Performance Study of Congestion Price based Adaptive Service

Xin Wang, Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
xwang@ctr.columbia.edu, schulzrinne@cs.columbia.edu

Abstract

In a network with enhancements for QoS support, pricing of network services based on the level of service, usage, and congestion provides a natural and equitable incentive for applications to adapt their sending rates according to network conditions. In this paper, we first propose a dynamic, congestion-sensitive pricing algorithm, and also develop the demand behavior of adaptive users based on a physically reasonable user utility function. We then develop a simulation framework to compare the performance of a network supporting congestion-sensitive pricing and adaptive reservation to that of a network with a static pricing policy. We also study the stability of the dynamic pricing and reservation mechanisms, and the impact of various network control parameters. The results show that the congestion-sensitive pricing system takes advantage of application adaptivity for significant gains in network availability, revenue, and user-perceived benefit, relative to the fixed-price policy. The congestion-based pricing is stable and effective in limiting utilization to a targeted level. Users with different demand elasticity are seen to share bandwidth fairly, with each user having a bandwidth share proportional to its relative willingness to pay for bandwidth. The results also show that even a small proportion of adaptive users may result in a significant performance benefit and better service for the entire user population - both adaptive and non-adaptive users. The performance improvement given by the CPA policy further improves as the network scales and more connections share the resources.

1 Introduction

The development and use of distributed multimedia applications are growing rapidly. These applications usually require a minimum Quality of Service (QoS) from the network, expressed as requests on throughput, packet loss, delay, and jitter. To address these problems, one approach is to enhance the network with mechanisms such as resource reservation [8], admission control [14], and special scheduling mechanisms [30]. Another approach is to adjust the bandwidth used by an application according to the existing network conditions [28], relying on signaling mechanisms such as packet loss rates for feedback. Compared to resource reservation, the adaptation approach has the advantage of better utilizing available network resources, which change with time. But if network resources are shared by competing users, users of rate-adaptive applications do not have any incentive to scale back their sending rate below their access bandwidth, since selfish users will generally obtain better quality than those that reduce their rate.

In a network with enhancements for QoS support, pricing of network services based on the level of service, usage, and congestion provides a natural and equitable incentive for applications to adapt their sending rates according to network conditions [26]. Increasing the price during congestion gives the application an incentive to back-off its sending rate and at the same time allows an application with more stringent bandwidth and QoS requirements to maintain a high quality by paying more.

In earlier work, we presented a Resource Negotiation and Pricing (RNAP) protocol and architecture [26]. RNAP enables the user to select from available network services with different QoS properties and re-negotiate contracted services, and enables the network to dynamically formulate service prices and communicate current prices to the user.

In this paper, we first propose a dynamic, congestion-sensitive pricing algorithm, and also develop the demand behavior of adaptive users based on a physically reasonable user utility function. We then develop a simulation framework to compare the performance of a network supporting congestion-sensitive pricing and adaptive reservation to that of a network with a static pricing policy. We also study the stability of the dynamic pricing and reservation mechanisms. We try to answer questions such as how much do the network and users gain in terms of revenue and perceived benefit (or value-for-money) under the dynamic and static systems, and how do various pricing and adaptation parameters affect the functioning of the dynamic system. The simulation framework is based on the RNAP model [26], but we try to derive results and conclusions applicable to static and congestion-driven, dynamic pricing schemes in general.

In Section 2 of this paper, we present a brief outline of the RNAP framework. In Section 3, we discuss various network pricing models and their suitability. We discuss in detail a volume-based, congestion-sensitive pricing strategy, also presented earlier in [26]. In Section 4, we consider user adaptation in response to congestion dependent pricing. We present a physically reasonable form of an user utility function, and derive the specific demand function for a given network price based on this utility function. In section 5, we describe the simulation topology and parameters, performance metrics, and experimental plan. The largest section in this paper is Section 6,in which we describe simulation experiments in detail, and present and discuss the results. In Section 7, we describe some related work by other authors. We summarize our findings in Section 8.

