Extracting Consumers' Private Information for Implementing Incentive-Compatible Internet Traffic Pricing

ALOK GUPTA, BORIS JUKIC, DALE O. STAHL, AND
ANDREW B. WHINSTON

ALOK GUPTA is an Assistant Professor in the Department of Operations and Information Management, University of Connecticut. He received his Ph.D. in Management Science and Information Systems from The University of Texas at Austin in 1996. His areas of specialization include data communication, electronic commerce, mathematical modeling of information systems, large-scale systems simulation, and economics of information systems. His research has been published in various information systems, economics, and computer science journals, such as *Information Systems Research, Communications of the ACM*, *Journal of Economic Dynamics and Control, Journal of Computational Economics, Decision Support Systems,* and *IEEE Internet Computing*. In addition, his articles have been published in several leading books in the area of the economics of electronic commerce. His current research and teaching interests are in the area of economic modeling and analysis of electronic commerce. He is co-director of Treibick Electronic Commerce Initiative, an endowed research initiative at the Department of OPIM, University of Connecticut. He serves on the editorial boards of *Decision Support Systems* and *Brazilian Electronic Journal of Economics.*

BORIS JUKIC is an Assistant Professor in the School of Management, George Mason University. He received his B.S. in Computer Science from the University of Zagreb and a MBA from Grand Valley State University. In 1998, he received his Ph.D. in Management Science and Information Systems from The University of Texas at Austin. His research interests include estimation of user demand characteristics and efficient infrastructure investment strategies for the Internet, as well as management of network resources. His research has been published in various information systems, economics, and computer science journals, such as *Journal of Computational Economics, Information Systems* and *IEEE Internet Computing.*

This research was funded in part by National Science Foundation #IRI-9225010, but does not necessarily reflect the views of the NSF. First author's research was supported in part by Treibick Electronic Commerce Initiative (TECI) at the department of OPIM, University of Connecticut.

Journal of Management Information Systems / Summer 2000, Vol. 17, No. 1, pp. 9–29.

0742–1222 / 2000 $9.50 + 0.00.

DALE O. STAHL is Malcolm Forsman Professor of Economics at The University of Texas at Austin. He received his B.S. and M.S. degrees in the field of Electrical Engineering from Massachusetts Institute of Technology in 1969 and 1970, respectively. In 1982 he received his Ph.D. from the University of California at Berkeley in the field of economics, with a focus on mathematical economics. Since then he has held positions at Duke University, M.I.T., Boston University, and Tilburg University in the Netherlands. He has published over 35 articles in the top economics journals in the areas of general equilibrium theory, dynamics and stability theory, game-theoretic approaches to price determination, and experimental game theory.

ANDREW B. WHINSTON is Director of the Center for Research in Electronic Commerce at The University of Texas. He received his Ph.D. at Carnegie Mellon University and is currently Hugh Roy Cullen Professor of Information Systems, Economics, and Computer Science at The University of Texas at Austin. He has published extensively on resource allocation issues, and is currently working on modeling the Internet to determine pricing strategies for both end user services and infrastructure. He has completed numerous research projects that integrate economics and operations research in the study of information systems issues. He has published over 250 articles in professional journals, including: *Management Science, Operations Research, American Economic Review, Journal of Political Economy, International Economic Review, Journal of Public Economics, Bell Journal of Economics and Management, IEEE Computer, IEEE Expert, Southern Economic Journal, Accounting Review, Decision Support Systems, ACM Transaction on Database Systems, Journal of Economic Theory, Water Resources Research, Econometrica, S.I.A.M. Journal of Mathematics, Journal of Combinatorics,* and *Information Systems Research.* He is Editor-in-Chief of *Decision Support Systems* and *Journal of Organizational Computing and Electronic Commerce*.

ABSTRACT: Internet traffic pricing is necessary for the vitality of electronic commerce because uncontrolled congestion creates a detrimental effect on quality of the Internet services. Pricing approaches based on negative externality have potential to address the issue of congestion. However, most externality-based pricing approaches require the knowledge of consumers' private demand characteristics, and this requirement is often pointed out as the single most important shortcoming of these mechanisms. The fact that the Internet is a "public good" presents challenging information extraction problems for network managers in implementing any pricing mechanism. Ideally, we seek *an incentive-compatible mechanism*—a means of extracting the required information that provides no incentives for users to alter their behavior in an attempt to manipulate the information extraction and price setting processes. We present a solution based on a new nonparametric statistical technique that was developed for this purpose. While the results in this paper are presented in the context of our prior research on pricing, the approach presented here applies to information extraction and implementation in other resource pricing approaches.

KEY WORDS AND PHRASES: demand estimation, economic mechanism design, incentive compatibility, Internet pricing, simulation.

DESPITE THE PHENOMENAL GROWTH in both the technology and applications spheres and the feverish pitch of excitement fueling electronic commerce (EC), there is a very real possibility that only a tiny fraction of the potential benefits of these innovations will be realized. Early Internet applications usually required low bandwidth usage and worked well with the best-effort service provided by Internet protocols. However, network applications have drastically changed in the last few years, with a tremendous increase in traffic volume and a growth in demand for higher-quality service. Also, the usage patterns have shifted significantly, with an average user spending much more time on the network. Consequently, the traditional fixed-fee charges, which provided unlimited access, have been struggling with overloads, resulting in poor data transmission rates.

The electronic commerce (EC) environment requires different quality of service (QoS) for different applications. For example, at the application level e-mail does not require any specific data transmission rate or variance in the data transmission rate and can be put in the lowest-priority class. On the other hand, real-time audio requires that the packet stream constituting the voice signal receive short and uniform delay with minimal variance. Additionally, the packets should arrive at the receiver's end in correct order. As the complexity of multimedia applications grows, the issues of interoperability also play a key role, such as the synchronization of audio and video streams for a movie.

The intention of providing and implementing multiple-priority classes with IPv6 (Internet Protocol version 6; see Deering and Hinden [3] for details) is to facilitate different QoS by channeling traffic through different priority classes. However, several challenges will be faced in implementing a protocol such as IPv6. First, an application-level QoS assignment does not account for users' need for a higher-priority for applications that may not require higher QoS from the point of view of application performance. For example, e-mail could be used in an Electronic Data Interchange (EDI) application where the timely receipt of the message is imperative and thus requires a higher-priority. Furthermore, current fixed fee Internet access approaches would be woefully inadequate in maintaining the intended use structure of priority classes. For example, users may send an e-mail in higher-priority by enveloping the packets in a higher-priority application. Even a higher access fee will not solve the issue. Instead, it will perhaps restrict the misuse to the users who subscribe to a higher class.

Further, even the projected growth of bandwidth will not manage to completely satisfy the growing information intensiveness (the amount of bandwidth required) and required degree of rapid and predictable performance for network applications (see [10] for a detailed discussion on these issues). Lack of adequate performance may lead to a severely crippled set of network applications that are incapable of providing value-added services to consumers. Specifically, applications whose performance is most critically dependent on the data transmission rate are bound to languish in a congested network.

