

MEASURING AND PRICING QoS

S. Faizullah* and I. Marsic**

Abstract

The authors study a scalable pricing framework for QoS enabled networks that support real-time, adjustable real-time, and non-real-time traffic. The scheme, which belongs to usage-based methods, is independent of the underlying network and the mechanisms for QoS provisioning. The framework is credit based, ensuring the fairness, comprehensibility, and predictability of usage price. It also provides a means for the network providers to ensure, with high probability, cost recovery and profit, competitiveness of prices, and encouragement of client behaviours that will enhance the network's efficiency. This is achieved by appropriate charging mechanisms and the provision of suitable incentives. Simulation results suggest that users have better overall satisfaction, that providers are able to recover costs, and that better network utilization is achieved while reduced call blocking probability is observed. The implementation and usage costs of the framework are low.

Key Words

Pricing, quality of service, utilization, call blocking probability, user satisfaction, revenue

1. Introduction and Related Work

Ever since its inception, the Internet has seen unprecedented growth. The characteristics of the Internet are rapidly changing in many dimensions. It has made a remarkable transition from a research testbed, which was available only to a small society of aware and selfless users, to a commercial enterprise. The scope and number of Internet services and applications are also growing rapidly. These services and applications could consume widely differing amounts of resources. Some common examples of earlier Internet services/applications include Telnet, FTP, e-mail, Gopher, and WWW, with few real-time audio and video services such as MBONE (multicast backbone). More recent services include telephony, audio- and video-conferencing, distance learning (DL), interactive games, distributed interactive simulation activities such as tank battle simulations, remote visualization, VoD (video on demand), video e-mail, computer-based FAX, virtual reality, exchanging of medical records, exchanging of experimental weather maps, and wireless personal digital assistants

(PDAs). Businesses, research, and social institutions will be greatly utilizing these services for interactions. Consequently, the future Internetworks is expected to facilitate communication requirements for a wide variety of services.

In addition, the number of users, sites, and hosts has grown dramatically. Also, in the past few years, access to information on the Internet and Intranets has become easier and more user friendly and highly data intensive. The result of availability of user-friendly interfaces, faster PCs and hosts, and high speed of access (in gigabit/sec), coupled with higher awareness of these services, has already been creating congestion problems on the Internet for some time now.

The sources of funding are also changing. Internet was funded by federal government agencies; individual users have not been charged for their use of networks, and have not generally been aware of the impact of their use on network performance. When the U.S. government decommissioned the NSFNET in 1994, alternative sources of funding became a necessity [1-15]. The pricing structures adopted since then are *ad hoc*, complex, and based on unrealistically simplified assumptions (about underlying network, charging/pricing policies, etc.) [2, 3, 5].

Traditional pricing strategies are usage insensitive and prices are relatively low. These pricing schemes are either free (subsidized through government funds) or flat rate for unlimited usage. These strategies were simple and worked reasonably well, as the bulk of the costs are borne by government and respective organizations with little contribution from users. Some variations have emerged that price bandwidth of the connection, or charge flat rate to a certain number of hours and per-hour charges thereafter. These charging mechanisms are crude, inefficient, and, with the Internet's establishment as a commercial enterprise, inadequate.

Moreover, the greatly increased usage of the Internet and the resultant performance degradation have focused attention on the inefficiencies of the traditional pricing structure as well as necessitated a renewed focus on the research to improve hardware, software, and protocols. Although there have been dramatic and outstanding successes in infrastructure research, resulting in high bandwidth backbones having gigabit/sec transfer capability, widespread availability of PCs, easy network connections from homes, faster routers, and sophisticated protocols, there has been a critical vacuum in pricing research.

In the remainder of this section, we briefly discuss some of the relevant existing approaches for pricing services in

* Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA; e-mail: sfaizull@paul.rutgers.edu

** Electrical and Computer Engineering Department, Rutgers University; e-mail: marsic@caip.rutgers.edu
(paper no. 202-1575)

today's Internet as well as establishing the shortcomings of these traditional pricing mechanisms.

Since traffic demands increase as bandwidth (and other resources) improves, it is a mistake to argue that overprovisioning the capacity is the only solution for achieving high network performance [16]. Repeatedly, history has shown that creative application developers and researchers invalidate the claim of limitless bandwidth. Also, there is always potential for bottlenecks even in the case of ample capacity. Furthermore, the unpredictable and bursty nature of traffic sources can cause congestion problems even in lightly loaded systems. Efficient pricing mechanisms coupled with traditional congestion control protocols are the ultimate solution to congestion control, which will result in better overall network performance. These pricing mechanisms are based on user incentives, for example, performance versus monetary as well as administrative, that seem to be the answer to the challenges posed by upcoming Quality-of-Service (QoS) enabled Internetworks.

Researchers have recently focused on pricing a network that offers heterogeneous services. Following the early and important work by Cocchi *et al.* [2], a number of authors have worked on the issue of pricing with different emphasis and objectives [3, 4, 7-13, 17, 18]. Following is a brief summary of few representative schemes of these studies. We have compiled a comprehensive survey in [9-13], to which interested readers are referred.

Cocchi *et al.* [2] have studied the pricing of a single reservation-less network that provides multiple services at different performance levels (four classes only). Each user is characterized by a utility function and can request a Type of Service (ToS) by setting bits in its packets. They showed that in comparison with flat-rate pricing for all services, pricing based on performance objectives enables every customer to derive a higher surplus from the service. At the same time, it generates greater profits for the service provider. Hence it is reported that quality-sensitive pricing is more efficient than a flat pricing scheme.

However, the scheme is not handling a more general case of Internetworks that enable and guarantee QoS. Also, because resources cannot be reserved, users may suffer QoS degradation. In this case, the only way to measure goodness of the pricing scheme is to measure the net satisfaction from the network, that is, the user utility functions have to be assumed. Because of the difficulty in determining a valid user utility function, this is undesirable. Therefore, this scheme (otherwise an important and elegant one) does not seem appropriate for reservation-oriented networks, where QoS guarantees can be made. Furthermore, their method is computationally expensive, as it is based on a fine-grain per byte accounting. Finally, in the optimal pricing models, the fact that different applications may have different performance objectives was usually not considered.

Parris *et al.* [15] also presented a pricing scheme, which studies a host of important issues in connection with pricing (peak/off-peak traffic, elasticity of users' demand, call blocking, etc). However, they assume that the network provides only two classes of services (both requiring some fixed amount of bandwidth to be reserved for the connection). This scheme only considers one resource of

real network, namely link bandwidth. In reality, any successful scheme must consider a host of resources. Also, a single-node network model is an oversimplification in any pricing model.

Sairamesh [14] proposed a method that provides few fixed QoS levels that are specified by a set of parameters such as packet loss probability and average delay or packet loss probability and maximum and average delay. Connections utilize/purchase a service identified by the QoS level. This scheme is restricted due to the fact that certain flows cannot be accommodated in the setup of fixed QoS levels, because of the nature of the applications and their bursty traffic characteristics, and hence the required quality is not delivered, which results in decreased user satisfaction. Any attempt to classify applications into a fixed number of classes or impose fixed QoS levels is restrictive and bound to be unable to encompass a broad set of applications, with traffic characteristics unknown or indescribable with currently known traffic characterization mechanisms. New applications with new traffic characteristics will also not be covered. The scheme will therefore fail to scale.

The research in this field is still in its early stages, and formulating a strategy in the face of network and usage dynamism is a challenging task. We will use the simulation and analytical results from our proposed scheme to compare and contrast the strength and robustness of our scheme to the approaches presented in this section.

1.1 Integrated Multiservice Internetworks

As a result of the rapid development of network technology, it is becoming increasingly efficient to provide different telecommunication services through one integrated-services network instead of multiple single-service networks. In order to support these services, the public data networks will have to be transformed from best-effort to QoS capable and secure Internetworks (e.g., upcoming Integrated Service Internet consisting of a collection of domains as depicted in Fig. 1).

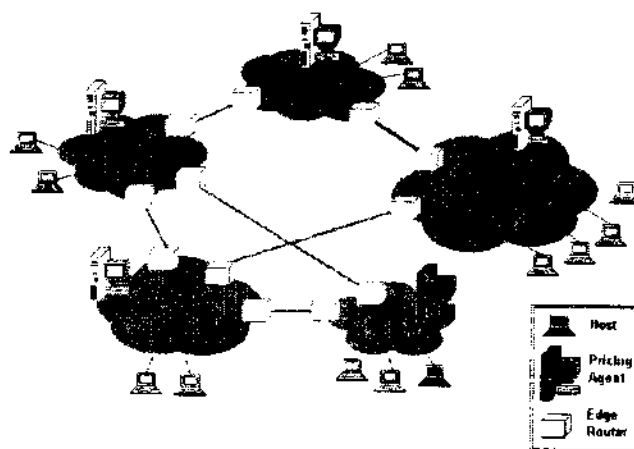


Figure 1. Federated multi-domain QoS-enabled Internetworks. Each cloud represents a domain, an independent entity. Only the edge routers are shown in the figure.

For this reason, this work is based on the assumption that the underlying network is QoS capable, and that the QoS is negotiated at connection setup [19–22]. Each router has buffers for each outgoing link. Appropriate resources are allocated to connections that are admitted. Users can make requests with degraded QoS parameters so that their requests are accepted in case their current request is rejected due to network overload. Alternatively, the network can make suggestions to users about QoS parameters that can be accepted. Users can also renegotiate, during the connections, their QoS parameters [23, 24]. We assume that users desiring best-effort level services can present their unrestricted QoS requirements and the network will accept the connection, and such users will be treated according to the network policies laid down for this category of service. We also assume that the underlying network employs policing and shaping mechanisms to enforce the network policies and guarantee the delivery of QoS. We do not state (and for that matter do not care about) how these capabilities are provided or how the negotiated QoS parameters are guaranteed. This makes our pricing framework independent of network infrastructure and hence one that can be deployed in any such networks that will emerge as Future QoS-Enabled Internetworks.

1.2 Challenges in Pricing QoS

As mentioned earlier, it is expected that in the very near future integrated QoS capable networks will emerge that provide a variety of transmission services, such as telephony, video, and file transfer.

In Future Internet, the issue of pricing is more relevant and critical than it is today. Without appropriate pricing, every future user can and will opt for highest QoS available, thus creating a huge congestion problem—thus the role of incentives. Lack of such pricing will also result in a high rate of rejection of call admission requests.

Note that, unlike with the traditional Internet, there are great differences among the services offered by the QoS network. Therefore, one might ask whether the prices of these services should also differ, and if so, how. Also, it is trivial that each service class specifies more than one QoS parameter, such as average delay, bandwidth, and packet loss probability. When this is the case, the question remains how to price each of such service classes. For example, average delay = 1 ms and bandwidth (denoted as BDW) = 4 Mbs versus avg. delay = 2 ms and BDW = 8 Mbs or 1% packet loss probability and 400 ms avg. delay versus 2% packet loss probability and 200 ms avg. delay. How to price them? What is the appropriate ratio for each QoS parameter? Which should cost more? Should the requirement for security be priced too? Integrating multiple services into a single network generates economies of scale; however, heterogeneous services complicate pricing decisions. Pricing issues in Internetworks that offer heterogeneous QoS services has been the research focus of a number of authors [1–4, 9–13, 14, 15].

The rapidly changing Internet characteristics make it necessary to devise new and improved pricing frameworks suitable for multiservices QoS enabled environment. The

framework will additionally provide proper interface to users to find out the price of communication between two hosts at different times of day given the QoS parameters. It will also provide a standard interface for requesting connection establishment and tear-down. Accounting, billing, and other price-related aspects will also be handled by this system.

1.3 The Proposed Pricing Framework

In this work, we have proposed, analysed, and evaluated a computationally inexpensive and scalable pricing framework for QoS enabled Internetworks [9–13]. The framework, which belongs to usage-based methods, is independent of the underlying network, and the mechanisms for QoS provisioning. The framework is credit based, ensuring fairness, comprehensibility, and predictability of the usage cost. On the other hand, it provides a means for network providers to ensure, with high probability, cost recovery and profit, competitiveness of prices, and encouragement of client behaviours that enhance the network's efficiency. This is achieved by appropriate charging mechanisms and suitable incentives. This charging mechanism can be made sensitive to class of services, to the time of usage (such as peak and nonpeak times), to the network conditions (congested and noncongested), and other situations. We have evaluated users' satisfaction, expressed using the utility functions, network utilization, and measured call blockings through a series of simulations conducted on three network configurations with increasing complexities.

1.4 Paper Outline

Section 2 discusses testbed network infrastructure. Our method for quantifying and pricing QoS is discussed in detail in Section 3. Simulation results are presented in Section 4. Section 5 presents conclusions, explores issues still to be tackled, and describes possible extensions of this work.

2. Testbed Network Infrastructure

In this section, we describe the simulated network infrastructure testbeds, ranging from a single bottleneck link to a very highly complex network representing a real-life scenario. We base our pricing scheme on QoS enabled Internetworks supporting real time, adjustable real time, non-real time, and best-effort traffic. We consider bandwidth, delay, delay jitter, and packet loss probability as QoS parameters in this work. The network configurations used in studying the performance of our pricing mechanism utilize admission control and employ scheduling, queuing mechanisms, policing, and shaping functions to ensure the traffics conform to contracts.

