of the price; *we model only the users' service class decisions.*

Users try to maximize their overall level of satisfaction $U$. If only the performance incentives were present, then all

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

[7] In addition to more general functions, setting $U$ to be an arbitrary linear combination of $V$ and $C$ is an obvious possibility. However, by varying the overall level of revenue generated, $D$, we have the same effect as an arbitrary linear combination.

users would set both priority flags on. If only monetary incentives were present, then all users would set both priority flags off. With both incentives present, the situation is a little less obvious.

In the flat pricing scheme, the monetary incentives are irrelevant, since the user has no action that can modify cost $C$.[8] Thus, all users set both priority flags on. In the priority pricing scheme, we assume that users only set a priority flag if it improves the network performance in a way that is relevant to their application. $V_{Email}$ only weakly depends on the average delay, and mostly depends on a very loose delay constraint which is unlikely to be threatened, even in cases of occasional losses and retransmissions, by not setting the priority flags. Therefore, simulated Email users choose to not set either of the priority flags. $V_{Telnet}$ is very sensitive to delay, and so simulated Telnet users set both priority flags. $V_{Voice}$ is sensitive to delay but can tolerate drops, so simulated voice users set the priority service flag but not the no-drop flag. $V_{FTP}$ is a function of the average file transfer time; since files tend to be relatively large, this transfer time is more a function of the average throughput than the delay of each individual packet. Thus, simulated FTP users choose not to set the service priority flag, but do set the no-drop priority flag. These service class decisions were made with the assumption of a moderately loaded network; under extreme overload or underload users might choose to set both priority flags in ways different from the above.

## 5 Simulation: Configuration and Results

We have so far given a general description of the problem we are addressing and the various issues involved. We now turn to describing the details of our simulation.

### 5.1 Simulation Configuration

The applications are built upon two transport protocols. Email, FTP, and Telnet use TCP [8] whereas Voice uses UDP [7]. 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 4.1 the user repeatedly requests service from their 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 $\overline{s}$ and $\overline{t}$ 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[9]. A voice request is a conversation; the size of the request is the duration of the conversation. During a conversation, 180 byte packets are transmitted in packet trains supporting a 64 Kbps data rate. The strict delay bound in $V_{Voice}$ is 200ms. The loose delay bound in $V_{Email}$ is 5 minutes.

[8] Recall that we do not consider demand elasticity, so users cannot adjust their offered load in our model.

[9] We model only the Telnet client.

We tested the benefits of priority pricing in two different configurations. In the first configuration, there were 2 voice applications, 4 FTP applications, 5 Email applications, and 2 Telnet client applications. In the second configuration, there were 2 voice applications, 3 FTP applications, 5 Email applications, and 4 Telnet applications. The values for the means of the request size and interval distributions $\overline{s}$ and $\overline{t}$ are in Tables 1 and 2 respectively. The bottleneck link in the network had a bandwidth of 772 Kbps in configuration 1 and 600 Kbps in configuration 2. In both configurations, the total revenue $D$ to be generated over the course of the simulation period was chosen to be \$100. The per-byte prices needed to raise this level of revenue were $p_{flat} = \$2.43 \times 10^{-7}$ and $p_{priority} = \$1.28 \times 10^{-7}$ in configuration 1 and $p_{flat} = \$3.05 \times 10^{-7}$ and $p_{priority} = \$1.75 \times 10^{-7}$ in configuration 2. Thus, in both configurations, the price $p_{0,0}$ of the cheapest priority class is roughly half the flat price $p_{flat}$. Furthermore, the prices $p_{1,0}$ and $p_{0,1}$ of the intermediate priority classes are roughly comparable to the flat price. Only the highest priority class price, $p_{1,1}$, is significantly higher (approximately 50% higher) than the flat price.

In terms of rough aggregate measures, the fraction of the bytes due to Telnet, Email, Voice, and FTP in configuration 1 are, respectively, 0.37%, 16%, 20.13%, and 63.5%. For configuration 2, the same figures are 1%, 27%, 26%, and 46%. Thus, the main difference between the configurations is that, in terms of percentages, configuration 2 has roughly three times the Telnet traffic, almost double the Email traffic, slightly more Voice traffic, and three-quarters the FTP traffic compared to configuration 1.

The total simulation time was 90 minutes per run, not including an initial warmup period of two minutes. In both instances, the bottleneck link was 80% utilized averaged over the 90 minute time period.