Academics in information systems, economics, and, recently, computer science have begun to consider pricing as a means of ensuring the proper allocation of Internet infrastructure. Gupta, Stahl, and Whinston [5, 7, 8], Choi, Stahl, and Whinston [2], MacKie-Mason and Varian [13], and Wang, Peha, and Sirbu [21] are only a few of the many researchers that have argued that the goal of providing an adequate level of performance and customer satisfaction lies beyond technology—in the realm of resource allocation mechanisms based on economic theory. This trend is not surprising given the current popularity of deregulation. Markets seem to solve most of our resource allocation problems in the business world, so it is not unreasonable to think that markets with prices can solve the Internet resource allocation problem.

The fundamental idea of pricing is to find a price that clears the market, or in other words, to find a price such that all the demand at that price level is met. However, there is a fundamental difference in the problem environment presented here. When the demand is a stochastic flow (such as the demand for Internet services), "market clearing" can be obtained at infinitely many prices because congestion delays will adjust the throughput.

For example, consider a simple M/D/1 system (a Poisson arrival process into a queue for a single server with deterministic service time), and assume that the arrival rate depends inversely on the queue waiting time, so as waiting times become very large the arrival rate falls to zero. Then, with no pricing, the "stochastic equilibrium" will be characterized by a steady-state queue waiting time, which induces a steady-state arrival rate into the system that is achieved through self-perpetuating rationing. In other words, if the waiting time is any higher, then the arrival rates will tend to go down because of excessive waiting time, and if the waiting time is any lower then the arrival rates will tend to increase. On the other hand, if the arrival process is also influenced by prices, then imposing prices will result in another stochastic equilibrium (i.e., steady-state arrival rate and waiting time). Note that extremely high waiting times result in high losses to the economic system in the form of opportunity cost due to delay, and extremely high prices result in economic losses due to underutilization. Therefore, to achieve the goal of maximal system-wide steady-state benefits, one must impose not just any price, but the correct (optimal) price.

How can a system manager know what the correct prices are in such a dynamic stochastic environment? In regular markets, excess supply signals that the price should be decreased and excess demand signals that the price should be increased. Thus, by simply watching these readily observable signals, a manager can steer the price toward a market equilibrium. However, in an M/D/1 system, there must always be "excess supply" in the steady state (or else the queues would become infinitely long). Hence, the apparent "signal" suggests always cutting prices, but that is not the optimal solution. The optimal solution is to set the price of a job equal to the aggregated cost of additional delay, imposed by this job, on all other jobs (see [7, 12, 13, 14] for some representative samples of externality pricing). To determine this optimal price, a system manager must have knowledge of the value of time of its users (customers). This requirement is analogous to requiring knowledge of potential buyers' private values for, say, antiques.

How can this knowledge be obtained? In general, it would be naive to simply ask the user to reveal this private information, especially if the respondent knows that the answer will be used to set the price. [Buyer: "How much is that Chippendale table?" Seller: "How much is it worth to you?"] The respondent has no incentive to reveal this information truthfully, and has a definite incentive to provide misinformation when the price can be so affected. Economists call this the "incentive compatibility" problem. William Vickrey [20] provided one celebrated solution: the second-price auction. Recently, MacKie-Mason and Varian [13] suggested using second price auctions for Internet traffic pricing. Unfortunately, this and related auction mechanisms will not solve the problem in a dynamic stochastic environment (see Stahl [19]). Auction mechanisms such as the one presented by MacKie-Mason and Varian also have several other problems, such as the problem of packet valuation (consumers have value for a service, not for individual packets) and the problem of dividing the total bid in sub-bids at multiple hops during a transmission. Shenker et al. [18] and Gupta et al. [6] provide a critique of various suggested pricing mechanisms.

Naor [16], Mendelson [14], Mendelson and Whang [15], Westland [22], Lode and Lee [12], and Lederer and Lode [11] have proposed that levying congestion based tolls can result in optimal allocation of resources. However, none of these researchers discuss computational aspects of these pricing mechanisms. In our prior research [7], we developed a model for network resource pricing based on congestion pricing and showed that it is feasible to compute prices in real-time. All these studies assume that consumers' demand characteristics (in terms of their delay costs) are known. However, Shenker et al. [18] criticize these approaches and claim that the consumers' delay costs are *fundamentally unknowable,* and even though, theoretically, these pricing mechanisms are incentive-compatible the consumers will not provide their delay costs.

In this paper, we address the issue of estimating these (fundamentally unknowable) delay costs based on users' choices. The results presented in this paper are presented in the context of our prior research to readily demonstrate the viability and validity of our approach. However, the approach presented here has application for virtually any externality pricing scheme. Our solution is along the lines of classical microeconomic theory. First, note that we usually do not worry about incentive compatibility issues for markets of ordinary commodities, *because* we assume that there are so many participants that one person's actions will have a negligible effect on the market price. If we did not make this assumption, then the optimality of competitive market equilibria would fail to hold, and the foundations of our free market society would crumble. Therefore, we feel comfortable and justified in using this same assumption of many participants, which means that our network users will not attempt to change their market behavior in an effort to manipulate network prices. Our problem is then reduced to designing a statistical method of uncovering the critical information (users' values of time) from their observed market behavior. If we can solve this statistical inference problem, then we can use that information to set optimal prices, and we will have no incentive-compatibility problem at all.

Econometricians are accustomed to inferring demand characteristics from market observations. However, it turns out that in this dynamic stochastic environment many of the standard techniques, such as maximum likelihood estimates, fail to yield statistically consistent results [4, 9]. The reason for this failure has to do with the network traffic characteristics. The network traffic characteristics are found to be fractal [1], that is, the short-term demand pattern has much larger variation and mean than the long-term time-averaged demand pattern. This means that while there may be periods of no demand there will be periods when demand will be extremely high. Since dynamic prices have to reflect the changing status of network nodes, only short-term observable demand should be used for estimating any parameter. However, because of fractal demand the traditional parameterized econometric methodology does an extremely poor job of estimation. Therefore, we had to invent and test a new technique tailored to network data characteristics. This paper presents this new nonparametric statistical technique and the results obtained from applying it to a price computation mechanism that was tested via a simulation study.

The main purpose of this paper is to demonstrate the consistency and robustness of this technique. Our testbed is a simulation platform that was developed to test dynamic-priority pricing (see Gupta, Stahl, and Whinston [7, 8]). When using our technique to estimate values of time and to set prices *in real time,* we find that the system-wide efficiency loss (compared to having perfect information) is minimal. Thus, we have succeeded in designing an incentive-compatible mechanism for pricing Internet traffic that only requires the knowledge of observable user actions.

This paper is organized as follows. Section 2 presents a brief description of the priority pricing used in Gupta, Stahl, and Whinston [7] and the associated computational incentive-compatibility problem.[1] Section 3 describes our approach for estimating the delay cost parameter from the observable choices made by users. Section 4 presents results from the simulation study, which evaluates the effectiveness of our estimation technique by comparing the system performance with estimation to the performance with correct information. Finally, conclusions are presented in section 5 with directions of future research.