In conventional QoS models, the required bandwidth or the BitRate has no rigid relation to the PacketRate. However, all the forwarding, and hence the pricing, takes place based on packets. Therefore, we choose to use PacketRate instead of BitRate as the QoS parameter. All other QoS parameters also consider a per-packet behaviour (e.g.,

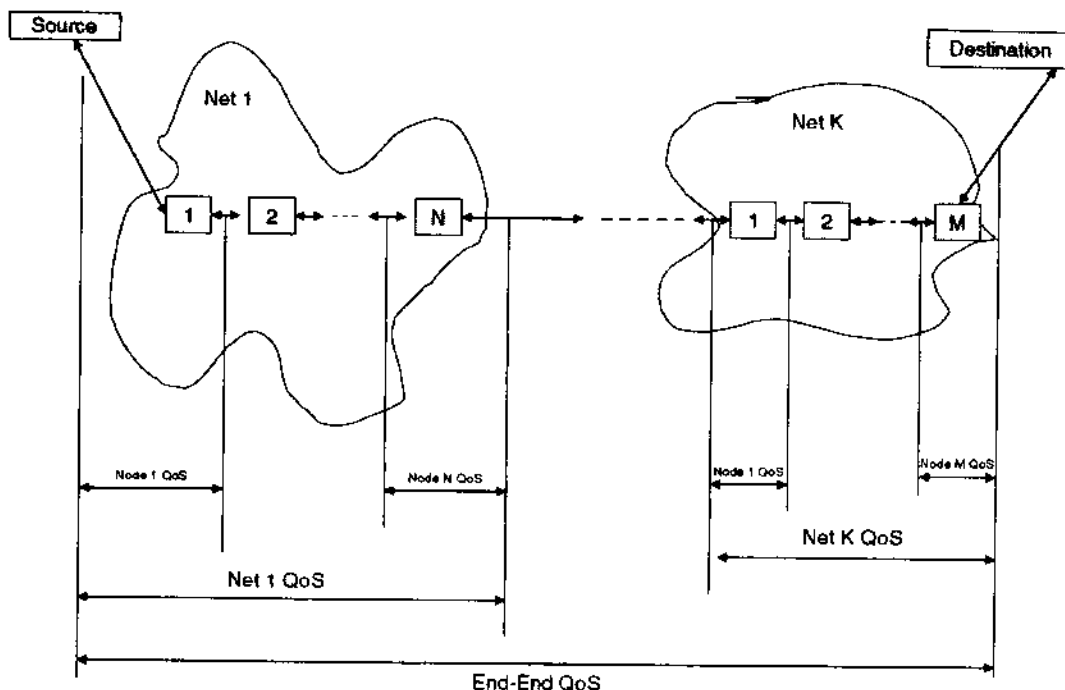


Figure 2. Reference model for Internetworks/ATM.

Delay, Jitter and LossProbability). Hence, considering PacketRate instead of BitRate will help relate all the QoS parameters. The principle of our pricing framework is the same even if BitRate is used during the call admission phase.

The packet size is fixed for constant bit rate (CBR) service and is variable for variable bit rate (VBR) service. The users/applications need to declare the mean packet size (and possibly the standard deviation, SD, which would be helpful in QoS-Routing mechanism) in advance.

2.1 Quality of Service

Several definitions of quality of service exist in the literature: perhaps the most adequate is given in ATM terminology, which states that QoS is the performance observed by an end user. The main QoS parameters are loss, delay, and delay jitter. Fig. 2 depicts a reference model for QoS in ATM on an end-to-end basis—the perspective more relevant to an end user. This model generally consists of one or more intervening networks (Fig. 1); each of them in turn may have multiple nodes. Each of these networks introduces loss, delay, error due to multiplexing, error due to switching, or error due to transmission, and thereby affects the QoS. In addition, statistical variations in the offered traffic may result in loss due to buffer overflow on links connecting congested networks nodes in the QoS-enabled Internetworks. ITU-T Recommendation I.356 [25] takes the approach of defining a worst-case concatenation of networks and devices for specifying the QoS. As long as concatenation across networks stays within these bounds, the users will experience a consistent level of QoS.

It is not a trivial task to select precise values for QoS parameters such as delay, delay jitter, and loss. One

common approach is to group applications with similar QoS requirements into broad generic classes and then specify the parameters for these classes. ATM protocols allow applications to specify QoS in two ways, namely through the generic service categories (such as CBR, rt-VBR, nrt-VBR, and UBR), or through explicit enumeration of the QoS parameters in signalling messages. There are other dynamic and predefined QoS in ATM networks. Readers can explore these topics elsewhere [19, 22], as they are not discussed further here. RSVP [20], on the other hand, defines a means to request and confirm specific bounds on delay variation for the Guaranteed QoS Integrated Services. The Internet standards presently assume that loss is always low for applications given the preferred levels of quality.

2.2 Network Setups

In order to study the robustness of our pricing mechanism, we conducted extensive simulations on a wide range of increasingly complex network configurations. The first configuration (referred to as Configuration 1) is given in Fig. 3. It is a simple network setup consisting of six bottleneck links. In Configuration 1, one of the bottleneck links (in this case R3-R4, Fig. 2) was studied. In order to study the scalability of our scheme, we also conducted experiments on a highly complex network configuration (Configuration 2), shown in Fig. 4. Here also, a single bottleneck link was randomly chosen and studied. The pricing agents, discussed in great detail in Section 3.7, are attached to edge routers only: core routers do not have any pricing agents attached to them.

Each router has buffers for each outgoing link. We employed a work-conserving weighted round-robin scheduling

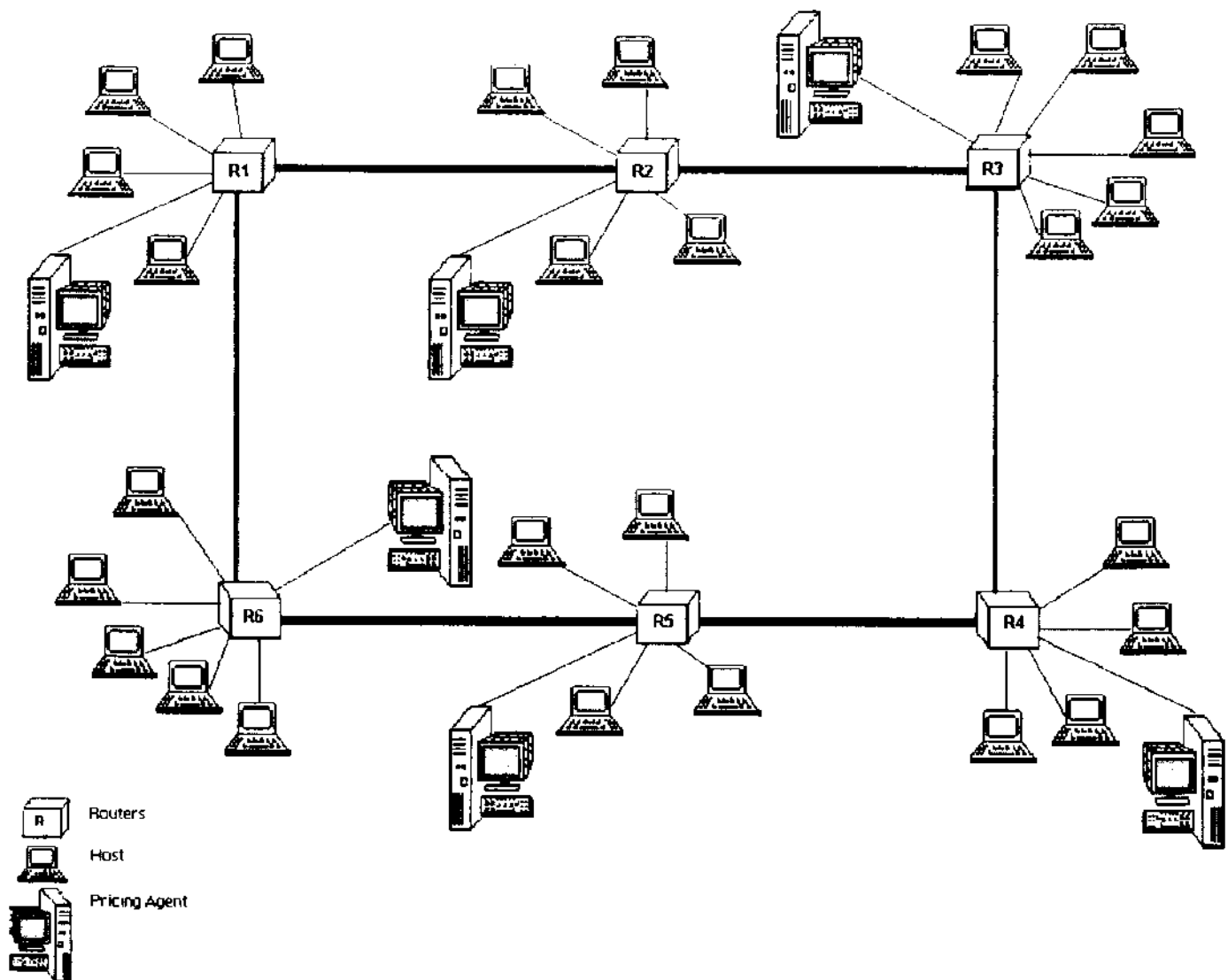


Figure 3. Configuration 2, six bottleneck links. All the routers are edge routers.

scheme. Bottleneck links, in both configurations, connect two routers and are 2Mbps with 10 ms of propagation delay. All other links, which connect hosts to routers, are 10Mbps with propagation delay of 1 ms.

3. Quantifying and Pricing QoS

In this section, we provide the details of our proposed pricing QoS. This method of quantifying and pricing is the basis for the charging mechanism, which is the subject of Section 3.6. It is a simple method yet covers the most important aspects of a practical pricing framework. It is scalable framework for QoS enabled multidomain Internetworks. The Internetworks support real-time, adjustable real-time, and non-real-time traffic. We consider bandwidth, average delay, delay jitter, and packet loss probability as QoS parameters in this work. The scheme, belonging to usage-based [2, 8] methods, is independent of the underlying network and the mechanism for QoS provisioning, and it can be deployed in any QoS enabled environment where best-effort is one of the available classes.

The scheme is credit based, ensuring fairness (from the user's point of view), comprehensibility, controllability, predictability, and stability. On the other hand, it provides a means for network providers to ensure cost and profit recovery, competitiveness of prices, and encouragement of client behaviours that will enhance the network's efficiency. This is achieved by appropriate charging mechanisms and suitable incentives.

The main problem in pricing of Internetworks services is to find a way that fairly quantifies and represents the relative merit of each service. In this work, we present a novel technique that not only includes the definition of such quantification but also is used as a basis for a charging mechanism. The technique also covers the practical issues of its implementation in the existing or future QoS-enabled Internetworks.

3.1 Assumptions

In this section, we state the assumptions made in this work. Also, for comparison with other related work in

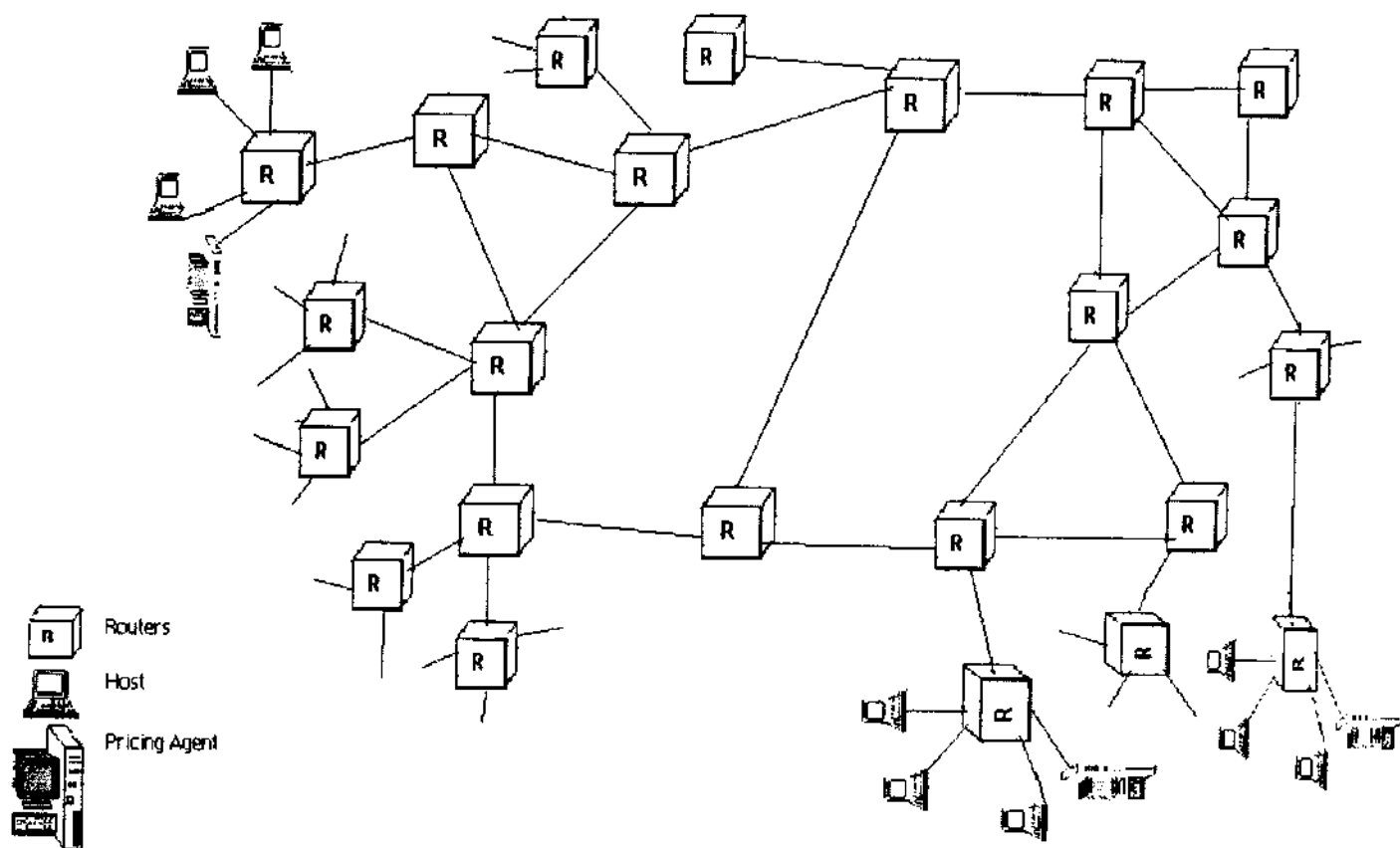


Figure 4. Configuration 3, multiple bottleneck links. Both edge and core routers are present.

this area [2-8, 14, 15, 26-31], and when relevant, we state whether we relaxed any of the assumptions made by other researchers.