There are a couple of issues that we do not include in the scope of the current work. The first issue is that of pricing in the presence of competing networks with dynamic pricing, and also user adaptation when multiple paths with different prices (from competing networks) are available to the same user. In general, we believe that if a user receives a reasonably stable and satisfactory QoS and price, there will be little incentive for it to switch networks unless there is a large price advantage to be gained. This is certainly an open topic.

The other issue is that congestion pricing can help to balance the network traffic, by raising the price along a path with heavy load, and finding the cheapest path to route the packets. However, in this work, we restrict ourselves mainly to a particular path, and study the dynamics of pricing and user adaptation among competing users due to a bottleneck on this path.

2 Resource Negotiation through RNAP

In this section, we briefly describe the RNAP protocol and architecture [26], as a framework within which incentive-driven adaptation by the user takes place.

In the RNAP framework, we assume that the network makes services with certain QoS characteristics available to user applications, and charges prices for these services that, in general, vary with the availability of network resources. Network resources are obtained by user applications through negotiation between the Host Resource Negotiator (HRN) on the user side, and a Network Resource Negotiator (NRN) acting on behalf of the network. The HRN negotiates on behalf of one or multiple applications belonging to a multimedia system. In an RNAP session, the NRN periodically provides the HRN updated prices for a set of services. Based on this information and current application requirements, the HRN determines the current optimal transmission bandwidth and service parameters for each application. It re-negotiates the contracted services by sending a *Reserve* message to the NRN, and receiving a *Commit* message as confirmation or denial.

The HRN only interacts with the local NRN. If its application flows traverse multiple domains, resource negotiations are extended from end to end by passing RNAP messages hop-by-hop from the first-hop NRN until the destination network NRN, and vice versa. End-to-end prices and charges are computed by accumulating local prices and charges as *Quotation* and *Commit* messages travel hop-by-hop upstream.

In addition to the basic service and price negotiation, RNAP also provides optional data field to facilitate sender and receiver HRNs to negotiate charge sharing and service rate requirements. RNAP is designed to facilitate dynamic price quotation and resource negotiation, and specific pricing strategies are outside the scope of the RNAP protocol itself. Also, RNAP is intended for use by both adaptive and non-adaptive applications. Non-adaptive applications may choose services that offer a static price, or absorb any changes in price while maintaining their sending rate. Adaptive applications adapt their sending rate and/or choice of network services in response to changes in network service prices.

3 Pricing Strategy

A few pricing schemes are widely used in the Internet today [22]: access rate dependent charge (AC), volume dependent charge (V), or the combination of the both (AC-V). An AC charging scheme is usually one of two types: allowing unlimited use, or allowing limited duration of connection, and charging a per hour fee for additional connection time. Similarly, AC-V charging schemes normally allow some amount of volume to be transmitted for a fixed access fee, and then impose a per-volume charge. Although time-of-day dependent charging is commonly used in telephone networks, it is rarely used in the current Internet.

The explosive use of the Internet makes the usage-sensitive charging scheme an appealing way to control the exponential growth [3]. In general, user experiments [4] indicate that usage-based pricing is a fair way to charge people and allocate network resources. Both connection time and the transmitted volume reflect the usage of the network. However, the current popular time-based charging is more appropriate for circuit-based transmission, such as the traditional telephone network, or low bandwidth transmission. It does not reflect the different costs of the huge number of diverse Internet applications, ranging from the simple email to the high bandwidth tele-conference, video on demand, etc. We envision that a viable future Internet pricing scheme needs to take into account this wide range of costs to allow fair and efficient use of network resources; volume-based pricing is more appropriate for this purpose. In this paper, we study two kinds of volume-based pricing: a pricing system with a fixed (or relatively static) unit volume price, and a dynamic system in which the unit volume price has a congestion-sensitive component. We describe the latter system in detail, and also present a generic pricing framework to accommodate the different pricing models. In our simulations, we implement a pricing system with a fixed unit volume price, as well as the dynamic, congestion-sensitive pricing system described in this section.