Network Externality Pricing and Incentive Compatibility

IN THIS SECTION WE REVIEW THE THEORETICAL AND SIMULATION MODEL proposed in Gupta, et al. [7] and present the incentive-compatibility problem associated with computing these prices. Figure 1 depicts the schematic representation of the priority pricing approach presented in [7]. In this model each service can be delivered using several different *schemes*. Each scheme involves potential processing at several *servers*.[2] Each of the servers is associated with a priority queuing mechanism having a predetermined number of priority classes. Associated with each priority class is an expected waiting time and a posted price.[3] Thus, the total number of *options* a user has for a given service request is the product of the total number of schemes for that service and the total number of priority classes.

As shown in Figure 1, the user[4] demand process is modeled as a stochastic process. Each user, i, and service, j, is randomly assigned an instantaneous value and a rate of decay for this instantaneous value—delay cost factor (δ_{ij}). The instantaneous value represents the increase in net worth of a user by obtaining the service if there were no delay and if the user were not charged anything for the service. Each available *scheme*[5] for a service request is then evaluated, in all available priority classes, for the total cost, which includes the total price and the total delay cost.[6] A user will choose an option (a scheme in a particular priority class) with minimum total cost, since it maximizes her net utility for the service. Note that for identical services the minimum total cost for different users will be different. For example, a user with a delay cost factor of zero will choose the server and the priority class with minimum price regardless of the actual delay while, a user with a high delay cost factor will perhaps choose a server and a priority class that has smaller delays.

The optimal prices in this model are computed to maximize the nonpecuniary benefits (i.e., total of instantaneous benefits—total delay costs). Gupta et al. [7] prove that these prices ensure that the economic system achieves stochastic equilibrium where: (1) expected delays are correct ex-post average delays, that is, users' expectations are met on average, and (2) realized flow in the system is such that users' utility is maximized given the prices and expected delays.

The price at a particular server for a particular priority class is characterized by the following system of equations in [7]:

$$p_{mk}(q) = \Sigma_l [\partial \Omega_l / \partial \chi_{mkq}] \Sigma_i \Sigma_j \delta_{ij} x_{ijlm} \qquad (1)$$

where: $p_{mk}(q)$ is the price of a job of size q at server m for priority class k
χ_{mkq} is the arrival rate of jobs of size q at machine m in priority class k
Ω_l is the expected waiting time in priority class l at machine m
δ_{ij} is the delay cost parameter of consumer i for service j
x_{ijlm} is the flow rate of service j for consumer i with priority l at server m

The first term on the right side ($\partial \Omega_l / \partial \chi_{mkq}$) is the derivative of waiting time with respect to the arrival rate of jobs sized q. Since the waiting time is a strictly increasing function of this arrival rate, an increase in the arrival rate of a certain priority class increases the prices for that priority class. The second term ($\Sigma_i \Sigma_j \delta_{ij} x_{ijlm}$) can be interpreted as the aggregated delay cost of the system. An increase in this cost increases the price. Since the jobs in the highest-priority class impose delays on the jobs in all other priority classes, whereas the jobs in the lowest-priority classes impose very little delay on the jobs in other priority classes, the prices for higher-priority classes are higher than those of lower-priority classes.

Another important thing to note here is that prices on each machine depend only on the measurable parameters at that machine and no network-wide information is required. Therefore, the size of the network does not increase the complexity of price computation.

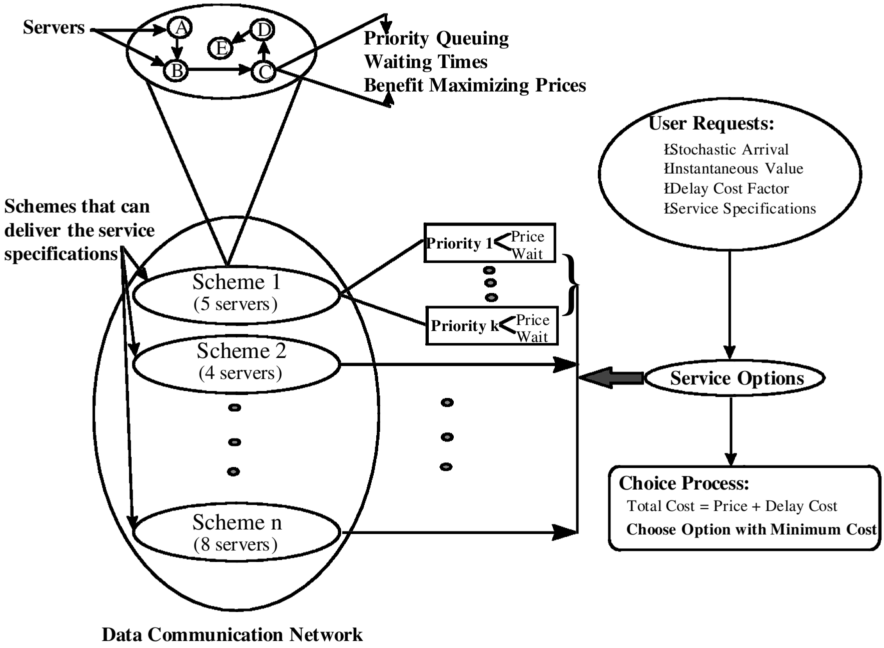

Figure 1. Schematic Representation of the Network Priority Pricing Model

The problem of incentive compatibility arises here because of the delay cost factor δ_{ij}, which is the users' private information and is unobservable. The prices are based on the values of δ_{ij}. However, this information is not directly available to the network manager. If directly asked, users have no incentive to truthfully report their δ_{ij}. Moreover, some *large* users may believe that they can affect the prices and hence network traffic characteristics by misreporting their delay cost factor. Theoretically, in the design of incentive-compatible mechanisms, it is assumed that even though the users' private information is unobservable, this private information comes from a *known* distribution. In our case, if we know the distribution of δ_{ij} and the waiting times can be predicted exactly, the derived prices will be incentive-compatible since a rational user will suffer higher costs if they do not choose the minimum cost option. However, the problem we face in a real-world implementation of pricing arises from the violation of the assumption of known distribution for private demand characteristics.

In the Internet environment the demand characteristics change rapidly with respect to time, and long-term demand patterns are of little use. These demand patterns shift drastically during the day and from day to day. In such an environment, a dynamic and real-time price-computing mechanism is needed. Associated with such a price-computing mechanism is the problem of *computational incentive compatibility*, that is, How do we set prices to provide an incentive-compatible mechanism without known distribution for users' delay cost characteristics? We will address this problem in "Estimation of Delay Cost Factors."