The first assumption is that the underlying network is QoS enabled, and that QoS are negotiated at connection setup phase.

We assume that a call will be rejected if the requested delay (one of the QoS parameter) is less than the propagation delay—where we assume that the propagation delay is a standardized delay metric between two routers and is independent of the path taken. Enforcing this assumption is not part of our pricing scheme; rather, it should be handled by the call setup mechanism.

In order to maximize profit (though not necessarily revenue), the routing mechanism should be smart enough to select the route that incurs minimum cost yet delivers the demanded QoS.

In this work we use coarse pricing (for a given session), where accounting is done per packets rather than bytes, which substantially reduces the overhead. A service (like customer service) is available to the users to find out the price of communication between two hosts, given the QoS parameters, at different times.

We use PacketRate instead of BitRate as the QoS Parameter, as routers forward packets and all other QoS parameters also consider a per-packet behaviour (e.g., Delay, Jitter, and LossProbability). Hence, considering PacketRate instead of BitRate will help relate all of the QoS parameters for pricing purposes.

3.2 Types of Service

We categorize the services as falling within the following broad types of classes, each having a different set of QoS parameters. Within each type of service different qualities of service are available by adjusting the QoS parameters [9-13].

Real-Time Service (RT): services that have critical/tight upper bound on the time at which the bits/packets should arrive at the destination. Any data that arrive beyond the estimated time of arrival (ETA) are discarded. Packets delayed are of no use to users. Examples of such services are telephony, teleconferencing, and covering ATM's CBR and rt-VBR.

Adjustable Real-Time Service (A-RT): In this service class, we do not discard data if they are delayed. Instead, the ETA is adjusted as long as the occurrences and durations of these delays and discontinuations are within some acceptable bounds. For example, half-duplex video may be resumed after a short pause due to the delay in a packet's arrival. In this case, we can adjust the acceptable delay parameter, increasing it by the stalled time. Examples of such services are video on demand, video, interactive games, and distance learning.

Non-Real-Time Service: This service is provided for those applications that require certain guarantees but no tight bound of the end-to-end delays; rather, a very loose bound is imposed on the delays. Examples of such service are traditional e-mail and file transfer.

Best-Effort Service: The system supports a fourth service class, without QoS guarantees, which exploits available resources. In our simulations, the background flows were serviced on a best-effort basis.

3.3 Quality Metric

The main reason for complicating the pricing of QoS-enabled networks services is the lack of a method to compare the quality of different types of services. We need a common metric that fairly represents the relative merit of each connection that is characterized by ToS and QoS. Finding such a metric is difficult, as even comparing quality within the same type of service is difficult. For example, comparing two connections of the same type of service but different QoS requirements or different hop counts (or distances) is not straightforward. In other words, we need a *common metric* $f(ToS, QoS)$ such that:

$$f(T1, Q1) > f(T2, Q2) \quad (1)$$

implies user of service $T1$ is more satisfied than user of service $T2$.

The common metric is a good indicator of the price, representing the relative merit of each service. We term this common metric Price ($P(\sigma_{QoS})$), where σ_{QoS} is the set of QoS parameters, such as bandwidth, average delay, delay jitter, packet loss probability, and so on. $P(\sigma_{QoS})$, is also used as a basis for charging.

In this work we formulate a technique that not only includes the definition of such a metric but also covers the practical issues of implementation in existing or future Internetworks.

3.4 Deriving a Common Metric for RT Service

In this section, we derive the metric Price for RT Service class ($P(\sigma_{QoS})$). In order to derive such a metric, for pricing purposes, we reduce the QoS parameter set through a series of redefinitions of these parameters such that $P(\sigma_{QoS})$ involves as few parameters as possible. At the end, we are getting raw indicators, $P(\sigma_{QoS})$, which satisfy the property of the common metric. In what follows we show the treatment of QoS parameters in this redefinition process where “=” means the particular parameter is used as is in the derivation of $P(\sigma_{QoS})$, and “→” means that the parameter in the left-hand side is transformed into the parameter in the right-hand side. The delay, packet rate, packet size, and distance parameters (see below) are only used to derive $P(\sigma_{QoS})$.

Delay Jitter → Delay: Jitter is removed by the use of play-out buffers, which introduces an increase in the end-to-end Delay. In other words, Delay Jitter is reduced into an increase in the end-to-end Delay as follows:

$$\begin{aligned} \text{Revised Delay} &= \text{Acceptable Delay} \\ &+ \text{Acceptable Delay Jitter} \end{aligned}$$

PacketLossProbability → PacketRate: For pricing purposes again, we absorb Packet Loss Probability (PLP) into revised Packet Rate, shown via the following example:

<i>Connection “A”</i>	<i>Connection “B”</i>
PacketRate = 20 P/Sec.	PacketRate = 18 P/Sec.
PLP = 10% (i.e., 2 P/Sec)	PLP = 0%
	BestEffort = 2 P/Sec.

$$\text{Revised PacketRate} = \text{Requested PacketRate} * (1 - \text{PLP}).$$

PacketRate: We charge for any packets that are received in time at the destination. Hence higher PacketRate connections will end up paying more. Call Admission mechanisms should accept only those PacketRates that can be accommodated. Policer does not let the sender, using leaky bucket, send at higher than agreed rates.

Packet Size: Being a usage-based scheme, larger Packet Size means more usage of resources and, therefore, more cost. (i.e., larger PacketSize ⇒ more cost). Price is directly proportional to Packet Size. During a simulation, we assume fixed packet sizes ranging from 53 bytes (ATM cells) to 1500 bytes. For this reason, we use “cell” and “packet” interchangeably.

Delay: Because delay has two components, Queuing Delay and Propagation Delay (in other words, Delay = Queuing Delay + Propagation Delay), by using Standardized Propagation Delay (SPD) with some cushion to cope with larger routes, we define Acceptable Queuing Delay as:

$$\text{AcceptableQueuingDelay} = \text{Delay} - \text{SPD}$$

Requirement of lesser AcceptableQueuingDelay (AQD) means more cost, and hence Price is inversely proportional to the AcceptableQueuingDelay.

Distance: To provide the same QoS, connections between distantly located hosts need more resources than those between closely located hosts. Again using Standardized Distances (in Hops), the network routing mechanism should find smaller routes to save costs. Here, Price is directly proportional to Distance.

Per-Packet Accounting: We compute ETA for each packet based on the Acceptable PacketDelay. Any packet that has arrived within its ETA, satisfies the required QoS and is charged at:

$$\text{Price} = \text{ActualPacketSize} * \frac{\text{Distance}}{\text{AQD}}$$

Any packet slot that is missed by the sender is charged at:

$$\text{Price} = \text{MinimumPacketSize} * \frac{\text{Distance}}{\text{AQD}}$$

The packets that are delayed beyond their ETA are credited at:

$$\text{Price} = \text{ActualPacketSize} * \frac{\text{Distance}}{\text{AQD}}$$

The packets that are dropped due to congestion are credited at:

$$\text{Price} = \text{MeanPacketSize} * \frac{\text{Distance}}{\text{AQD}}$$

As we can see, the pricing framework for each packet that is sent only performs a constant number of simple steps. Also, the framework needs to keep the state for each flow and therefore requires memory. This requirement is very small; only one record is kept per flow. In summary, both time and space complexity are low.

3.5 Deriving a Common Metric for A-RT Service

Recall that A-RT Service does not discard data if they are delayed. Instead, each acceptable pause results in an increase in Acceptable Delay parameter by the stalled time and therefore an adjustment in ETA. In case the frequencies or durations of such delays are violating the agreement, users will be credited using the same method of RT service class.

Noting this difference, we are using the same procedures and formulae of RT service with adjusted Acceptable Delay being recalculated after each acceptable pause. This yields lower overall price for A-RT service class.

As no tight delay bounds are given for non-RT service, the pricing of this service class is simply usage-based such that the packets that arrive within a loose bound are charged. Because AQD is large for this service class, using the same procedures as Section 3.4, users end up paying less.

We formulated a common metric that takes the QoS parameters into account and bases itself on the QoS provided in relation with what the network guaranteed: as such it is a quantification and measuring of this guarantee that this method provides, which is used for charging. This scheme is independent of the underlying network and the mechanism for QoS provisioning, and it can be deployed in any QoS capable environment. It is also a credit-based scheme, hence ensuring fairness.

The metric $P(\sigma_{QoS})$, is used as the basis for our pricing framework, which is the subject of the next section. The charging mechanism uses this metric for billing purposes, and it also provides the basis for the utility functions, which are used to gauge the users' satisfaction—an oversimplification of users' behaviour, but nonetheless a practical approximation.

The time complexity of calculating this metric is constant (for each packet) and the memory requirements are negligible: therefore, our pricing framework is practical, scalable, and conforms to end-to-end argument [32], which fundamentally advocates leaving complexity at the edges and keeping the network core simple.

3.6 Charging Methods

A number of charging-related issues are discussed in this section. The following is a generic charging formula for usage-based pricing where the network administrators set

domain-wide charging coefficients. The formula consists of three components: usage charges, reservation charges, and access charges.

$$C_{\text{traffictype}}(\sigma_{QoS}) = \alpha_{\text{traffictype}} * P(\sigma_{QoS}) + \beta * R(\sigma_{QoS}) + \gamma \quad (2)$$

where:

$C_{\text{traffictype}}(\sigma_{QoS})$:	Cost for traffic type (e.g., RT, A-RT, etc.)
$P(\sigma_{QoS})$:	Common metric
σ_{QoS} :	QoS parameter set (bandwidth, packet loss probability, average delay, delay jitter, etc.)
$R(\sigma_{QoS})$:	Resource reservation charge (may include connection establishment charge)
$\alpha_{\text{traffictype}}$:	Coefficient for usage charges (e.g., α_{rt} is for real time, α_{art} for A-RT, and so forth)
β :	Coefficient for reservation
γ :	Fixed access charge

In this work, we only apply a fixed charge for connection establishment. Also, we assume that the access charge component is not per connection: rather, the access-providing entity can charge this per month, and during the revenue distribution this amount is charged by the billing mechanism.

We also consider the following simplified formulae in some of the simulations conducted in this work:

$$C_{rt/art}(\sigma_{QoS}) = \alpha_{rt/art} * P(\sigma_{QoS}) + R \quad (3)$$

And:

$$C_{nrt/be}(\sigma_{QoS}) = \alpha_{nrt/be} * P(\sigma_{QoS}) \quad (4)$$

where R is a fixed connection establishment charge. These formulae were helpful in keeping our focus on a particular issue, such as the effect of charging and/or fixed connection coefficients, in situations where we did not want to be concerned with other issues.

Note that it makes sense to assume that α_{rt} is higher in value than α_{art} , which in turn is higher than α_{nrt} , with α_{be} being the lowest of all the values. We assume these coefficients are defined by pricing agents—discussed below—and not individual routers that are involved in the connection. Network providers can make these coefficients sensitive to different time periods. For example, in a corporation these coefficients can be made such that individual departments, to conserve virtual-money, delay their batch work to off-peak times or to be triggered by low network utilization. Also, administrators can set different high coefficients between certain links so that the traffic load in a certain portion of the network is kept low. However, as network performance degradation may become apparent in extreme cases, the swing should be carefully designed.