3.1 Fixed Pricing

Even though the volume-based charging scheme in current Internet normally assumes a fixed rate per megabyte regardless of the network state, we further divide it into four categories: service class independent flat pricing (FP-FL), service class sensitive priority pricing (FP-PR), time-dependent time-of-day pricing (FP-T), and time-dependent service class sensitive priority pricing (FP-PR-T). Since our focus is on the congestion-based dynamic pricing, and the fixed-price system serves as a reference, we assume a general fixed pricing structure that represents all the four categories depending on the underlying network service infrastructure and the service provider's business model. We call the service under this framework as a fixed price based service (FP).

3.2 Congestion-based Pricing

With a fixed pricing structure independent of network conditions, bandwidth-adaptation capable customers will tend to request the maximum bandwidth that their budget permits. As more customers request a service, the network resources become scarce. There is nothing to motivate a customer whose application is less bandwidth sensitive to reduce its resource requirement. As a result, either the service request blocking rate will increase sharply at the call admission control level, or the packet dropping rate will increase greatly at the queue management level. Having a congestion-dependent component in the service price provides a monetary incentive for adaptive applications to adapt their service class and/or sending rates according to network conditions. In periods of resource contention, quality sensitive applications can maintain their resource levels by paying more, and relatively quality-insensitive applications will reduce their sending rates or change to a lower class of service. In this section, we outline a pricing strategy for a congestion price based adaptive service (CPA).

In general, the total price has a network state independent fixed price component, and a network state dependent congestion price component. In the extreme case, the fixed price component can be zero, i.e., users only get charged if there is congestion in the network. A non zero fixed charge part can also be classified into the same four categories as the fixed pricing policy, and the corresponding congestion-based charging policies can be represented as CP-FL, CP-PR, CP-T, CP-PR-T.

We assume the most general structure with multiple service classes (CP-PR-T). However, in subsequent definitions and the actual experiments, we assume that the fixed price component for a service class is time-independent, a reasonable assumption because of the much shorter time scale of changes in the congestion price.

We assume that routers support multiple services and that each router is partitioned to provide a separate link bandwidth and buffer space for each service, at each port. In the discussion that follows, we consider one such logical partition.

We use the framework of the competitive market model [25]. The competitive market model defines two kinds of agents: consumers and producers. Consumers seek resources from producers, and producers create or own the resources. The exchange rate of a resource is called its price. Prices are set such that the amount of resource demanded equals the amount of resources supplied.

The routers are considered the producers and own the link bandwidth and buffer space for each output port. The flows (individual flows or aggregate of flows) are considered consumers who consume resources. The congestion-dependent component of the service price is computed periodically, with a price computation interval τ. The total demand for link bandwidth is based on the aggregate bandwidth reserved on the link for a price computation interval, and the total demand for the buffer space at an output port is the average buffer occupancy during the interval. The supply bandwidth and buffer space need not be equal to the installed capacity; instead, they are the targeted bandwidth and buffer space utilization. When the bandwidth demand exceeds the supply, the network imposes the congestion charge to force the users to reduce the demand.

Thus, in general, the total network price for a service has two components, so that total charge = fixed charge + congestion-sensitive charge. We now discuss the formulation of the fixed charge, which we decompose into the *holding charge* and *usage charge*, and the formulation of the *congestion charge*.

3.2.1 Usage Charge

The usage charge is determined by the actual resources consumed, the average user demand, the level of service guaranteed to the user, and the elasticity of the traffic. For example, on a per-byte basis, best-effort traffic will cost less than reserved, non-preemptable CBR traffic. The usage price (p_u) will be set such that it allows a retail network to recover the cost of the purchase from the wholesale market, and various static costs associated with the service. In a monopoly model, a service provider would set this price by maximizing its total profit. When multiple providers exist, p_u will also depend on the prices set by peer networks.