Even if delay cost factors are known, there are still significant challenges in computing prices in a dynamic environment where predictions based on long-term demand patterns are of little use. A real-time mechanism, which computes prices on an ongoing basis and changes them periodically, is a necessity if multiple service classes with predictable performance classes have to be maintained. We developed and tested real-time computation of priority prices via simulation based on the system of equations presented earlier. The parameter estimates in our simulation are based on the system feedback (for example, information obtained from a Topology Management Application's[7] time-based polling). However, since these prices are not estimated at the equilibrium conditions, they are approximate at any given time. Supposing (t, t+1) is the time interval between successive estimations, the following iterative equation can be used to update the prices at any given time (t+1):

$$p_{mk}(t+1) = \alpha \underline{p}_{mk}(t+1) + (1-\alpha)\, p_{mk}(t) \qquad (2)$$

where: α is a number between (0,1)
$\underline{p}_{mk}(t+1)$ is the estimated new price for time (t+1) using Equation 1
$p_{mk}(t)$ is the implemented price during the time (t, t+1)

Updating the prices this way provides a shield against local fluctuations in demand and in the stochastic nature of the process. In other words, if short-term stochastic demand is higher (or lower) than long-term demand, it should not affect the prices inordinately. Only if the relatively higher (or lower) demand is observed on a sustained basis should the prices be changed in the appropriate direction. Essentially, α defines how close the implemented prices are to computed prices in the current time period. A lower value of the parameter indicates that the price adjustment will be gradual in time, whereas a higher value will result in potentially large changes in the prices from period to period. However, computation of $\underline{p}_{mk}(t+1)$ requires the knowledge of delay cost factors and thus can be manipulated by users by providing incorrect information. In the next section we describe an approach for delay cost estimation that ensures an incentive-compatible mechanism by avoiding the necessity of asking users for their delay cost factors.

Estimation of Delay Cost Factors

WE DEVELOPED SEVERAL SIMULATION MODELS to evaluate the performance of the dynamic price computation method using Equation 2. The simulation model relevant for this study has 50 servers and 100 distinct services. A server can provide up to 25 distinct services. In earlier studies, we evaluated the performance and stability of our pricing mechanism against different pricing policies, such as fixed charges (or zero pricing) and time-based (or flat rate) pricing. The comparative evaluation was based on the net benefits and consumer surplus accumulated in the system under different pricing policies.[8] These results indicated that priority pricing produces significantly higher net benefits and consumer surplus as compared to other pricing policies. The complete description of the simulation model and results are presented in [8].

The user decision process in this model is depicted in Figure 2. V_{ij} is the instantaneous value of service j to user i. In the simulation models we assume a distribution

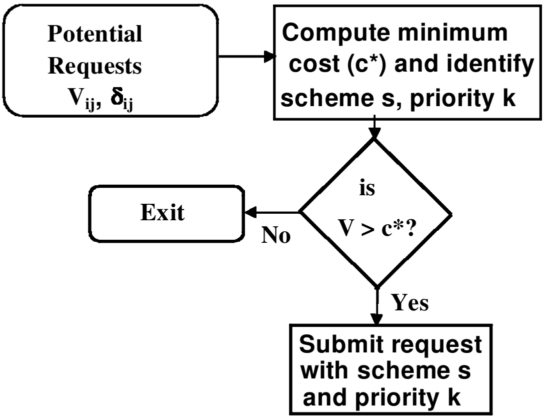
Potential Requests V_{ij}, δ_{ij}
Compute minimum cost (c*) and identify scheme s, priority k
is V > c*?
Exit
No
Yes
Submit request with scheme s and priority k

Figure 2. User Decision Process

for users' instantaneous value (V_{ij}) for a service and their delay cost factors (δ_{ij}). Services are available among a variety of "schemes" (e.g., different servers, scheduling, priorities), and each scheme generates a cost consisting of the monetary cost (based on prices p_{jmk}) and delay cost proportional to the delay cost factor δ_{ij} and the throughput time w_{jmk}. The minimum cost is denoted by:

$$c^*_{ij} = \min_{m,k} [\delta_{ij} w_{jmk} + p_{jmk}] \tag{3}$$

As discussed earlier, users may have incentives to misreport their δ_{ij} for several reasons. However, a rational user will still choose a minimum-cost scheme according to their true δ_{ij}. Thus, user choices potentially reveal information about the distribution of δ_{ij}.

As an example, consider the case where a particular service can be provided by three servers, each with a certain price and estimated throughput times. Each of these three servers could be the optimal choice for users with a certain range of delay cost factors, since the total cost is a combination of price charged and delay cost. Figure 3 graphically illustrates this example. Users with delay cost factors between 0 and 3 will choose server 1, users with delay cost factors between 3 and 6.34 will choose server 2, and users with delay cost factors of more than 6.34 will choose server 3. Thus, the user's choice reveals information about her δ_{ij}, which can be used to learn about the underlying distribution of δ_{ij}.

Assuming that the δ_{ij} are independent and identically distributed (as in our simulation), let $F(\delta_{ij})$ be the underlying distribution, with finite support. For each update period, given the prices and waiting times at each server, we partition the support of F into intervals corresponding to the piece-wise linear cost functions (as in Figure 3) and then using this partition we can estimate the δ by measuring the arrivals corresponding to each partition. During each period we can represent the data collected at each server by a tuple (p_{sm}, w_{sm}, n_{sm}), where p_{sm} is the price for service s at server m, w_{sm} is the estimate of throughput time for service s at server m, and n_{sm} is the number of requests submitted for service s at server m.[9] For a given service s, we will have the data from several servers, which can be used to compute the underlying delay cost

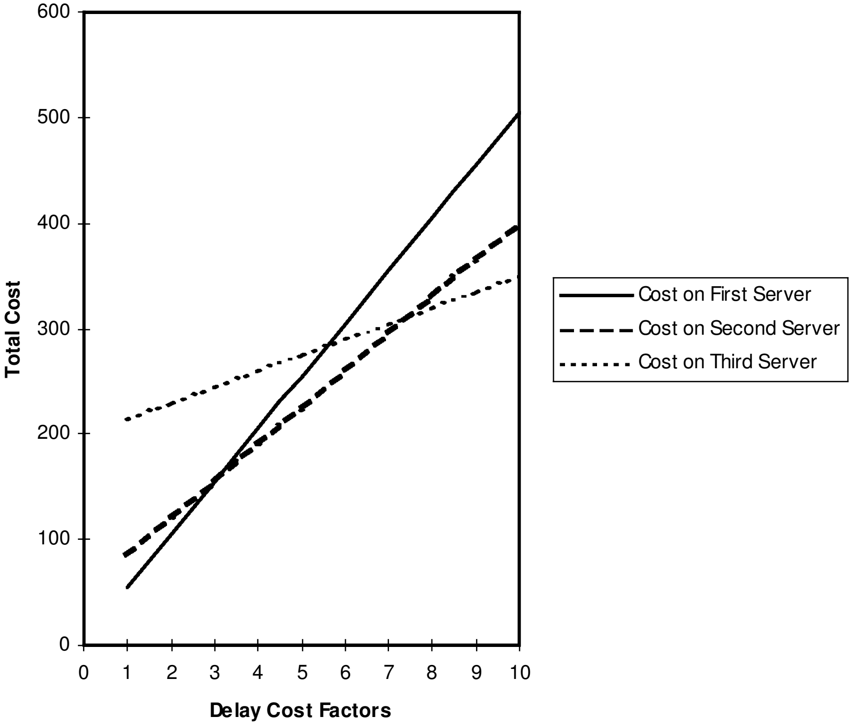

Figure 3. Optimal Choices for Users with Different Delay Cost Factor

distribution. The following two propositions form the basis of our computational scheme, which computes a *current period histogram* for the underlying distribution of δ for a particular service s.