3.7 Pricing Agents

In order for our pricing framework to be scalable, we choose endpoints (edge routers) to be the place for accounting, where dedicated agents receive duplicate headers from corresponding edge routers on the receiving host side. They are only dummy hosts with no overhead to the network. Pricing agents discard the messages after logging the header information. In addition, they can be used to perform other activities, such as coefficients estimations, providing charging information, acting as a call admission mechanism, deciding queue sizes for network providers, and a host of other activities including metering, billing, advertisements, and revenue distribution. They can also bill electronically and receive payments via network.

The pricing framework is implemented in a complete decentralized manner. A handshake is needed between the call admission mechanism of the network and the pricing mechanism so that the latter knows about the QoS parameters (and their values) that the network guaranteed to the user. Also, users can enquire about their accounts. In case the QoS network supports renegotiations [14], this renegotiation is accepted by the network, so that the price handshake needs to take place every time such negotiation occurs, and final, net charge is calculated accordingly. A pricing agent can also be used to implement a renegotiation mechanism such as the one reported in [14].

3.7.1 Network Interface

We use a standard interface to request call admission or inquire about the charging amount, for billing and accounting information. For call admission requests or renegotiations, we use **Call(ID, Sr, Des, ToS, σ_{QoS} , PS, MPS, ...)**. **ID** is the user identification, **Sr** is source host, **Des** is the destination, **ToS** is the type of service being requested, **σ_{QoS}** states the requested QoS parameters, **PS** is packet size, and mean packet size is **MPS**. **Charge(ID, ConnID, Sr, Des, ToS, σ_{QoS})** are used to inquire about the charging amount per unit. **ConnID** is connection ID. Finally, **Bill(ID, [ConnID])** is used to request the billing information and balance for a host or even for a connection of a particular host.

3.8 Utility Functions

We base the definition of the utility functions used in this work on economic theory, which states that given a congested resource, the price one pays to send a message (i.e., its utility) should reflect the loss of utility inflicted on other users whose messages did not get the same treatment. The utility functions need to show the performance one's application gets and the price others pay for [2]. In general, the utility function takes the following form:

$$U(P(\sigma_{QoS})) = L_{traffictype}(P(\sigma_{QoS})) - C_{traffictype}(\sigma_{QoS}) \quad (5)$$

where $L_{traffictype}(P(\sigma_{QoS}))$ is the apparent degradation level (in favour of other users) to a user, and

$C_{traffictype}(\sigma_{QoS})$ is the price one is charged. $U(P(\sigma_{QoS})) - R$ is a mapping from a non-negative real number to non-positive real number showing the utility perceived by a user (for a usage request with quality given by σ_{QoS} and for which cost $C_{traffictype}(\sigma_{QoS})$ was paid).

Following is one set of $L_{traffictype}(P(\sigma_{QoS}))$ for real-time, adjustable real-time, and non-real-time traffic. In this work, best effort traffic was utilized for the background flows, and hence we did not explicitly study the quality of the service they received.

$$L_{rt}(P(\sigma_{QoS})) = \text{number of packets that did not meet ETA} \quad (6)$$

$$L_{art}(P(\sigma_{QoS})) = \text{number of packets that did not meet adjustable ETA} \quad (7)$$

$$L_{nrt}(P(\sigma_{QoS})) = \text{number of packets that did not arrive within a loose bound (generally a few minutes)} \quad (8)$$

User satisfaction can be asserted by using utility functions, given above, for different traffic types. Users prefer no (or less) performance degradation to their application as well as lower costs. These two factors are captured in U . The lower value of U (for a user/application) means more user satisfaction. In this work, U (as well as L) is an average value for several runs of the same application (characterized by the same service class, traffic source, and traffic mix). Traffic sources and traffic mixes utilized in this work are the subject of Section 4.2. To determine whether a user/application is satisfied or not, we compare the computed utility function, U , for that user/application to that of the ideal case of U' that represent the situation of no performance degradation (i.e., where L is zero). A user is satisfied if the difference is not wide (for RT we used lower difference for this purpose than A-RT). Utility functions defined in this section can also be used to compare different pricing schemes (for example, to compare flat-rate versus differentiated pricing). Note, that because our proposed pricing scheme is credit based, there is already a built-in compensation (in terms of less cost) during the derivation of $P(\sigma_{QoS})$. This enables us to compare our proposed scheme with credit-based feature enabled versus the case where we disabled this feature.

3.9 Additional Pricing Components

Metering is another important and problematic issue that needs to be included within the pricing framework. What is needed here is to log traffic by the pricing agents. Moreover, those agents are used to perform other activities such as coefficients estimations, providing charging information, acting as call admission mechanism (not part of our pricing framework), deciding queue size (not part of our pricing framework) for network providers, and a host of other activities including metering, billing, advertisements, and revenue distribution. They can also bill electronically and

receive payments via network. In order for our pricing framework to be scalable, we choose endpoints for pricing-related work and no increased work at the core routers.

In this work, the billing is done per connection and accounting is done on specific time periods (of all completed connections).

Distribution of revenue—like accounting—is done per connections where actual billing will happen in predetermined time periods for all completed connections. As we mentioned earlier, every domain/router can set its charging coefficients, which will be reflected in charging the user, but revenue must be distributed according (proportionally) to the delays in each router. For the most part we used uniform charging coefficients in this work. The routers causing the packets to be delayed must have favoured packets from other connections, and hence it is fair to receive less revenue. Note that the routers only set the coefficients in specific time periods independent of any particular connection and that the pricing agents at the endpoints only do the pricing-related work. Users can probe pricing agents for the values of different coefficients before requesting a connection.

Each domain takes its independent course to set their coefficients; the routing and revenue distribution decisions can become complicated. In this work they are assumed to be uniform. However, in a corporate domain, when pricing is used as regulatory tool and as an incentive-based means, different coefficients can be used for different tasks, between different sites, and during different time periods. If chosen carefully, this tactic could increase efficiency.

Adjustable real time can be useful for sponsored multimedia service where a network provider, in return for some reduced (possibly free) cost, broadcasts (for example by using a tool like realplayer) advertisements to users during pauses.

In multicasting the revenue distribution is more complex and, in addition to being proportional to the delay experienced at routers, is also proportional to the number of receivers depending on them (as more processing work is carried out by the router).

3.10 Implementation and Deployment Issues

Our pricing framework can be implemented in a completely decentralized manner. A handshake is needed between the QoS provisioning mechanism of the network and the pricing scheme so that the later knows about the QoS parameters that the network guaranteed to the user. As we can see, the edge routers route a small number of packets per flow in connection with pricing mechanisms, and the destination edge router copies the header off of each packet and forwards it to the destination pricing agent. Therefore, the computational overhead for the deployment of our pricing framework is low, which enables it to be scalable.

In case the QoS provisioning mechanism supports renegotiations, this handshake needs to take place every time such renegotiation is successfully accepted by the network, so that the price and quality metric (and finally charging) are calculated accordingly. The pricing agent can be used

to implement a renegotiation mechanism such as the one reported in [24].

Note that the routers only set the coefficients in specific time periods independent of any particular connection and that the pricing agents at the endpoints only do the pricing-related work. As we mentioned earlier, we assume that all the routers set the same charging coefficient. However, it is also our goal to simulate networks where routers set different charging coefficients (perhaps by using MPLS-like methods [33]). However, this will also complicate the distribution of revenue because in addition to delays in each domain, the charging coefficients of each domain now have to be considered. In addition, it will make the task of finding optimum path more difficult. Also, using the interface described above, users can inquire about their accounts. Most of these inquiries are handled by the pricing agents.

4. Simulation Results

In this section, we present the results for a large number of simulations conducted on a variety of network configurations and under realistic traffic workload and traffic mix. Most of the simulations included between 24 and 48 flows of which 4 to 8 were background flows (simulating best-effort traffic). We have also conducted simulations with up to 128 flows with 10 to 15 background processes. This number of flows was kept constant for the entire duration of the simulations. Whenever a connection was completed and disconnected another one was generated (after the lapse of a small and random period of time). User requests arrived according to Poisson distribution with a rate of a request per μ minutes, and with the duration of connections exponentially distributed with a mean λ . Users chose between RT and A-RT services classes. This was modelled as a bimodal distribution with a fraction τ of users choosing RT and $1-\tau$ opting for A-RT. Unless otherwise stated, we assume a single domain network, and hence the revenue distribution related issues would not come into the picture. Also, we have used, unless otherwise indicated, 1024 bytes packet size. Here, offered load (load for short) is defined as the ratio between the total amount of bandwidth reserved (for all connections) and total bandwidth of bottleneck. Also, link utilization is defined as the ratio between the total amount of bandwidth required (for all connections) and total bandwidth of bottleneck. Experiments were conducted for at least two hours (most of the simulations were conducted for five hours) with varying flow durations, and results were observed after the lapse of an initial warm-up time of few minutes, until link utilization reached 60%. Data were collected at different load levels. As mentioned earlier, the simulations performed were *trace driven*, that is, the traffic was first generated (for a particular traffic mix) and then the same traffic was applied to each of the simulated algorithms. For each simulation setup, we applied a number (between 10 and 20) of such traces and averaged the results. All the instances of each simulation setup were conducted using the same traffic mix (explained later in this section). We have chosen traffic sources/mix to be bursty as well as non-bursty. Each router has buffers

for each outgoing link. We employed a Work-Conserving Weighted Round-Robin Scheduling scheme [19]. Bottleneck links in all of the configurations connect only two routers and are 2Mbps with 10ms of propagation delay. All other links, which connect hosts to routers, are 10Mbps with propagation delay of 1ms. In each experiment, one of the bottleneck links was studied. Note that once the link was randomly selected in the beginning of a simulation run, that same link was kept under study throughout the experiment runs. Every user has some fixed budget for network communication. Users are classified as rich, middle class, or poor based on their allocated budget. Unless communication is deemed important, users will try to conserve their money and hence will respond to monetary incentives in favour of performance. Users are using budget and incentives available, in terms of reduced charges, to modify their behaviours.

4.1 Scalability

Scalability [19, 34] is one of the most important factors when designing a distributed system. In order to achieve this goal, we designed our framework such that the core and distribution routers were not required to perform any activities needed by the framework, the edge routers were only involved in exchanging a small number of packets generated by the source host, the source and destination pricing agents as well as time stamping the packets on the source side while passing a copy of the headers of packets to the destination pricing agent. The pricing agents on their part perform few simple steps per packet, making the computation complexity linear on the number of packets. In addition, keeping with spirit of the end-to-end argument, the pricing agents are located at the edges of the network [32]. In summary, we kept the computations within pricing agent simple that only require a constant number of steps and constant space for storing state information per flow. We also pushed all the computation to the edges of the network with the bulk of the computations performed by the pricing agent and only a few data exchanges, per flow, carried out by the edge routers. All these design decisions were made to ensure that our pricing framework is scalable.

We have evaluated our framework for scalability by varying the following setups and variable in the simulations:

1. Increase in number of users, represented by the number of flows.
2. Increase in volume of data, to which the simulated networks were subjected.
3. The change in characteristics of data, represented by different traffic mixes.
4. Increase in complexity of the simulated networks, represented by different network configurations.
5. Introduction of other issues, such secure communication into the framework.

4.2 Traffic Sources

The emerging QoS-Enabled Internetwork is expected to support a wide range of traffic sources. It is generally

expected that the various traffic sources to be serviced by the Internetworks belong to the following main classes:

1. *Constant Bit Rate (CBR)*. These sources mainly consist of applications such as voice mail, audio-teleconferencing, or telephony. Voice signals have a constant bit rate of 64Kbps or as low as 4Kbps when the voice is heavily compressed. Audio ranges between 8 Kbps to around 1.3Mbps for CD quality. Some video-compressing standards, such as MPEG1, compress video into CBR (poor quality at 1.5Mbps; good quality at 3Mbps).

For video and voice to be of acceptable quality, delay and loss should be kept low. For example, the end-to-end delay should be less than 200ms for real-time video and voice conversations. The delay can be a few seconds for non-real interactive applications (for instance, interactive video). The maximum acceptable loss is about 10^{-4} for noncompressed video and audio and much less than 10^{-4} otherwise [19, 22, 35].

2. *Variable Bit Rate (VBR)*. These sources mainly include computer-related data sources, such as terminal emulation, electronic mail, or file transfer. The range of this type of traffic can be from a few dozen bits per second to a couple of Mega (even Giga) bits per second.

3. *VBR-Video*. These sources are mainly generated by multimedia applications. Some single-compression techniques, for example MPEG2, convert video into VBR. We have low rate for slow-moving scenes and high for fast-moving ones. The range can be from 6Mbps to 24Mbps or even higher with some other techniques.

1. *Connectionless and Connection-Oriented Data*. Sources of this type exhibit both burstiness and constant rate characteristics.

To study the suitability and robustness of a particular pricing mechanism, the workloads should have some desirable characteristics, such as:

1. The traffic loads will form a variety of common sources that cover bursty and non-bursty scenarios.
2. The traffic mix should provide an approximation of the actual traffic realities and emerging trends.
3. The correlation in space as well as in time is exhibited by the traffic sources. This correlation lasts for the entire duration of the connections and not just for a single burst. The factors forming the correlation are several, including burstiness nature of the sources and the pairing of the input hosts with the output hosts.