### 5.2 Results

For each configuration, we compared the flat pricing scheme with the priority pricing scheme. The values for $C$, $V$, and $U$ for each run are shown in Tables 1 and 2. Recall that it is inappropriate to compare $V$ and $U$ values between different applications; they are intended solely to compare the satisfaction of a given simulated application under the two situations. For instance, the fact that the $V$ values are different for the various FTP users is because they have different traffic generation parameters $\overline{s}$ and $\overline{t}$, not because one user is more satisfied than the other in some absolute sense. The main result apparent in these data is that every simulated user has a higher level of satisfaction with the priority pricing scheme.

When a flat price is set, all simulated applications use the highest quality service class available (i.e., both priority flags on), and the network performance is quite poor. Only about 82% of the voice packets are delivered on time in configuration 1, and only 65% in configuration 2. The Telnet delays ranged from 250ms to 1sec in both configurations. Note that this is not because there is no multiclass service mechanism available; rather, it is because in the absence of user incentives, we assume that every user requested high quality service.

In contrast, under priority pricing, users are motivated to choose the appropriate service class, and the multiclass service discipline enabled the more performance sensitive applications to achieve better performance. Roughly 99% of

the voice packets are delivered within the delay bound, the Telnet delays have decreased by an order of magnitude, and the FTP delays are significantly less (although the improvements are more moderate in configuration 2). For each of these simulated applications, the increased cost of the better service was worth the improved performance. However, further increases in the quality of service would not have yielded sufficient additional performance benefits to offset the additional cost. Thus, the priority pricing structure has provided the proper incentives for these simulated applications to choose the appropriate service class.

Email is the only application that received worse service under the priority pricing scheme. The performance degradation was modest, since the average delays are still very small, and all messages were delivered within 5 minutes (which was the loose delay bound in $V_{Email}$). This allowed the monetary benefits of the reduced price (half of the next cheapest service class) to outweigh any prospective performance gains.

This set of data demonstrates our central point: monetary incentives[10] can be used to induce users to choose the appropriate service classes and spread the resulting "utility" gains among all the users. This conclusion holds for the most straightforward of priority pricing schemes: $p_{0,0} = p_{priority}$, $p_{0,1} = p_{1,0} = 2p_{priority}$, and $p_{1,1} = 3p_{priority}$. We should note, however, that there are many other pricing schemes $p_{0,0}$, $p_{1,0}$, $p_{0,1}$, $p_{1,1}$ which every user prefers to the flat pricing scheme.

## 6 Related Work and Discussion

The novelty of our work is that it combines issues from several disparate fields. The weakness of the work is that 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 shortcomings of our 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, [4, 11]). 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 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 flags. However, it is doubtful that this approach could, in reality, support a sufficiently wide class of service requirements.

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 [6]. 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.

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 TOS requested, without some incentive to do so. The priority pricing literature in economics has shown that in the context of a simple model of allocation of a nonstorable good with fluctuating supply and constant demand, there are certain priority pricing schemes that can make everyone better off, when compared to a flat pricing scheme [10]. Our purpose, in this paper, was to apply this insight to a more realistic network setting. User responses to incentives has also been treated in the game theory literature. While only simple queueing network models are considered (see [9] and references therein), the incentive issues addressed are quite similar to those discussed in this paper.

In this first modeling effort we have only addressed the issue of TOS 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.

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 type of service delivered or priority), granularity (e.g., charge back to end users or to institutions for aggregate traffic) [2]. We have not addressed the many implementation and protocol support issues that arise in the context of any usage sensitive pricing systems.

## 7 Conclusions and Future Work

In this paper we have studied the role of pricing policies in multiple service class networks. We have demonstrated in a simulation that it is possible to set the prices so that every user is more satisfied with the combined cost and performance of a network with graduated prices and a multiclass service discipline.

On one level our conclusions are hardly surprising. Offering multiple service classes and charging differently for them is an obvious idea. However, it is a crucial idea that needs to be more fully explored. We expect that with a flat per-byte charge, user behavior will render the network equivalent to a single TOS network. We then think one of two outcomes is likely. One possibility is that the quality of service will not be high enough to support demanding applications like real-time video or voice, and 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 prices, 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.

We hope that this initial study will motivate further discussion, modeling, simulation, and eventually implementation activities. Our future work will pursue the issues mentioned in Section 6. We will also explore to what extent our conclusions, which are drawn from simulations of

[10] This statement also applies to other equivalent usage-sensitive incentives.

a simple network under moderate load, still apply to larger and more complicated networks under more extreme loads. Some questions that await answers are: (1) in these more complicated networks, does there always exist at least one set of priority prices that every user prefers to the flat pricing scheme, and (2) over what range of network loads do these prices retain their preferred property?

## References


[1] David D. Clark and David L. Tennenhouse. Architectural consideration for a new generation of protocols. In *Proceedings of SIGCOMM*, September 1990.

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

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

[4] 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.