Proposition 1: If there are two servers, m′ and m″, such that $p_{sm'} < p_{sm''}$, and both m′ and m″ received requests for service s during a given period, then $w_{sm'} > w_{sm''}$.

Proof: It is straightforward to see that if $w_{sm'} \leq w_{sm''}$, then ∀δ, the optimal cost, $c^*(s) = \min_m (p_{sm} + \delta\, w_{sm})$, will be at server m′ and hence no user will choose server m″.

Proposition 2: If there are two servers, m′ and m″, such that $p_{sm'} < p_{sm''}$ and $w_{sm'} > w_{sm''}$, then any user having a $\delta \leq (p_{sm''} - p_{sm'})/(w_{sm'} - w_{sm''})$ will choose server m′ for their service.

Proof: A user will choose m′ if

$$p_{sm'} + \delta\, w_{sm'} \leq p_{sm''} + \delta\, w_{sm''} \qquad \text{(P2.1)}$$

$$\Rightarrow \qquad \delta \leq (p_{sm''} - p_{sm'})/(w_{sm'} - w_{sm''}) \qquad \text{(P2.2)}$$

Using Propositions 1 and 2, we can easily design a computational algorithm where for a given service s, we can sort the data record (p_{sm}, w_{sm}, n_{sm}) in ascending order of

p_{sm}. Then, starting from a δ value of 0 we can compute the intervals corresponding to each partition by computing $(p_{sm''} - p_{sm'})/(w_{sm'} - w_{sm''})$, iteratively. Let $|s|$ denote the number of services provided by the network. Then, in each period we compute $|s|$ histograms, which correspond to underlying delay cost distributions for each service.

The data collected in each period could be affected by the stochastic nature of the process and, for example, may be concentrated too high or too low as compared to the true population characteristics. To adjust for such "local disturbances" we need to refine the estimates generated by *current-period histograms*. The refinement is done over time by merging data from subsequent periods in Bayesian fashion, that is, by merging the information from each current-period histogram into an existing *cumulative histogram.* That is, we use a historical distribution to further refine the current distribution, which is estimated by observing user choices, and then incorporate this information into historical information.

For instance, suppose an interval (δ_l, δ_u) has a frequency of n, in a current-period histogram. Suppose in the historical data we have some subintervals of (δ_l, δ_u), say (δ_l, δ_b) and (δ_b, δ_u). Now, the distribution in these subintervals can be used to further refine the current period histogram. For example, if there are twice as many users in the historical histogram in the interval (δ_b, δ_u) as compared to (δ_l, δ_b), then we can allocate (n/3) users from the current histogram to the subinterval (δ_l, δ_b) and (2n/3) users to subinterval (δ_b, δ_u). In most instances, however, we will not find an exact match on the upper and lower bounds for the different intervals. In such cases further numerical extrapolation is used to allocate the appropriate number of users in each subinterval. The accumulated data for all the services creates a new knowledge base at the end of each period, which in turn is used for the Bayesian update during the next period.

The process of merging the current-period histogram into the cumulative histogram, with predefined intervals with a large upper bound on δ_{ij}, can be described in the following six steps:

(i) Let the cumulative histogram have a fixed interval width of δ_F with the maximum bound δ_m. Further, let the current period histogram, developed using Proposition 2, have the interval boundaries $0, \delta_1, \delta_2, ..., \delta_m$, where $(0, \delta_1)$ defines the first interval, (δ_1, δ_2) defines the second interval, and so forth. To update the *cumulative histogram* in a Bayesian manner we create a temporary set of subintervals by joining the intervals from the cumulative histogram and the current-period histogram. For example, suppose $\delta_1 < \delta_F < 2\delta_F < \delta_2$, then the temporary subintervals are going to be $(0, \delta_1)$, (δ_1, δ_F), $(\delta_F, 2\delta_F)$, $(2\delta_F, \delta_2)$, and so on.

(ii) Next, the total cumulative arrivals are redistributed into the new temporary intervals. Let the cumulative histogram have N_1 users in interval $(0, \delta_F)$, N_2 users in interval $(\delta_F, 2\delta_F)$, N_3 users in interval $(2\delta_F, 3\delta_F)$. . . . Then the redistribution is done as follows:

Ł If a temporary interval is the same as an interval in the cumulative histogram, then the number of arrivals in that interval is not modified. For example, in the temporary interval (δ_F, $2\delta_F$) constructed in (i) the number of users remains N_2.

Ł If a temporary interval is a subdivision of an interval in the cumulative histogram, then the number of arrivals is apportioned to the temporary interval in the proportion of its relevant length to the total length of the interval in the cumulative histogram. For example, in the temporary interval (0, δ_1) we apportion N_1*[(δ_F – 0)/δ_F] users and in the temporary interval (δ_1, δ_F) we apportion N_1*[(δ_F – δ_F)/δ_F] users.

(iii) The new arrivals entailed in the current-period histogram are added to the temporary intervals using the same approach as described in step (ii). For example, let the current-period histogram have n_1 users in interval (0, δ_1), n_2 in interval (δ_1, δ_2), n_3 in interval (δ_2, δ_3). . . . Now, for the example presented in (i), the redistribution in the temporary histogram is done as follows:

Ł In the subinterval (0, δ_1), n_1 users are added to N_1*[(δ_1 – 0)/δ_F] users from step (ii)

Ł In the subinterval (δ_1, δ_F), n_2*[(δ_F – δ_1)/(δ_2 – δ_1)] additional users are apportioned; in the subinterval (δ_F, $2\delta_F$), n_2*[δ_F/(δ_2 – δ_1)] additional users are apportioned; and in the subinterval ($\delta 2_F$, δ_2) n_2*[(δ_2 – $2\delta_F$)/(δ_2 – δ_1)] additional users are apportioned.

(iv) Finally, the temporary intervals are merged into the original fixed intervals of the cumulative histogram by adding the subintervals. For instance, for the example in (i), the temporary subintervals (0, δ_1) and (δ_1, δ_F) are merged to recreate (0, δ_F) with a total number of users (N_1 + n_1 + n_2*[(δ_F – δ_1)/(δ_2 – δ_1)]).