4.2.1 Simulated Traffic Source Models

Many scholars have studied the traffic source characterization [35]. For any particular traffic source, the essential characteristics are all those parameters that are needed to completely characterize the randomness in the source. These important traffic parameters are used to develop/simulate a traffic model for the given source.

In the ON-OFF traffic model, throughout the lifetime of a virtual connection, the corresponding traffic source will be in either active (or ON) or idle (OFF) mode. In the former mode, the source is transmitting packets/cells at a given rate. This will be followed, depending on type of the source, by an idle period during which the source will be

silent. Hence the model is known as the ON-OFF model. With the exception of CBR sources that have no OFF state, there is a general belief that all other traffic sources exhibit this cyclic behaviour, depicted in Fig. 5 [19, 35]. Note that the packets generated during the same ON-period form a burst, and it is always assumed that successive active and idle periods are statistically independent. According to the suggestion by the ITU-T [36], the length of the active period as well as that of the idle period is exponentially distributed, with average lengths $1/a$ and $1/b$ respectively, as is given in Fig. 5.

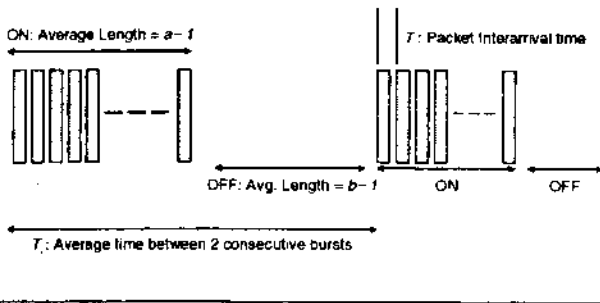


Figure 5. ON-OFF traffic source model.

Several other traffic source models exist, such as the Markov Modulated Poisson Process Model (MMPP) or the Generally Modulated Deterministic Process Model (GMDP) [35]. The ON-OFF model is, however, the least complex and is the most widely used by researchers to model network traffic sources. Also, this basic ON-OFF model is flexible enough to accommodate most of the existing traffic sources with reasonable accuracy and with ease of implementation.

4.2.1.1 ON-OFF Traffic Generation. In order to generate the simulated traffic sources, we, in this work, have assumed that the following three parameters are always known about each such source:

- The average packet rate, m , which is used to calculate the average bit rate, as we assume a fixed packet size (and this is why we use the work cell and packet interchangeably)
- The mean burst length B in packets/cells
- The burstiness factor k

Given the parameters m , B , and k , we can derive all the required parameters that are necessary to implement the ON-OFF model. We have also assumed that the active and idle periods are exponentially distributed (with average lengths of a^{-1} and b^{-1} respectively).

Now from the knowledge of m , B , and k , we will show the process of calculating parameters a^{-1} , b^{-1} , as well as other needed parameters, such as T and p . This is illustrated with the help of the following example about VBR traffic source.

Exponential distribution can be generated from the uniform distribution as follows. Let y be an exponentially distributed random variable with mean equal to $1/j$, and $F(y)$ be the cumulative distribution function of y . Then:

$$F(y) = 1 - e^{-y \cdot j} \quad (9)$$

Let:

$$z = F(y) \quad (10)$$

We can prove that z is uniformly distributed over the interval $(0, 1)$. Because:

$$z = F(y) \quad (11)$$

$$z = 1 - e^{-y \cdot j} \quad (12)$$

$$\Rightarrow y = -\left(\frac{1}{j}\right) * \ln(1 - z) \quad (13)$$

Now because both z and $(1 - z)$ have the same distribution, we have:

$$y = -\left(\frac{1}{j}\right) * \ln(z) \quad (14)$$

By taking $(1/j)$ to be a^{-1} and b^{-1} , we can generate exponentially distributed durations for any source in a given time period for both the active and the idle periods respectively. Finally, using these values, we can calculate the number of packets in the i -th active period by using the equation:

$$B_i = \frac{a_i^{-1}}{T} \quad (15)$$

Lately, we have seen great interest in the development of techniques to generate traffic patterns that exhibit self-similar properties [19, 37]. There are several techniques proposed including a technique involving the modelling of a number of ON-OFF sources and another technique that simulates $M/G/\infty$ queue (with the arrival being Markovian, but the general service time distribution is not). In the ON-OFF technique, the distribution of the times between ON and OFF transitions uses a heavy-tailed distribution, like those distributions modelled by Pareto distribution, instead of negative exponential distribution. We use this model, which has recently drawn the attention of many researchers, to simulate self-similar traffic sources [37].

4.2.2 Simulation Traffic

We have developed software that simulates the traffic sources. Following ITU-T recommendations, all VBR sources follow the aforementioned ON-OFF model, where each VBR source is characterized by three parameters: average cell rate m , average number of cells per burst B , and the burstiness factor k .

It is impractical to conduct simulation for all possible traffic mixes. Instead, we limited ourselves to some of the typical workloads expected by a network. We experimented with the following typical traffic mixes:

Traffic Mix 1: The source consists of 30% Data, 50% Voice, and 20% Video.

Traffic Mix 2: 20% of the sources are Connectionless Data, 20% Connection-Oriented Data, 20% Voice, and 40% VBR Data/Video.

Traffic Mix 3: The sources are as in Traffic Mix 2, but with output concentration, where close hosts are chosen so as to create a scenario of locality.

Traffic Mix 4: In this mix mode, the peak/nonpeak scenario is captured. In this mixing scenario, we increase certain traffic classes by inducing a surge in the number of flows, with average durations of these flows set to the duration of the peak time that is intended to be simulated. This cycle is repeated as many times as the number of peaks desired to be included in the life of a simulation.

Traffic Mix 5: In this traffic mix, in addition to including known traffic classes given the above, we try to capture some of the emerging and futuristic traffic trends. As we move towards high-speed high-capacity (HSHC) networks, we can clearly see that this trend also enables different service architectures. For instance, an HSHC network enables transfer of an entire database to a user for high-speed local searching, versus the current paradigm of query-response applications, such as many web pages do today, which is poor in productivity and introduces long delays. This traffic class is highly bursty and tolerates almost no errors or losses.

Traffic Mix 2 subjects the network to a much larger workload than generated by Traffic Mix 1. The reason is that the second traffic mix has a higher percentage of burstier sources with larger bandwidth requirements. The change in performance behaviour from Traffic Mix 1 to Traffic Mix 2 will expose the sensitivity of the pricing mechanism to the workload. Traffic Mix 3 generates as high a workload as Traffic Mix 2, with the important difference that the level of internal (internal links) as well as external (output hosts) contention is increased. Traffic Mix 4 generates as high a workload as Traffic Mix 2 or 3, but in a piece-wise manner, with each piece having a duration and with different burstiness, so as to simulate the peak/nonpeak scenarios. In essence each peak and nonpeak period itself is Traffic Mix 2 or 3. Traffic Mix 5 generates emerging traffic sources mixed with traditional traffic sources such as Traffic Mix 1, 2, 3, or even Traffic Mix 4.

As mentioned, the interarrival times of the packets generated by different traffic sources are usually exponentially distributed, and hence the traffic can considerably vary from one run of the simulation to the next. Hence, the trace-driven technique is adopted for comparison reasons.

4.3 Traffic Generation

During the simulation lifetime there were between 24 to 128 flows, including some 4-15 background flows. User requests arrived according to Poisson distribution and the duration of connections exponentially distributed. Users chose between RT and A-RT services classes using a bimodal distribution. The traffic generated by a user, after the request is admitted, is modelled by ON-OFF process explained earlier. In order to characterize a traffic source, three traffic parameters were specified: peak packet arrival

rate, average number of packets in a burst, and average time between two consecutive bursts.

We have used typical values for traffic parameters for various traffic source types reported in [35] in order to generate the traffic for our simulations. Given these parameters, we were able to calculate a^{-1} and b^{-1} for ON-OFF model. The traffic sources for N flows consist of N ON-OFF models that are multiplexed to a router. We have conducted simulation for a variety of traffic mixes.

In order to simulate peak-time scenarios, a sudden surge in number of flows is made by triggering a timer. After the lapse of a specified time (a randomly chosen time period), the number of flows are brought back to the normal range as specified above.

In this work, we conducted extensive simulations on a wide range of increasingly complex network configurations. Our earlier simulations were conducted on a simple network setup (referred to as Configuration 1 and given in Fig. 3). It consists of six bottleneck links. This configuration mimics a corporate or campus environment. In our first experiments the single bottleneck link was observed, and in the second experiment one of the bottleneck links (in this case R3-R4, Fig. 3) was studied. In order to study the scalability of our scheme, we conducted experiments on a highly complex network configuration (Configuration 2), shown in Fig. 4. This configuration mimics an enterprise network. Here also, a single bottleneck link was randomly chosen and studied.

4.3.1 Call Admissions

The network that is assumed in this work is capable of negotiating QoS parameters at connection setup, and upon accepting user connection it is responsible for guaranteeing the agreed quality. Based on the admission control scheme, requests are accepted or rejected. Appropriate resources are allocated to connections that are admitted. Users can make requests with less stringent QoS parameters so that their requests are accepted in case their current request is rejected (because of network overload). Alternatively, the network can suggest to users QoS parameters that can be accepted. Users can also renegotiate, during the connections, their QoS parameters [23, 24]. During the connections lifetime users, if needed, can also renegotiate their QoS parameters. We assume that users desiring best-effort-level services can present their unrestricted QoS requirements and the network will accept the connection, and such users will be treated according to the network policies laid down for this category of service. We do not state (and for that matter do not care about) how this capability is provided or how the negotiated QoS parameters are guaranteed. This makes our pricing framework independent of network infrastructure and hence one that can be deployed in any emerging QoS-enabled Internetworks.

The admission control requires that each node check every new request (whether a request for a new flow/connection or renegotiation of a current flow/connection) against available capacity and current QoS capabilities. A request will be accepted only if the network can provide the required QoS to existing flows/connections

Table 1
Typical Emerging Applications and their Costs

Application	Rate	Real Life Scenario	Service	Duration (Minutes)	Cost (\$)
Teleconferencing	1.544 Mbps	Project meeting	RT	60	6.38
Teleconferencing	2.048 Mbps	Formal business meeting	RT	60	11.06
Teleconferencing	3 Mbps	Medical examination	RT	60	16.88
Movie	1.154 Mbps	Entertainment	A-RT	90	4.47
Movie	1.544 Mbps	Entertainment	A-RT	90	6.40
Distance learning	384 Kbps	Live class	A-RT	50	1.73
Web browsing	64 Kbps	Browsing/Surfing	RT	60	0.69
Web browsing	32 Kbps	Browsing/Surfing	RT	60	0.30
Telephony	128 Kbps	Conversation	RT	60	1.38
Telephony	64 Kbps	Conversation	RT	60	0.69
Radio	16 Kbps	Information/Entertainment	RT	60	0.35
Data transfer (50 MB)	N/A	Information	NRT	N/A	0.31
Peer-to-peer media sharing	512 Kbps	Chat	A-RT	60	1.49
Peer-to-peer media sharing	128 Kbps	Chat	A-RT	60	0.70

and to the requesting flow/connection. RSVP, MPLS, Diff-serv, and ATM all have comparable admission control policies, as they all effectively operate in a connection-oriented paradigm [19, 20, 33].

In order to avoid being involved in complex issues of call admission mechanisms (such as those utilizing scheduling region or Connection Admission Control (CAC), which is widely implemented in ATM switches), and to focus on issues of pricing, we utilize a few simple call admission algorithms. The first algorithm is a simple one, where a request for call admission is accepted if network utilization is under $u\%$ (in our simulations we have varied the value of $u\%$ from 95% to 100%) and at least one path with required bandwidth is found. The network can use multipaths, if available.

4.3.2 Charging Parameters

In these simulations we used charging coefficients of 8.328×10^{-4} for RT (i.e., α_{rt}) traffic, and $\alpha_{art} = 4.164 \times 10^{-4}$ for A-RT traffic is used in our experiments. We used $\alpha_{nrt} = 0$ or $\alpha_{nrt} = 2.083 \times 10^{-4}$ for NRT traffic, and $\alpha_{be} = 0$ for best-effort traffic, to mimic present-day Internet. These coefficients were computed as follows: using 64 Kbps link we can deliver 480 KB of data per minute. If it costs us 10 cents per minute for this link, we can deliver ~ 5 MB data for a dollar. If we are using 1 KB packets, then it costs 2.083×10^{-4} for one such packet. Instead, if it costs 20 cents per minute for this link, then the cost of sending a 1 KB packet is 4.164×10^{-4} and it will cost 8.328×10^{-4} for such a packet if the link costs 40 cents.

A wide series of charging coefficients (multiples of 2.082×10^{-4}) and with and without fixed connection estab-

lishment charges were used in these simulations. We used \$2.00 for A-RT and \$4.00 for RT as connection-establishing charges in some simulations and set both to zero in others. The charging coefficient for A-RT is always equal to twice that of the real time. We refer to the case when we used 8.328×10^{-4} for RT and 4.164×10^{-4} for A-RT and 2.083×10^{-4} for NRT as *SetA1*. *SetA2* is double (of each element) of *SetA1*, *SetA3* is thrice, *SetA4* is four times *SetA1*, *SetA5* is five times of that of *SetA1*, and so forth. The charging coefficients for peak time are three times the normal coefficients. By default, the charging coefficient for best-effort service class is set to zero.