In general, it can be represented as:

$$p_u = f(\text{service}, \text{service\_demand}, \text{destination}, \text{time\_of\_day}, ...) \tag{1}$$

The usage_charge $c_u(n)$ for a period n in which $V(n)$ bytes were transmitted is given by:

$$c_u(n) = p_u \times V(n) \tag{2}$$

3.2.2 Holding Charge

The holding charge can be justified as follows. If a particular flow or flow-aggregate does not utilize the resources (buffer space or bandwidth) set aside for it, we assume that the scheduler allows the resources to be used by excess traffic from a lower level of service. The holding charge reflects revenue lost by the provider because instead of selling the allotted resources at the usage charge of the given service level (if all of the reserved resources were consumed) it sells the reserved resources at the usage charge of a lower service level. The holding price (p_h) of a service class is therefore set to be proportional to the difference between the usage price for that class and the usage price for the next lower service class. The holding price can be represented as:

$$p_h^i = \alpha^i \times (p_u^i - p_u^{i-1}), \tag{3}$$

where α^i is a scaling factor related to service class i. The holding_charge $c_h(n)$ when the customer reserves a bandwidth $R(n)$ is given by:

$$c_h(n) = p_h \times R(n) \times \tau \tag{4}$$

where τ is the duration of the period. $R(n)$ can be a bandwidth requirement specified explicitly by the customer, or estimated from the traffic specification and service request of the customer.

Defining a usage charge and a holding charge separately allows the customer to reserve resources conservatively, without penalizing him excessively for unused resources. As an example, an audio stream can have periods of silence, when the reserved resources are not used by the customer. Also, not charging the customer purely on the basis of reserved resources makes it easier for the customer to keep his reservation level constant even during idle periods.

3.2.3 Congestion Charge

The congestion charge is imposed when congestion is deduced, that is, the resource request or average usage for a partition (in terms of buffer space or bandwidth) exceeds supply (the targeted buffer space or bandwidth). The congestion price for a service class is calculated as an iterative tâtonnement process [25]:

$$p_c(n) \quad = \quad \min[\{p_c(n-1) + \sigma(D,S) \times (D-S)/S, 0\}^+, p_{max}] \tag{5}$$

Where D and S represent the current total demand and supply respectively, and σ is a factor used to adjust the convergence rate. σ may be a function of D and S; in that case, it would be higher when congestion is severe. The router begins to apply the congestion charge only when the total demand exceeds the supply. Even after the congestion is removed, a non-zero, but gradually decreasing congestion charge is applied until it falls to zero to protect against further congestion. In our simulations, we also used a price adjustment threshold parameter θ to limit the frequency with which the price is updated. The congestion price is updated if the the calculated price increment exceeds $\theta \times p_c(n-1)$.

The maximum congestion price is bounded by the p_{max} parameter so that the total price for a service class does not exceed that for a higher level of service. When a service class needs admission control, all new arrivals are rejected when the price reaches p_{max}. If p_c reaches p_{max} frequently, it indicates that more resources are needed for the corresponding service and new configuration for local resources may be needed.

For a period n, the total congestion charge is given by

$$c_c(n) = p_c(n) \times V(n). \tag{6}$$

Based on a price formulation strategy such as the one we have discussed, a router arrives at a cost structure for a particular RNAP flow or flow-aggregate at the end of each price update interval. The total charge for a session is given by

$$c_s = \sum_{n=1}^{N_s} [p_h \times R(n) \times \tau + (p_u + p_c(n)) \times V(n)] \tag{7}$$

where N_s is the total number of intervals spanned by a session.

In some cases, the network may set the usage charge to zero, imposing a holding charge for resource reservation only, and/or a congestion charge during resource contention. Also, the holding charge would be set to zero for services without explicit resource reservation, for example, best effort service.