Illustrative Example

Let us illustrate the approach with the help of a simple example. Suppose a current-period histogram is as represented in Figure 4 with 300 submissions for a particular service *s,* with 50 of them having a δ in the range of 0–10 and the rest having a δ in the range of 10–20. Let the existing cumulative histogram, based on historical data so far, have a distribution as shown in Figure 5, with 20% of the submissions in the δ range of 0–5, 30% in the δ range of 5–10, 30% in the δ range of 10–15, and 20% in the δ range of 15–20. Now we redistribute the current-period histogram such that 40% (or 20) of the submissions in the δ range of 0–10 are apportioned to the subrange 0–5. This is done because if we look at the cumulative histogram, 40% (200) of the total 500 observations in the δ range of 0–10 are in the subrange of 0–5. Using similar principles, 60% (or 30) of the submissions of the current period in the δ range of 0–10 are apportioned to the subrange 5–10; 60% (or 150) of the submissions in the δ range of 10–20 are apportioned to the subrange 10–15; and 40% (or 100) of the submissions in the δ range of 10–20 are apportioned to the subrange 15–20. The resulting new cumulative histogram will be as shown in Figure 6, with 20.37% submissions in the δ range of 0–5, 30.56% in the δ range of 5–10, 41.67% in the δ range of 10–15, and 27.77% in the δ range of 15–20.

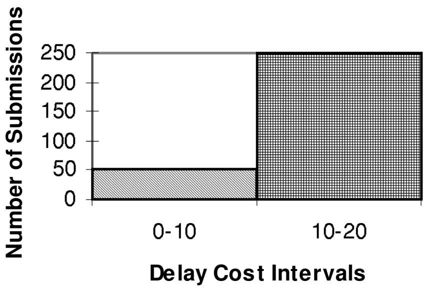

Figure 4. A Crude Histogram for Service s

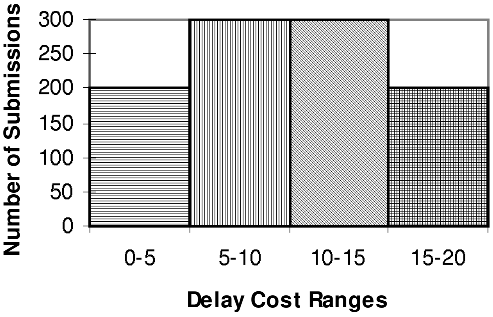

Figure 5. Existing Refined Histogram

After a new cumulative histogram is created, it is used to compute the mean value of δ, which is then used for the purpose of computing new prices. In the next section we provide some computational results, which indicate that the estimation procedure performs with minimal loss in efficiency with respect to system-wide benefits.

Computational Results

THE FIRST STEP IN THE DEVELOPMENT OF THE ESTIMATION PROCEDURE was to check the robustness of the price-computation mechanism with respect to delay cost factors. Specifically, we wanted to know what levels of estimation errors might be tolerable and not affect the system performance with respect to system-wide benefits. To explore this we added random noise, in our simulations, in the delay cost factors used to compute prices to simulate the misinformation that network managers might have. However, since users do know their true delay costs, the simulated users made their submission decisions using actual delay cost factors. We then compared the net benefits generated in the simulation runs with random noise to those where actual delay costs were used to compute prices. Figure 7 presents results from simulation runs where the actual delay costs were generated from a N(4,1) distribution and the error factor was generated from a N(0,0.3) distribution. The horizontal axis on this figure represents the exogenous arrival rate (demand) and the vertical axis represents the net

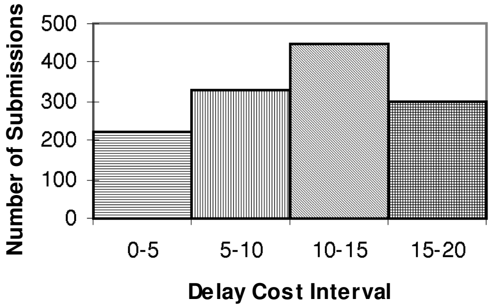

Figure 6. New Refined Histogram

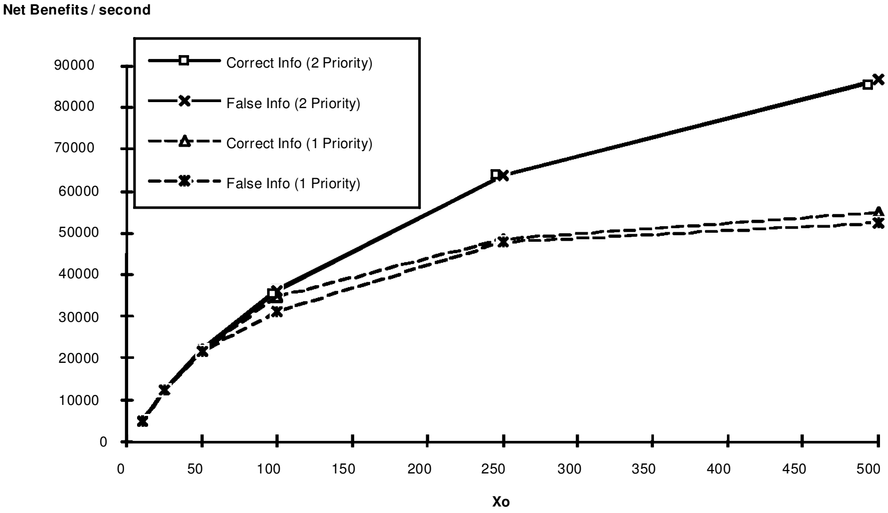

Figure 7. Robustness of the Price Computation Mechanism

benefits per second. As the figure shows, there is minimal loss in efficiency. In fact, in 2-priority cases the benefits are virtually indistinguishable.

Next, we used our simulation model to assess the robustness of the estimation procedure described earlier by comparing the actual delay cost factors to estimated delay cost factors. Table 1 presents the mean relative absolute deviation[10] of the estimated delay cost factor values for the six cases presented below. The largest deviation is 10.48%. It also appears that if the underlying distribution has a larger standard deviation, the estimates have a larger absolute deviation.

The results of Figure 7 and Table 1 suggest that our estimation procedure is likely to work well. However, there are two additional issues. First, the estimates shown in Table 1 were computed using simulation data where prices were computed based on actual δ_{ij}. However, it is conceivable that prices based on an estimated aggregate δ, using the approach described in this paper, could distort the data, resulting in a dete-

Table 1. Mean Relative Deviations of δ Estimates

Load	Underlying Distribution	Relative Deviation (%)
50	N(4, 1)	5.07
100	N(4, 1)	2.56
200	N(4, 1)	2.22
100	N(10, 3)	2.83
100	U(1, 7)	10.48
100	U(0, 20)	9.36

rioration in the accuracy of the estimates of δ. Secondly, the robustness of benefits shown in Figure 7 is based upon unbiased independent and identically distributed noise added to δ_{ij}. It is conceivable that the error characteristics of our estimate might introduce intertemporal distortions that affect prices, the data, and the estimates.