We also have used another set of charging coefficients, *SetB1*, where we have used 1.12×10^{-5} for RT, 5.6×10^{-6} for A-RT, and 2.8×10^{-6} for NRT. These coefficients were computed as follows: using 1.154 Mbps link we can deliver 8.655 MB of data per minute. If it costs us 10 cents per minute for this link, we can deliver ~ 86.6 MB data for a dollar. If we are using 1 KB packets, then it costs 1.12×10^{-5} for one such packet. On the other hand, if such a link costs 5 cents per minute, the 1 KB packet costs 5.6×10^{-6} . Similarly, we compute the coefficient for NRT. Using this set if, for example, we are watching a 90-minute movie using 1.154 Mbps and A-RT service class, this movie will cost the customer \$4.47. Like the case with *SetA1*, we have used *SetB1* to define *SetB2*, *SetB3*, and so forth. Table 1 lists some emerging applications, the traffic service they utilize, and their typical cost to users.

4.3.3 Simulated Pricing Schemes

In order to provide the relative merit of our pricing model, we have simulated four pricing mechanisms. Firstly, we

Table 2
Simulation Parameters

Simulations Parameters	Configuration	Traffic mix	Charging coefficients
Fig. 6	Configuration 1	2, 3 and 4	<i>SetA1</i>
Fig. 7	Configuration 2	2 and 4	<i>SetA1</i>
Fig. 8	Configuration 1	2	<i>SetA1</i>
Fig. 9	Configuration 2	2 and 4	<i>SetA1</i>
Fig. 10	Configuration 1	2, 3 and 4	<i>SetA1</i>
Fig. 11	Configuration 2	2, 4 and 5	<i>SetA1</i> ... <i>SetA7</i>
Fig. 12	Configuration 2	2, 4 and 5	<i>SetB1</i> ... <i>SetB7</i>
Fig. 13	Configuration 2	2, 4 and 5	<i>SetA1</i> ... <i>SetA7</i>
Fig. 14	Configuration 2	2, 4 and 5	<i>SetB1</i> ... <i>SetB7</i>
Fig. 15	Configuration 2	2, 4 and 5	<i>SetA1</i> ... <i>SetA7</i>
Fig. 16	Configuration 2	2, 4 and 5	<i>SetB1</i> ... <i>SetB7</i>
Fig. 17	Configuration 2	2, 3, 4 and 5	<i>SetA1</i> ... <i>SetA7</i>

simulated a flat-rate (per packet) scheme. As there is no incentive for the users in this scheme, every user will choose the highest level of service they can get. Secondly, we simulated an algorithm based on ToS with no QoS available, which assigns a connection into one of four available ToS based on settings of two bits in the header of packets (similar to work in [2], we refer to this scheme as Fixed ToS scheme). Thirdly, an algorithm that assigns a connection to one of a few ToS with fixed QoS levels was simulated. The QoS levels were defined by parameters such as packet loss probability and average delay, or packet loss probability, maximum and average delay. The users in the same class compete for the shared resources. This scheme is similar to the one reported in [14]. We refer to this scheme as Fixed QoS Levels scheme. Finally, our proposed pricing mechanism where the users can specify their QoS parameters and ToS freely was simulated.

4.4 Simulation Results

A pricing scheme must be able to generate sufficient revenue for network operation and expansion. In addition, as with any other commodity, the users of the network want to be satisfied with their performances and the providers want to be profitable and competitive. In this section, we have studied user satisfaction, expressed using the utility functions, and the percentage of blocked/rejected calls (recorded by call admission mechanism), indicating the successes and failures of potential connections and hence expressing a second form of user satisfaction. The study is performed through a series of simulations conducted on two network configurations with increasing complexities and a variety of traffic sources. We also study network utilization, expressed as a ratio between the total amounts

of bandwidth required for all connections and total bandwidth of bottleneck, and investigate the generated revenue. We have also studied the relative performance and scalability of our pricing scheme and conducted an elaborate comparison with other techniques. The effect of varying charging coefficients is also studied. In each section below, we begin by presenting the simulation on simple network configuration and one type of traffic mix. Then we move to simulations conducted on more complex configuration subjected to several traffic mixes in order to measure the effect of burstiness, locality, peak/nonpeak time, and emerging traffic source trends.

We simulated different network scenarios and traffic sources under the above traffic mixes. Simulations conducted under genuine traffic sources with traffic mixtures are essential to study the robustness of our pricing scheme. Below are the aspects of our proposed pricing mechanism that were simulated and analysed:

- Network performances
- Rejection of call admission requests
 - One form of user satisfaction that indicates the satisfaction of potential users
- User satisfaction
 - Using utility functions, this indicates the satisfaction of admitted users
- Effect of secure communications
- Effect of different charging mechanisms
- Effect of network size on performance, call blockings, etc.—scalability
- Effect of peak/nonpeak scenarios
- Comparison with other proposed pricing mechanisms

Table 2 gives details of simulation parameters used for simulations presented in Figs. 6-12.

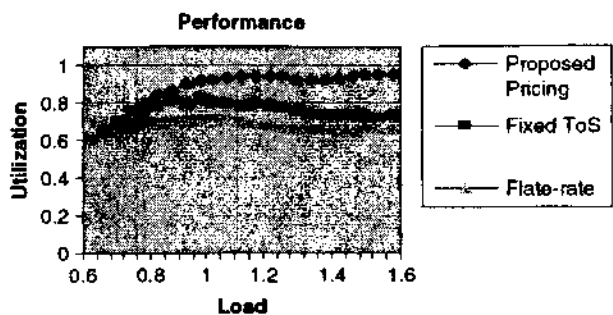


Figure 6. Utilization for configuration 1 with no background flows.

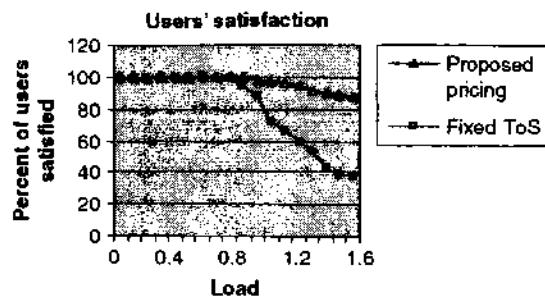


Figure 10. Users' satisfaction for configuration 2.

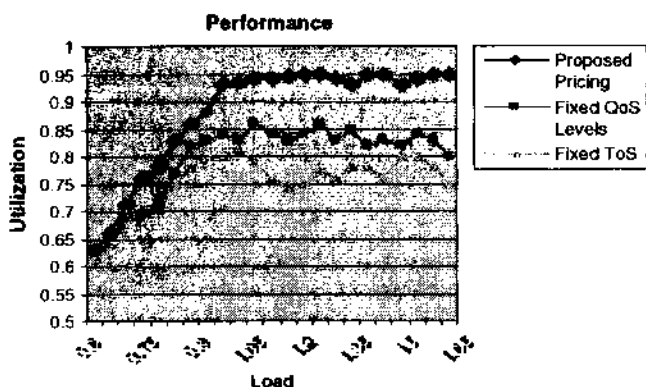


Figure 7. Utilization for configuration 2 for all pricing schemes.

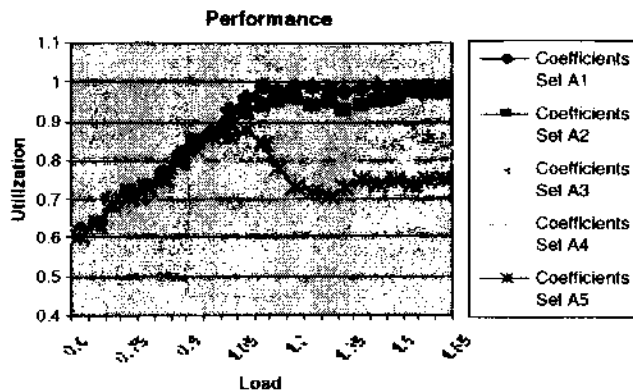


Figure 11. Utilization for configuration 3 and varying coefficients; *Set A1*.

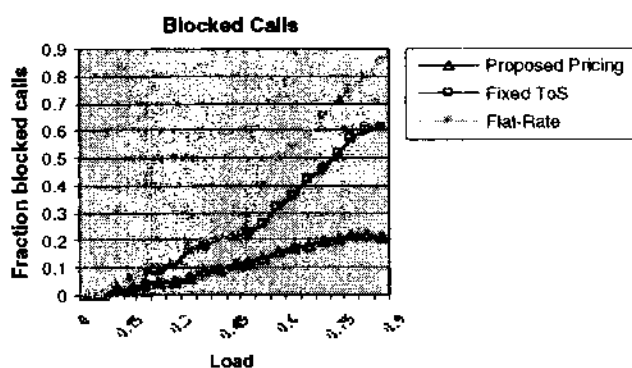


Figure 8. Call blockings for three schemes for configuration 1.

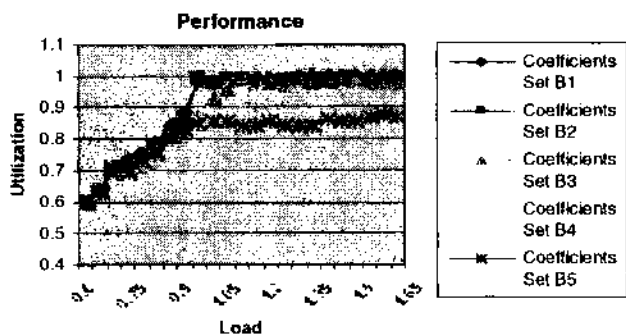


Figure 12. Utilization for configuration 3 and varying coefficients; *Set B1*.

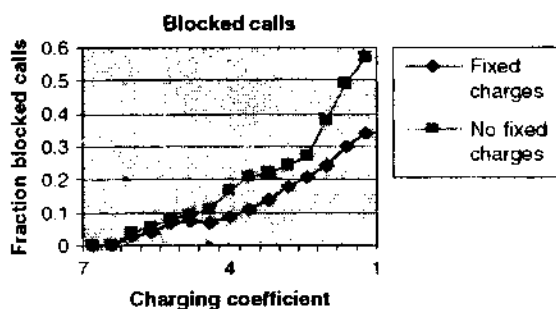


Figure 9. Call blockings for configuration 2.

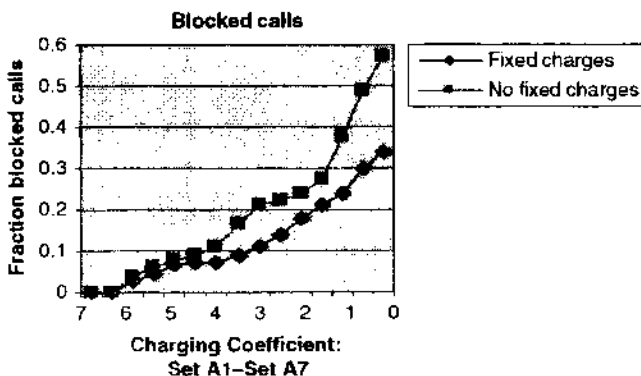


Figure 13. Call Blockings for Varying Coefficients & Fixed Charges; *Set A1-Set A7*.

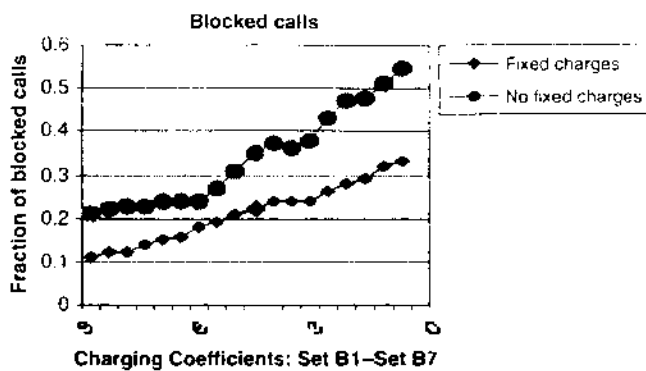


Figure 14. Call blockings for varying coefficients & fixed charges: *SetB1-SetB7*.

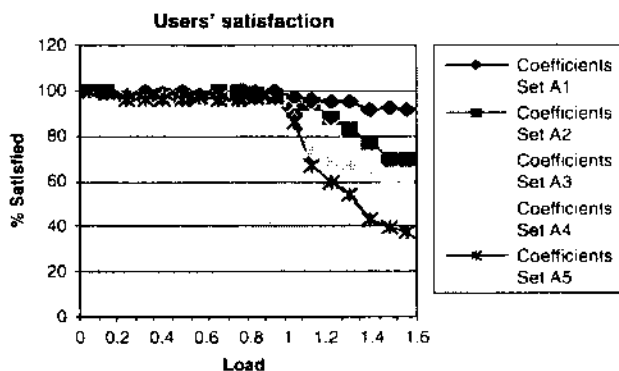


Figure 15. Users' satisfaction. Varying coefficients. *SetA1-SetA7*.