3.3 A Generic Pricing Structure

We have now discussed several approaches to charging the customer for network services, and described one of them (usage sensitive congestion based pricing) in detail. The following generic equation represents the charge incurred by a customer for a single billing cycle in all these cases:

$$cost = c_{ac}(R_{ac}) + p(R_{ac}) \times (t - T_m)^+ + \sum_{i=1}^{I} \sum_{n=1}^{N_b} [p_h^i(n) \times R^i(n) \times \tau + (p_u^i(n) + p_c^i(n)) \times V^i(n)](V^i - V_m^i)^+ \tag{8}$$

Here I is the number of service classes in the network, i represents a particular service class, c_{ac} represents the access rate dependent fixed charge, $p(R_{ac})$ is the unit time connection price charged for the excess time above a contracted free of charge duration T_m, t is the total duration of a billing cycle, N_b is the number of price update intervals during a billing cycle, V^i is the total volume of class i traffic transmitted during the billing cyle, V_m^i is the volume of traffic from class i that is free of charge, and other parameters have the same meaning as in Section 3.2. Multiple service classes may be used during a billing cycle, either at different times, or simultaneously for different co-existing applications (for example, belonging to a teleconference application). Also note that, in equation 8, p_h and p_u can vary over a relatively long period of time, for instance a few times a day.

For the different charging modes discussed in previous sections, equation 8 takes the following specific forms:

Note that we have divided the volume-based charging mode into four types of fixed price charging schemes and four types of congestion-based pricing schemes. Further, as equation 8 shows, a volume based charging scheme can also have an access charge component. In that case, the network may either specify a certain threshold volume below which only the access charge applies, or alternatively, specify a threshold rate R_m (less than or equal to the access link rate), so that the volume threshold for a single price updation period is of the form $R_m \times \tau$.

AC:	$cost = c_{ac}(R_{ac}) + p(R_{ac}) \times (t - T_m)^+$
FP-FL:	$cost = \sum_{n=1}^{N_b}[p_h \times R(n) \times \tau + p_u \times V(n)]$
FP-PR:	$cost = \sum_{i=1}^{I}\sum_{n=1}^{N_b}[p_h^i \times R^i(n) \times \tau^i + p_u \times V^i(n)]$
FP-T:	$cost = \sum_{n=1}^{N_b}[p_h(n) \times R(n) \times \tau + p_u(n) \times V(n)]$
FP-PR-T:	$cost = \sum_{i=1}^{I}\sum_{n=1}^{N_b}[p_h^i(n) \times R^i(n) \times \tau^i + p_u^i(n) \times V^i(n)]$
CP-FL:	$cost = \sum_{n=1}^{N_b}[(p_h \times R(n) \times \tau + (p_u + p_c(n)) \times V(n)]$
CP-PR:	$cost = \sum_{i=1}^{I}\sum_{n=1}^{N_b}[p_h^i \times R^i(n) \times \tau + (p_u^i + p_c^i(n)) \times V^i(n)]$
CP-T:	$cost = \sum_{n=1}^{N_b}[p_h(n) \times R(n) \times \tau + (p_u(n) + p_c(n)) \times V(n)]$
CP-PR-T:	$cost = \sum_{i=1}^{I}\sum_{n=1}^{N_b}[p_h^i(n) \times R^i(n) \times \tau + (p_u^i(n) + p_c^i(n)) \times V^i(n)]$

$$cost = c_{ac}(R_{ac}) + \sum_{n=1}^{N_b}[(p_u(n) + p_c(n)) \times (V(n) - R_m \times \tau)^+] \tag{9}$$

Setting a contracted threshold rate instead of a threshold volume encourages users to smooth out their traffic, and thus allows resources to be provisioned more economically.

In our simulations, we implement both a congestion-dependent pricing model for the CPA service, and a fixed price model for the FP service. Since we do not consider service class interactions, and do not consider time-of-day dependence, in effect, we implement the CP-FL and FP-FL models. However, we believe the results from the CPA and FP to be applicable to all the CP and FP pricing models, as well as the access charge inclusive CP model of equation 9, in a lot of important respects, since the most important and influential feature of the models is the presence or absence of congestion-dependent pricing.

4 User Adaptation

In a network with congestion dependent pricing and dynamic resource negotiation (through RNAP or some other signaling protocol), *adaptive* applications with a budget constraint will adjust their service requests in response to price variations. In this section, we discuss how a set of user applications performing a given task (for example, a video conference) adapt their sending rate and quality of service requests to the network in response to changes in service prices, so as to maximize the benefit or *utility* to the user, under the constraint of the user's budget.