To address these issues, we integrated the delay cost estimation procedure into our simulation code. We estimated the mean of δ_{ij} on the fly,[11] used this estimate to set prices, and then used the resulting user-choice data to update the estimates of δ. In this approach we start with a uniformly distributed cumulative histogram and update this histogram, as described in "Estimation of Delay Cost Factors," over the subsequent update periods. However, in an ongoing environment the effect of new arrivals on the cumulative distribution will be minimal (after a certain short period of time) and any underlying time-varying changes in delay cost distributions will not be detected for a long time. For example, if the historical histogram has arrivals of the order of millions and the new arrivals during a period are of the order of hundreds, then the effect of new arrivals on the cumulative histogram would be minimal. Therefore, in order to permit adaptation to time-varying changes in delay cost distribution, the cumulative histogram is scaled to a factor equal to the total number of new arrivals. For example, if the cumulative histogram has a total of 10,000 observations and the current-period histogram has 1,000 new observations, then the cumulative histogram is rescaled (in appropriate proportions) to have 1,000 observations for the Bayesian update. In other words, we provide equal weights to the frequency distributions in the current-period histograms and in the cumulative histograms before merging them.

Table 2 presents the mean absolute relative deviation of the estimated delay cost factor values from the integrated procedure. As the table shows, the difference between the deviations of estimates obtained by the integrated approach and those from Table 1 are of the same order. The results of Table 2 lead to the conclusion that the prices and delay cost estimates are not distorted significantly by using the integrated on-the-fly delay cost estimations.

In order to investigate the robustness of benefits while using estimated δ in computing prices, we compared the system-wide benefits between two sets of simulation runs: (1) when the customers' delay cost factors are known exactly, with the delay cost factor for a particular job drawn from a particular distribution, as shown in Table 2; and (2) when the average delay cost factors are estimated based on users' observed choices using the Bayesian approach and used for computation of prices. Figure 8

Table 2. Mean Absolute Relative Deviations of δ Estimates Generated On the Fly

Load	Underlying Distribution	Relative Deviation (%)
50	N(4, 1)	2.70
100	N(4, 1)	0.54
200	N(4, 1)	1.20
100	N(10, 3)	2.94
100	U(1, 7)	9.31
100	U(0, 20)	6.97

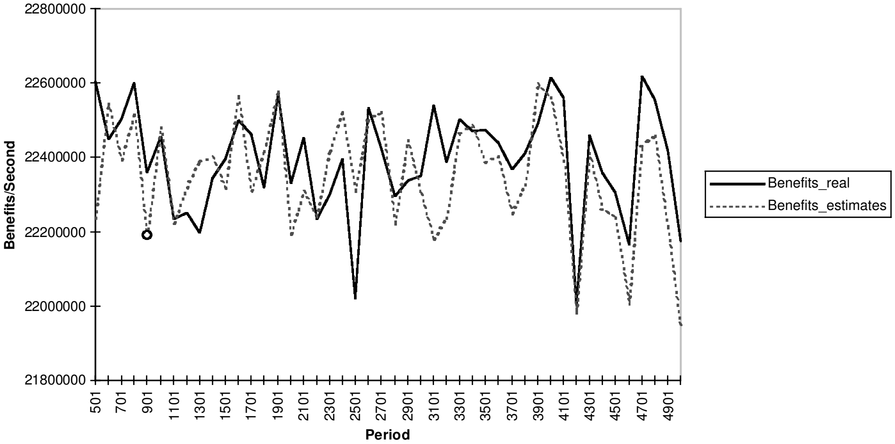

Figure 8. Comparison of Benefits with a N(4, 1) Distribution and Demand of 50

presents the results of this comparison where the true delay cost factors are generated from a N(4,1) distribution, and the demand is 200 requests per second. The results in Figure 8 show that the two systems behave almost identically.

We tested the effectiveness of our approach using other distributions as well. Table 3 summarizes these results using loss of efficiency as a metric to test the effectiveness of our estimation technique. The loss of efficiency is defined as:

$$\text{Loss in Efficiency} = \frac{\text{Benefits with Exact Information} - \text{Benefits with Estimated Information}}{\text{Benefits with Exact Information.}} \qquad (4)$$

As discussed in "Network Externality Pricing and Incentive Compatibility," the objective of this research is to develop a computationally incentive-compatible pricing mechanism that can support multiple-priority classes. In case of multiple-priority classes, users with different ranges of delay cost factors partition themselves, appropriately, into the priority class that is best for them. This partition is achieved by setting appropriate prices such that consumers with different delay costs choose the priority class that is optimal for them. For example, in Figure 9 consumers with low

Table 3. Loss in Efficiency for the Test Cases

Load	Underlying Distribution	Loss in Efficiency (%)
50	N(4, 1)	0.58
100	N(4, 1)	0.26
200	N(4, 1)	0.10
100	N(10, 3)	0.33
100	U(1, 7)	0.85
100	U(0, 20)	2.85

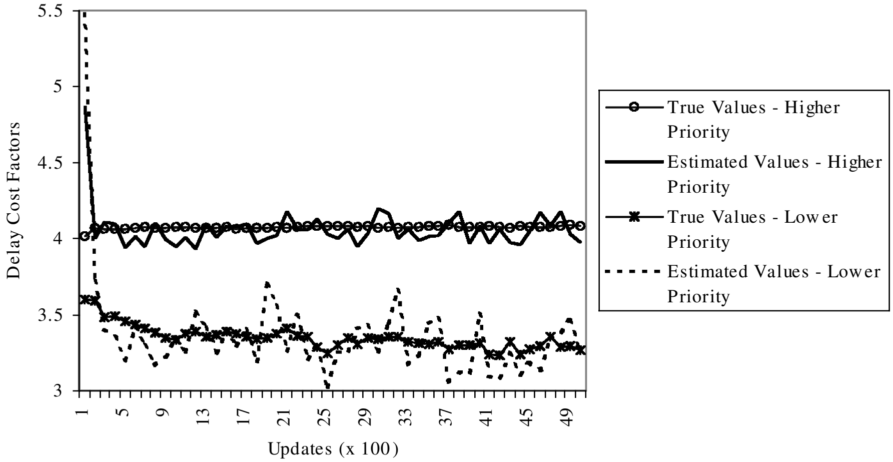

Figure 9. Delay Cost Factors with Two Priority Classes

delay cost factors choose lower-priority since their costs are lower in that priority class. However, consumers that have higher delay cost factors face a higher cost in the lower-priority class because of higher delay and, thus, higher total delay costs.

The estimation technique has to replicate the partition to maintain the effectiveness and benefits of multiple priorities. Note that, in terms of estimation, each priority class at a server just requires maintenance of histograms for that priority class. If the delay characteristics of customers in different priority classes are different, then different histogram values and thus different means would be computed for these customers. Figure 9 shows that such a partition is effectively achieved while using true delay cost factors with two priority classes during a simulation run with an arrival rate of 100 and the partition achieved while using the estimation technique. As the figure shows, the estimation technique provides effective partition by following the true delay cost parameters when on-the-fly estimations are used. The efficiency loss due to these estimates, with respect to net benefits, is less than 1%. These results give us confidence that computationally incentive-compatible mechanisms can be developed and implemented for Internet pricing and other distributed and client-server applications.