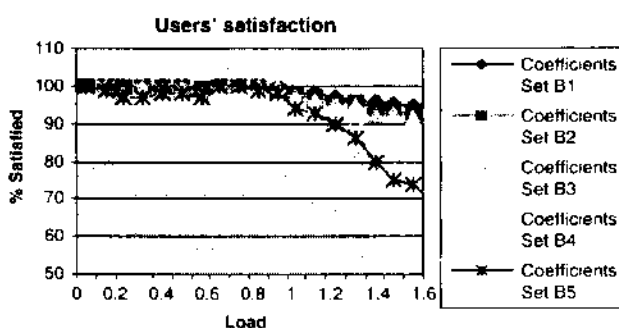


Figure 16. Users' satisfaction. Varying coefficients. *SetB1-SetB7*.

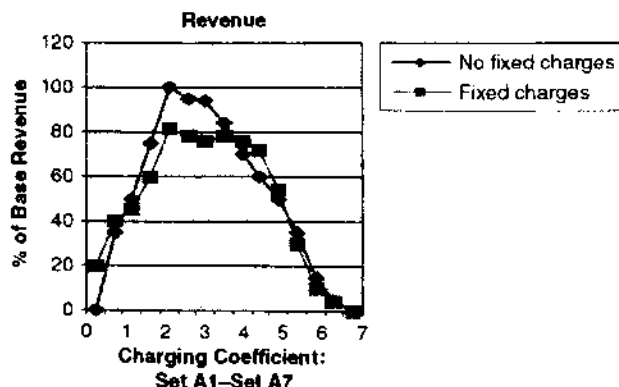


Figure 17. Revenue. Varying coefficients and with/without fixed charges.

4.4.1 Network Performance

Fig. 6 shows network utilization for our scheme versus that of Fixed ToS scheme as well as flat-rate scheme. The default values for all other pricing parameters and simulation variables were used in these simulation runs.

Utilization increases continuously when our pricing mechanism is used and settles near the target utilization for the experiments. On the other hand, utilization is low (with various degrees) when flat-rate pricing is used as a means to provide monetary incentives for the users. In this case, users opt for the highest quality of service each time they request a connection establishment. Before the load reaches the capacity, the proposed scheme performs about 12%-15% better than the flat-rate scheme. After enough flows are admitted and the resources reservations exceed the capacity, the decline becomes sharper. After this point the system will start rejecting new requests. As new flows are not admitted to utilize some of the unused bandwidth, we notice a continued decline in utilization. In this case, the performance of flat-rate scheme degrades to about 52%, whereas our proposed scheme maintains its utilization around 95%. On average, however, our scheme utilizes the link over 35% better than the flat-rate scheme. Note that the small peaks in the graph are due to new connections that are established.

Fixed ToS scheme provides four types of services to users so that they can assign their traffic too, by setting appropriate bits in the header of packets. Because the users within each ToS have no incentives to use lower quality, they opt for the highest quality they can get. As this scheme provides incentives for choosing among four ToS, it shows better utilization than the flat-rate scheme. However, as we can observe from the results, our proposed scheme utilizes the network better. The performance of the Fixed ToS scheme degrades from 77%-80% (when the load is just approaching the capacity) to 55%-57% as we increase the capacity and stabilizes just below 60%.

The utilization of our scheme, on the other hand, exceeds 90% and stabilizes near 95%. Note that we have more than one path between source hosts and destination hosts: new calls are still admitted even after exhausting the resources on the bottleneck link under study. Hence, we do not observe continued decline. Due to burstiness in the traffic mix, there are small peaks in the utilization graphs.

On average the Fixed ToS scheme performs 15%-20% better than flat-rate. However, the flat-rate scheme suffers from over-reservation of resources and ultimate decline in utilization. As enough flows are admitted towards the beginning of the simulations, and because the flows are of longer durations, new requests for call admission are rejected. Hence, we observe that as we increase the load, the utilization for both flat-rate and Fixed ToS continue to decline. This is due to the fact that more and more reserved resources are not utilized, and because they are reserved the network cannot admit new flows. The end result is lower utilization. Before the load reaches the capacity, the proposed scheme performs about 12%-15% better than the flat-rate scheme. Our proposed scheme, on the other hand, maintains its superiority and performs 25%-35% better

than both schemes as the network load is increased beyond the capacity (after enough flows are admitted).

Fig. 14 shows the utilization for the Fixed ToS scheme, Fixed QoS Levels scheme, and the proposed scheme. The network configuration in this simulation is very complex, and the traffic sources are highly bursty with the peak-time usage effect also introduced. This is a typical real-life network scenario with the emerging new applications. The results show that for Fixed ToS and Fixed QoS Levels the utilization decreases with load exceeding the capacity and then settles near 71% and 76%, respectively. However, our proposed scheme has 95% utilization. Before the load reaches the capacity, the proposed scheme performs about 15%-18% better than both schemes (with the Fixed QoS Levels performing better than the Fixed ToS in these load levels). We can observe that our scheme utilizes the network much better than the other two schemes (Fixed QoS Levels and Fixed ToS schemes) in this complex network configuration. Note that the dynamic connection completion and admission of new connections, during the simulations, results in a small surge or drop in utilization. The burstiness in traffic and the nature of traffic mix also have this effect. This can be seen in the figures in the form of peaks.

The results in Fig. 15 in general reaffirm the results of Fig. 13. However, when proper pricing scheme is not used, utilization is worse. The last observation is important as it shows that the more complex the network configuration is (which is certainly the case with the Internet and its future derivatives), the more a robust and efficient pricing scheme will result in network-wide better utilization.

As a direct conclusion of these observations, we can deduce that by introducing proper pricing mechanism, we achieve enhanced utilization.

In addition, the pricing mechanism provides suitable monetary incentives that the users respond to by properly choosing the method, class, and QoS parameters for their connections such that they can balance their required performance in view of the monetary incentives provided by the network. Hence, pricing can achieve congestion avoidance by appropriate network usage patterns and adjusted user behaviour.

4.4.2 Call Blockings

In this section, we study the effect of pricing mechanism on the rate of rejection of new connection requests and analyse the simulation results. As we noted above, utilization increases continuously when pricing is used and settles near the target utilization for the experiments. On the other hand, utilization is generally low when proper pricing mechanism is not utilized as a means to provide monetary incentives for the users. In this case users opt for the highest quality of service each time they request a connection establishment. The end result is not only low utilization, but also high fraction of new connection requests being blocked. The reason is that resources are being unnecessarily reserved as users generally choose the high end of quality. This is shown in Fig. 16. The fraction of blocked calls increases with network complexity. How-

ever, the increase is marginal for the proposed scheme. As we can observe, Fixed ToS scheme performs about 18%-20% better than flat-rate scheme, whereas the proposed scheme results in very low blocking rate. In Fig. 17, the results for Fixed ToS and the proposed scheme are shown for the more complex network configuration. As we can observe from the results, the proposed scheme still enjoys very low blocking rate in this complex scenario (in terms of both network configuration and traffic mix). The scheme on average performs 30%-35% better than the Fixed ToS scheme.

4.4.3 User Satisfaction

User satisfaction was determined by using the utility functions defined in Section 3.8. Fig. 15 shows users' satisfaction as a function of load for the first configuration. The figure shows that users' satisfaction deteriorates (normalized for all user) as we increase the offered load, particularly when fixed four ToS scheme is employed. This is due to this scheme's failure to take into account the QoS requirement of users. As such, when several flow traffics are in active mode, multiple packets drop as the network becomes congested (due to high load). The proposed scheme, on the other hand, retains a 92% satisfied users' base, which is a huge performance advantage in comparison to the Fixed ToS scheme.

When we combine these results with the apparent increase in the percent of blocked calls, as given in Figs. 16 and 17, a complete picture of deteriorated user satisfaction emerges, both in the case where users do get connections and when their requests for connections are refused.

4.4.4 Effect of Charging Coefficients

Figs. 6-11 show a number of results for a wide series of charging coefficients and with and without fixed connection establishment charges. Note that we refer to the case when we used 8.328×10^{-4} for RT, 4.164×10^{-4} for A-RT, and 2.082×10^{-4} for NRT as *SetA1*. *SetA2* is double (of each element) of *SetA1*, *SetA3* is thrice, and so forth (see Table 3). We used \$2 for A-RT and \$4 for RT as connection-establishing charges. The charging coefficients for A-RT is always equal to twice of that of the real time.

Table 3
Charging Coefficients—Set A

	RT	A-RT	NRT
<i>SetA1</i>	8.328×10^{-4}	4.164×10^{-4}	2.082×10^{-4}
<i>SetA2</i>	16.656×10^{-4}	8.328×10^{-4}	4.164×10^{-4}
<i>SetA3</i>	24.984×10^{-4}	12.492×10^{-4}	6.246×10^{-4}
<i>SetA4</i>	33.312×10^{-4}	16.656×10^{-4}	8.328×10^{-4}
<i>SetA5</i>	41.640×10^{-4}	20.820×10^{-4}	10.410×10^{-4}
<i>SetA6</i>	49.968×10^{-4}	24.984×10^{-4}	12.492×10^{-4}
<i>SetA7</i>	58.296×10^{-4}	29.148×10^{-4}	14.574×10^{-4}

The charging coefficients for peak time are three times the normal coefficients. By default, the charging coefficient best-effort service class is set to zero.

We have also used a second set of charging coefficients *SetB1* where we have used 1.12×10^{-5} for RT, for 5.6×10^{-6} for A-RT, and for 2.8×10^{-6} for NRT. Using these coefficients, Table 1 gives costs for the usage of some of the emerging applications. Like the case with *SetA1*, we have used *SetB1* in the same manner to define *SetB2* and so forth (see Table 4).

Table 4
Charging Coefficients—Set B

	RT	A-RT	NRT
<i>SetB1</i>	1.12×10^{-5}	5.6×10^{-6}	2.8×10^{-6}
<i>SetB2</i>	2.24×10^{-5}	1.12×10^{-5}	5.6×10^{-6}
<i>SetB3</i>	3.36×10^{-5}	1.68×10^{-5}	8.40×10^{-6}
<i>SetB4</i>	4.48×10^{-5}	2.24×10^{-5}	1.12×10^{-5}
<i>SetB5</i>	5.60×10^{-5}	2.80×10^{-5}	1.40×10^{-5}
<i>SetB6</i>	6.72×10^{-5}	3.36×10^{-5}	1.68×10^{-5}
<i>SetB7</i>	7.84×10^{-5}	3.92×10^{-5}	1.96×10^{-5}

In Fig. 6 we observe the network utilization for complex network configuration. In this simulation, each user belonged to a specific class (randomly assigned) based on the fixed budget they allocated for network use. Here we used three user classes (rich, middle class, and poor). We observe that utilization is high for charging coefficients *SetA1* and *SetA2*. As we increase the charging coefficients, we see that the utilization drops (see *SetA3* and *SetA4*). In this case poor users stop using the network as their budget is exhausted. However, when *SetA5* is used the utilization is very low. This is because most middle-class users also dropped after using the network for only a few minutes. The drop is very steep here (dropping to 75% compared with 96% for *SetA2*). The small peaks in the graph are due to new connections that are established.

We have conducted the same simulations using more realistic charging coefficients *SetB1-SetB5* to study the utilization in Configuration 2 network infrastructure. The results are given in Fig. 7. As expected, the utilization remains very high (97%) for *SetB1-SetB3*; it slightly drops (94%-93%) for *SetB4*; and for *SetB5* it is 84% for high loads. Therefore, we can observe that the network utilization is very high when the charging coefficients are low. In this range, there is no noticeable difference between the results of *SetB1*, *SetB2*, and *SetB3*, suggesting that when the coefficients are very low and when the user budgets allow, increasing the coefficients does not adversely affect the utilization. When the charging is increased to higher levels (more than threefold), the difference becomes apparent. The network providers can choose charging coefficients in such a way that the desired utilization is achieved, keeping in mind realistic user budgets for the network usages.

As we can see in Fig. 8, when connection establishment charges are not applied, the rate of new connection rejection becomes low for larger coefficients, as fewer users can afford and/or want to use the network when the cost of usage becomes very high. The results are worse, in general, when fixed connection charges are applied. Fig. 9 shows similar results for *SetB1-SetB7*. We can observe that the rejection rate is higher than when we used *SetA1-SetA7*. However, in this case the increase is slower than in Fig. 8. We can conclude that as we reduce the charging coefficients, call rejection rate becomes higher. This is because more users can afford to use the network, or their budgets allow them to use it more, and hence the chances of call rejection increase.

We also observe a higher number of unsatisfied users (when we used *SetA1-SetA7* as shown in Fig. 10) as the coefficients are increased. The reason is that as the users pay more for the services, satisfaction deteriorates at a faster rate than when they pay less. In Fig. 11, we observe that the number of unsatisfied users increases more slowly with charging coefficients *SetB1-SetB7*.