[5] Walter Nicholson. *Microeconomic Theory Basic Principles and Extensions*, chapter 3, page 77. Dryden Press, fourth edition, 1989.

[6] 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.

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

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

[9] 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.

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

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

| application | $\overline{s}$ | $\overline{t}$ | $C_{flat}$ | $C_{pri}$ | $V_{flat}$ | $V_{pri}$ | $U_{flat}$ | $U_{pri}$ |
|---|---|---|---|---|---|---|---|---|
| Telnet(1) | 5 pkts | 2 sec | 0.17 | 0.27 | 0.26 | 0.05 | -0.42 | -0.32 |
| Telnet(2) | 8 pkts | 4 sec | 0.20 | 0.31 | 0.40 | 0.03 | -0.59 | -0.34 |
| FTP(1) | 100 KB | 16 sec | 8.42 | 8.96 | 19.16 | 10.86 | -27.58 | -19.82 |
| FTP(2) | 500 KB | 31 sec | 23.76 | 25.27 | 36.48 | 26.70 | -60.24 | -51.98 |
| FTP(3) | 1 MB | 87 sec | 16.60 | 18.05 | 175.32 | 76.15 | -191.92 | -94.20 |
| FTP(4) | 1 MB | 98 sec | 14.14 | 16.51 | 274.07 | 141.81 | -288.21 | -158.33 |
| E-Mail(1) | 5 KB | 8 sec | 0.82 | 0.43 | 0.05 | 0.33 | -0.87 | -0.76 |
| E-Mail(2) | 10 KB | 11 sec | 1.23 | 0.65 | 0.09 | 0.28 | -1.32 | -0.93 |
| E-Mail(3) | 15 KB | 10 sec | 1.92 | 1.02 | 0.16 | 0.39 | -2.07 | -1.41 |
| E-Mail(4) | 50 KB | 13 sec | 5.44 | 2.89 | 0.27 | 0.77 | -5.72 | -3.66 |
| E-Mail(5) | 100 KB | 19 sec | 6.69 | 3.56 | 0.47 | 1.16 | -7.16 | -4.72 |
| Voice(1) | 165 sec | 3 min | 11.46 | 12.28 | 18.34 | 1.24 | -29.80 | -13.52 |
| Voice(2) | 245 sec | 5 min | 9.15 | 9.80 | 17.58 | 1.42 | -26.73 | -11.22 |

Table 1: Utility Function Values from Configuration 1

| application | $\overline{s}$ | $\overline{t}$ | $C_{flat}$ | $C_{pri}$ | $V_{flat}$ | $V_{pri}$ | $U_{flat}$ | $U_{pri}$ |
|---|---|---|---|---|---|---|---|---|
| Telnet(1) | 6 pkts | 2 sec | 0.24 | 0.41 | 0.49 | 0.05 | -0.73 | -0.46 |
| Telnet(2) | 9 pkts | 4 sec | 0.30 | 0.51 | 0.46 | 0.03 | -0.76 | -0.54 |
| Telnet(3) | 10 pkts | 4 sec | 0.26 | 0.45 | 2.29 | 0.24 | -2.55 | -0.69 |
| Telnet(4) | 13 pkts | 5 sec | 0.24 | 0.42 | 0.38 | 0.03 | -0.62 | -0.45 |
| FTP(1) | 100 KB | 19 sec | 8.36 | 9.53 | 11.83 | 9.33 | -20.19 | -18.86 |
| FTP(2) | 500 KB | 33 sec | 23.10 | 26.40 | 27.41 | 23.83 | -50.50 | -50.23 |
| FTP(3) | 1 MB | 89 sec | 14.53 | 16.67 | 142.41 | 76.41 | -156.94 | -93.08 |
| E-Mail(1) | 5 KB | 4 sec | 2.13 | 1.22 | 0.12 | 0.30 | -2.25 | -1.52 |
| E-Mail(2) | 10 KB | 7 sec | 2.36 | 1.34 | 0.10 | 0.37 | -2.46 | -1.71 |
| E-Mail(3) | 15 KB | 8 sec | 3.32 | 1.89 | 0.18 | 0.47 | -3.50 | -2.36 |
| E-Mail(4) | 50 KB | 10 sec | 8.13 | 4.64 | 0.37 | 1.18 | -8.51 | -5.82 |
| E-Mail(5) | 100 KB | 14 sec | 11.09 | 6.33 | 0.78 | 3.58 | -11.88 | -9.91 |
| Voice(1) | 155 sec | 3 min | 7.89 | 9.24 | 35.20 | 0.93 | -43.09 | -10.17 |
| Voice(2) | 255 sec | 5 min | 18.06 | 20.96 | 33.53 | 0.69 | -51.59 | -21.65 |

Table 2: Utility Function Values from Configuration 2