Conclusions

MOST OF THE CURRENT RESEARCH IN INTERNET PRICING MECHANISMS assumes that the service providers know users' demand characteristics. This is not a realistic assumption because in stochastic environments it is not incentive-compatible for the users to state their preferences directly. The only observable factor in a real network is users' choice behavior. In this paper, we investigated whether demand characteristics could be estimated from observing the users' choice behavior, and whether such estimates could be used in pricing without much loss in efficiency. Indeed, demand estimation is vital to any pricing mechanism design intended to manage network traffic, and delay cost is but one attribute among several of the consumer demand.

The estimation of delay cost factors is the first step in demonstrating that a set of demand traits, such as the value for quality and timeliness of digital products, can be estimated feasibly by monitoring user choices. This step is crucial for corporate intranets as well, in that an optimal allocation of network resources to achieve organizational objectives would require the knowledge of comparative value of the resources for each individual entity in the organization.

In this paper, we tested a unique nonparametric estimation technique for estimating the delay costs, that ensures that the estimates are independent of the type of the underlying distribution. In future research, we will enhance our simulation models to compute several δ values in each period for different types of services, since it is likely that users of different services have different delay characteristics. However, the results presented in this paper are very encouraging and indicate that our estimation technique performs well, with minimal loss in efficiency with respect to system benefits. The results also indicate that the estimation technique provides necessary separation in delay cost factors associated with different priorities. This is an important characteristic, which is required to maintain appropriate incentives for users to choose the correct priority for their requirements.

NOTES

1. As discussed earlier, such problems exist in all externality pricing mechanisms. However, for clear exposition we concentrate on one model.

2. A server could be a router, information server, or information synthesizer.

3. We assume a FIFO queuing discipline with nonpreemptive priorities. Nonpreemptive priorities indicate that a job in service is finished first even if a higher-priority job arrives in the queue.

4. We do not apply the term "user" to every individual who uses the network, but to relatively homogeneous groups of individuals connected to the network backbone through an access point: a business, educational institution, government office, group of residential users through their access provider, etc., where each group creates a continuous flow of requests.

5. These schemes may be statically available for well-defined services or may be constructed by a software agent when the need arises.

6. Total delay cost is computed by multiplying δ by the expected time it takes to deliver a service.

7. Such as HP's OpenView™or IBM's NetView™.

8. Net Benefits = Cumulative Instantaneous Value – Delay Costs, and Consumer Surplus = Net Benefits – Price Paid.

9. For expositional simplicity we have dropped the priority class index here. However, the analysis can easily be extended to priority classes.

10. Relative absolute deviation = absolute(actual – estimated)/actual.

11. Estimation on the fly means that the delay cost estimates used to compute prices are generated using the Bayesian procedure described in "Estimation of Delay Cost Factors" during a particular simulation run. Traditionally, the performance of an estimation procedure is conducted by running a simulation to collect data and estimate the parameters, and then another simulation is performed using the estimated parameters.

References

1. Braun, H.W., and Claffy, K. Post-NSFNET Statistics Collection. *National Information Infrastructure White Paper* (1995). Available at http://www.nap.edu/readingroom/books/whitepapers/ch-11.html.

2. Choi, S.; Stahl, D.O.; and Whinston, A.B. *The Economics of Electronic Commerce.* Indianapolis: Macmillan, 1997.

3. Deering, S., and Hinden, R. Internet protocol, version 6 (IPv6) specification. *Technical Report,* IETF (1995). Available at http://globecom.net/ietf/rfc/rfc1883.html.

4. Gupta, A.; Jukic, B.; Li, M.; Stahl, D.O.; and Whinston, A.B. Estimating Internet users' demand characteristics. Paper presented at *Fifth International Conference of the Society for Computational Economics,* Boston, MA, June 24–26, 1999. Available at http://fmwww.bc.edu/DEF99/doc/pgm.html.

5. Gupta, A.; Stahl, D.O.; and Whinston, A.B. Managing the Internet as an economic system. Technical Report. University of Texas at Austin (1994). Available at http://cism.bus. utexas.edu/ravi/pricing.ps.Z.

6. Gupta, A.; Stahl, D.O.; and Whinston, A.B. Economic issues in electronic commerce. In R. Kalakota, and A.B. Whinston (eds.), *Readings in Electronic Commerce.* Reading, MA: Addison Wesley, 1996, pp. 197–227.

7. Gupta, A.; Stahl, D.O.; and Whinston, A.B. A stochastic equilibrium model of Internet pricing. *Journal of Economic Dynamics and Control, 21* (1997), 697–722.

8. Gupta, A.; Stahl, D.O.; and Whinston, A.B. Priority Pricing of Integrated Services Networks. In L. McKnight and J. Bailey (eds.), *Internet Economics.* Cambridge, MA: MIT Press, 1997, pp. 323–352.

9. Hausman, J.; Hall, B.H.; and Griliches, Z. Econometric models for count data with an application to the patterns–R&D relationship. *Econometrica, 52,* 4 (1984), 909–938.

10. IC Editors. Bob Metcalfe on what's wrong with the Internet: it's the economy, stupid. *IEEE Internet Computing, 1,* 2 (1997), 6–17.

11. Lederer, P.J., and Lode, L. Pricing, production, scheduling, and delivery-time competition. *Operations Research, 45* (1997), 407–420.

12. Lode, L., and Lee, Y.S. Pricing and delivery-time performance in a competitive environment. *Management Science, 40* (1994), 633–646.

13. MacKie-Mason, J., and Varian, H. Pricing the Internet. In B. Kahin and J. Keller (eds.), *Public Access to the Internet.* Englewood Cliffs, NJ: Prentice-Hall, 1995, pp. 269–314.

14. Mendelson, H. Pricing computer services: queuing effects. *Communications of the ACM, 28,* 3 (March 1985), 312–321.

15. Mendelson, H., and Whang, S. Priority pricing for the M/M/I queue. *Operations Research, 38,* 5 (1990), 870–883.

16. Naor, P. On the regulation of queue size by levying tolls. *Econometrica, 37* (1969), 15–24.

17. Shenker, S. Service models and pricing policies for an integrated services Internet. In B. Kahin, and J. Keller (eds.), *Public Access to the Internet.* Englewood Cliffs, NJ: Prentice-Hall, 1995, pp. 315–337.

18. Shenker, S.; Clark, D.; Estrin, D.; and Herzog, S. Pricing in computer networks: reshaping the research agenda. *Journal of Telecommunications Policy, 20,* 3 (1996), 183–201.

19. Stahl, D.O. The inefficiency of auctions in dynamic stochastic environments. Paper presented at DIMACS Workshop on Economics, Game Theory, and the Internet, Rutgers Univer-

sity, New Brunswick (April 18–19, 1997). Can be obtained from the author by e-mail at stahl@eco.utexas.edu.

20. Vickrey, W. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance, 16* (1961), 8–37.

21. Wang, Q.; Peha, J.; and Sirbu, M. The design of an optimal pricing scheme for ATM integrated-services networks. In L. McKnight and J. Bailey (eds.), *Internet Economics.* Cambridge, MA: MIT Press, 1997, pp. 353–376.

22. Westland, J.C. Congestion and network externalities in the short run pricing of information systems services. *Management Science, 38* (1992), 992–1009.