4.4.5 Revenue

Fig. 12 shows revenue for a fixed range of offered loads [1-1.2] and for a wide series of charging coefficients (see Table 3), and with and without fixed connection establishment charges. We used \$2 for A-RT and \$4 for RT as connection-establishing charges. Approximately \$4200 was collected in revenue for *SetA1*. All other revenues are given in terms of percentage of this base case. Revenue grows as we increase the value of coefficients and then decreases sharply after reaching a maximum. The reason is that when the provider charges more for each unit of usage, some users will no longer be able (or willing) to pay for the services and will stop using the network, hence decreasing the revenue. When connection establishment charges are applied, the figure shows a similar trend to the case where no such charges were applied. However, revenues are mostly low, as users who allocate smaller amount of money for network usage are discouraged.

As we previously noted, in general, the higher the charging coefficients are, the lower the network utilization is. Also, user satisfaction deteriorates faster as the charging increases beyond a certain amount. This acts as an incentive for the network providers to choose proper coefficients.

5. Conclusion

In this work, we have studied and analysed a computationally simple and scalable pricing framework. We have studied user satisfaction, expressed using utility functions, network utilization, and percentage of blocked calls through a series of simulations conducted on three network configurations with increasing complexities.

We have simulated four pricing mechanisms. Firstly, we simulated a flat-rate scheme. Secondly, we simulated an algorithm based on ToS with no QoS available, which

assigns a connection into one of four available ToS (similar to work by [2]). Thirdly, an algorithm that assigns a connection to one of a few ToS with fixed QoS levels was simulated. The QoS levels were defined by parameters such as packet loss probability and average delay, or packet loss probability and maximum and average delay. The users in the same class compete for the shared resources. This scheme is similar to the one reported in [14]. Finally, we simulated our proposed pricing mechanism where users can freely specify their QoS parameters and ToS. Comparisons with other schemes show that our pricing mechanism achieves better utilization, lower percentage of blocked calls, and/or greater user satisfaction.

Better network utilization, lower percentage of blocked calls, and better overall user satisfaction were some of the direct results of employing pricing in QoS-enabled Internetworks. It also provides a means for network providers to ensure, with high probability, cost recovery and profit, competitiveness of prices, and encouragement of client behaviours that will enhance the network's efficiency.

The framework not only provided sufficient incentives to users to modify their behaviour (through the usage of network resources), but also gave insights and incentives to providers (in terms of revenue and satisfied customers) to modify their behaviour (in terms of fair coefficients and policies). The scheme acts as an incentive tool to network providers, by keeping the charging coefficients reasonable. As we observed, when reasonable charging coefficients were used, user satisfaction was increased as well as revenues. Results were even better when we did not apply fixed connection charges. Also, the overhead involved with the initial "handshake" that is part of the security mechanism is marginal.

Furthermore, as mentioned earlier, all the pricing-related work was done either at the edge routers or at dummy hosts, called pricing agents, attached to the edges routers, and therefore the proposed pricing framework is designed to easily scale. To evaluate the scalability of the framework, we conducted specific simulations to evaluate the effects of certain predefined variables on the system. As we observed, the performance of the framework deployed in different network configurations, ranging from a simple one with only one bottleneck link to highly complex networks, remained consistent in all the simulations. In addition, the increase in the number of users (flows), the change in the characteristics of the traffic sources required by different service classes, and the increase in the volume of data transmitted through the network did not adversely affect the consistency of the performance of our proposed pricing framework. Therefore, it is evident that the proposed pricing framework is scalable.

We can deduce from these results that efficient pricing mechanisms coupled with traditional congestion control mechanisms are the ultimate solution to congestion and that such mechanisms can provide an accurate quantifier/indicator of the delivered QoS. Also, we can conclude that pricing is an effective regulatory tool. It can be deployed even in private networks to enforce quota and assign virtual money to organizations, individuals, systems, systems activities such as backups/restores and replications,

or even applications for their usages of the congestible and finite resources.

An additional advantage of pricing, therefore, is that it can be a guiding tool for network designers to have their designs not driven by peak-traffic-load considerations alone.

In this work, user satisfaction and behaviour were modelled using parameters detailed in this work. However, in order to fully understand the robustness of the system, it is necessary to deploy the pricing scheme in real Internetworks environment and subject it to real users. A university campus or a corporate domain is an ideal place to start practical deployment of such a pricing framework for analysis.

We have assumed that all the routers set the same charging coefficients and that each network domain received equal revenue shares when it was used in the route. More experiments are needed to study the issue of revenue distribution that can be made sensitive to the delay each packet experiences in each network domain in the federated Internetworks.

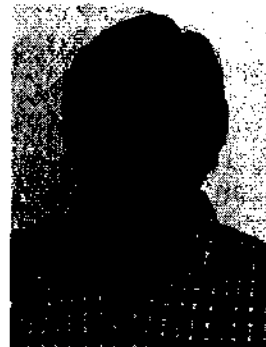
Also, the issue of coefficient estimation is to be investigated. The changing demand patterns (with potential modified network usage), revenue, suppliers' competitions, and other market forces are also to be used to guide the process. It is also interesting to study the feasibility and generalization of similar incentive-based pricing techniques for server and other congestible resources. This seems to be a natural supplement to pricing the network.

References

- [1] J. Altmann, H. Oliver, H. Daanen, & A. Sanchez-Beato Suarez. How to market-manage a QoS network. *Proc. Conf. on Computer Communications (IEEE Infocom)*, New York, June 2002, 284-293.
- [2] R. Cocchi, S. Shenker, D. Estrin, & L. Zhang. Pricing in computer networks: Motivation, formulation, and example. *IEEE/ACM Trans. on Networking*, 1(6), 1993, 614-627.
- [3] MacKie-Mason & Varian. Pricing the Internet, in B. Kahin & J. Keller (Eds.), *Public access to the Internet* (Boston, MA: ACM, 1993).
- [4] A. Gupta, D. Stahl, & A. Whinston. Priority pricing of integrated services networks, in L.W. McKnight & J.P. Bailey (Eds.), *Internet economics* (Cambridge, MA: MIT Press, 1997), 323-352.
- [5] S. Shenker. Service models and pricing policies for an integrated services Internet, in B. Kahin & J. Keller (Eds.), *Public access to the Internet* (Englewood Cliffs, NJ: Prentice-Hall, 1995).
- [6] J. Walrand & P. Varaiya. *High-performance communication networks*, 2nd ed. (San Francisco, USA: Morgan Kaufmann, 2000).
- [7] B. Parenteau & N. Rische. Internet pricing and prioritization. *Proc. 4th Int. Workshop on Community Networking*, Atlanta, Georgia, September 1997, 93-101.
- [8] D. Ferrari & L. Delgrossi. Charging for QoS, *Proc. IntQOS*, Napa, CA, May 1998, vii-xiii.
- [9] S. Faizullah & I. Marsic. A pricing framework for QoS capable Internet. *Proc. 12th IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS 2000)*, vol. 2, Las Vegas, NV, November 2000, 569-577.
- [10] S. Faizullah & I. Marsic. Pricing QoS: Simulation and analysis. *Proc. 34th Annual Simulation Symposium (ASTC 2001)*, Seattle, WA, April 2001, 193-199.
- [11] S. Faizullah & I. Marsic. An architecture for guaranteeing QoS and charging in future Internetworks. *Proc. IASTED Int. Symp. on Information Systems and Engineering (ISE'2001)*, Las Vegas, NV, June 2001, 202-208.

- [12] S. Faizullah & I. Marsic, QoS guarantees and security in future Internetworks, *Proc. 5th Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, FL, July 2001, 46-51.
- [13] S. Faizullah & I. Marsic, Charging for QoS in Internetworks, *Proc. IEEE GlobeCom 2001*, San Antonio, TX, November 2001, 937-943.
- [14] J. Sairamesh, *Economic Paradigms for information systems and networks*, Department of Electrical Engineering, Columbia University, New York, December 1996.
- [15] C. Parris, S. Keshav, & D. Ferrari, *A framework for the study of pricing in integrated networks*, Institute of Computer Science, no. TR-92-016, Berkeley, CA, March 1992.
- [16] P. Fishburn & A. Odlyzko, *Dynamic behavior of differential pricing and quality of service options for the Internet*, AT&T Labs—Research, Florham Park, N.J., June 1998.
- [17] R. Edell, N. McKeown, & P. Varaiya, Billing users and pricing for TCP, *IEEE Journal on Selected Areas in Communications*, 13(7), 1995, 1162-1175.
- [18] S. Dewan & H. Mendelson, User delay costs and internal pricing for a service facility, *Management Science*, 36(12), 1990, 1502-1517.
- [19] D. McDysan, *QoS & traffic management in IP & ATM networks* (New York: McGraw Hill, 2000).
- [20] L. Zhang, S. Deering, D. Estrin, S. Shenker, & D. Zappala, RSVP: A new resource ReSerVation protocol, *IEEE Network*, 7(5), 1993, 8-18.
- [21] C. Aurrecochea, A. Campbell, & L. Hauw, A survey of QoS architectures, *ACM/Springer Verlag Multimedia Systems Journal (Special issue on QoS architecture)*, 6(3), 1998, 138-151.
- [22] P. Ferguson & G. Huston, *Quality of service: Delivering QoS on the Internet and in corporate networks* (New York: Wiley, 1998).
- [23] D. Reininger & D. Raychaudhuri, Dynamic bandwidth allocation for VBR video over ATM networks, *IEEE Journal of Selected Areas on Communication*, 95-R-021 (AP), 14(6), 1996, 1076-1086.
- [24] S. Faizullah & A. Shaikh, A feedback-based UPC-parameters renegotiation strategy, *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Application (PDPTA '99)*, Las Vegas, NV, June 28-July 1, 1999.
- [25] ITU-T, *B-ISDN operation and maintenance principles and functions*, Recommendation I.356, October 1996.
- [26] S. Jordan, Pricing of buffer and bandwidth in a reservation-based QoS architecture, in *ICC 2003: Communication QoS, reliability, and performance modeling*, 3, Anchorage, Alaska, May 2003, 1521-1525.
- [27] M. Karsten, J. Schmitt, B. Stiller, & L. Wolf, Charging for packet-switched network communication motivation and overview, *Computer Communications*, 23(3), 2000, 290-302.
- [28] M. Yuksel, S. Kalyanaram, & A. Goel, Congestion pricing overlaid on edge-to-edge congestion control, in *ICC 2003: Global services and infrastructure for next generation networks*, Vol. 2, Anchorage, Alaska, May 2003, 880-884.
- [29] R. Sprenkels, R. Parhonyi, A. Pras, B.-J. van Beijnum, & L. de Goede, An architecture for reverse charging in the Internet, in *IEEE Workshop on IP-Oriented Operations and Management (IPOM 2000)*, Cracow, Poland, September 2000, 87-92.
- [30] D. Crawford, Pricing network usage: A market for bandwidth or market for communication?, *Proc. of MIT Workshop on Internet Economics*, Cambridge, MA, March 1995.
- [31] A. O'Donnell & H. Sethu, A novel, practical pricing strategy for congestion control and differentiated services, in *Conf. Record of the Int. Conf. on Communications (ICC)*, 2, New York, April 2002, 986-990.
- [32] J. Saltzer & D. Clark, End-to-end arguments in system design, *ACM Trans. on Computer Systems*, 2(4), 1984, 277-288.
- [33] R. Callon, Evolution of multiprotocol label switching, *IEEE Comm. Magazine*, May 1998, 165-173.
- [34] M. Laitinen, M. Fayad, & R. Ward, The problem with scalability, *Comm. ACM*, 43(9), September 2000.
- [35] G. Stamoulis, M. Anagnostou, & A. Georgantas, Traffic source models for ATM networks: A survey, *Computer Communications*, 17(6), 1994, 428-438.
- [36] ITU-T, *Traffic control and congestion control in B-ISDN*, Recommendation I.371, 1996.
- [37] W. Leland, M. Taqqu, W. Willinger, & D. Wilson, On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Trans. on Networking*, 2(1), 1994, 1-15.

Biographies



Safi Faizullah received his M.Sc. and M.Phil. degrees in computer science in 2000 and 2001 and his Ph.D. in computer science in 2002, all from Rutgers University, New Brunswick, USA. Dr. Faizullah also earned B.Sc. and M.Sc. degrees in information and computer science from KFUPM, Dhahran, KSA, in 1991 and 1994, respectively. His research interests are in computer networks, mobile computing, wireless networks, and distributed and enterprise systems. He has authored over 20 journal and conference papers. Dr. Faizullah works for Hewlett-Packard Inc. and is a member of IEEE and ACM.



Ivan Marsic is an assistant professor of electrical and computer engineering at Rutgers University. He received his B.Sc. and M.Sc. degrees in computer engineering from the University of Zagreb, Croatia, in 1982 and 1987, respectively, and his Ph.D. in biomedical engineering from Rutgers University, New Brunswick, NJ, in 1994. Prof. Marsic's research interests are in mobile computing, groupware, computer networks, and multimodal human-computer interfaces. He has authored over 80 journal and conference papers, one book, and three book chapters. He is a member of IEEE, ACM, IFIP WG2.7/13.4, and the International Academy for Advanced Decision Support and has been a consultant to industry and government.