Although we focus on adaptive applications as the ones best suited to a dynamic pricing environment, the RNAP framework does not impose adaptation capability as a requirement. Applications may choose services that provide a fixed price, and fixed service parameters during the duration of service. Generally, the long-term average cost for a fixed-price service will be higher, since it uses network resources less optimally. Alternatively, applications may use a service with usage-sensitive pricing, and maintain a high QoS level, paying a higher charge during congestion.

4.1 The Perceived Value Based Utility Function

We consider a set of user applications, required to perform a task or *mission*, for example, audio, video, and white-board applications for a video-conference. The user would like to determine a set of transmission parameters (sending rate and QoS parameters) from which it can derive the maximum benefit, subject to his budget. We assume that the user can define quantitatively, through a *utility function*, the value provided by the corresponding network resource allocation towards completing the mission. The utility function is therefore a function in a multi-dimensional space, with each dimension representing a single transmission parameter allocation for a particular application.

Clearly, the utility of a transmission depends on its quality as perceived by the user. However, since the user is paying for the transmission, it appears reasonable to define the utility as the *perceived value* of that quality to the user. For example, an audio transmission requiring a certain sending rate and certain bounds on the end-to-end delay and loss rate may be worth 15 cents/minute to the user, regardless of the real price quoted from the vendor.

4.2 Application Adaptation

Consumers in the real world generally try to obtain the best possible "value" for the money they pay, subject to their budget and minimum quality requirements; in other words, consumers may prefer lower quality at a lower price if they perceive this as meeting their requirements and offering better value. Intuitively, this seems to be a reasonable model in a network with QoS support, where the user pays for the level of QoS he receives. In our case, the "value for money" obtained by the user corresponds to the surplus between the utility $U(\cdot)$ with a particular set of transmission parameters (since this is the perceived value), and the cost of obtaining that service. The goal of the adaptation is to maximize this surplus, subject to the budget and the minimum and maximum QoS requirements.

We now consider the simultaneous adaptation of transmission parameters of a set of n applications performing a single task. The transmission bandwidth and QoS parameters for each application are selected and adapted so as to maximize the mission-wide "value" perceived by the user, as represented by the surplus of the *total utility* , $\hat{U}$, over the total cost C. We can think of the adaptation process as the allocation and dynamic re-allocation of a finite amount of resources between the applications.

In this paper, we make the simplifying assumption that for each application, a utility function can be defined as a function only of the transmission parameters of that application, independent of the transmission parameters of other applications. Since we consider utility to be equivalent to a certain monetary value, we can write the total utility as the sum of individual application utilities :

$$\hat{U} = \sum_i [U^i(x^i)] \tag{10}$$

where x^i is the transmission parameter tuple for the i_{th} application. The optimization of surplus can be written as

$$\begin{aligned} max \sum_i [U^i(x^i) - C^i(x^i)] \\ \text{s. t.} \sum_i C^i(x^i) \leq b \\ x^i_{min} \leq x^i \leq x^i_{max} \end{aligned} \tag{11}$$

where x^i_{min} and x^i_{max} represent the minimum and maximum transmission requirements for stream i, and C^i is the cost of the type of service selected for stream i at requested transmission parameter x^i.

One way of carrying out this optimization is to fit the utility function to a closed form function. The optimal solution is then obtained by using Kuhn-Tucker conditions for a maximum subject to inequality constraints.

In practice, the application utility is likely to be measured by user experiments and known at discrete bandwidths, at one or a few levels of loss and delay, possibly corresponding to a subset of the available services; at the current stage of research, some possible services are guaranteed [24] and controlled-load service [29] under the int-serv model, Expedited Forwarding (EF) [13] and Assured Forwarding (AF) [12] under diff-serv. In this case, it is convenient to represent the utility as a piecewise linear function of bandwidth (or a set of such functions). A simplified algorithm is proposed in [27] to search for the optimal service requests in such a framework.

4.3 An Example Utility Function and the Adaptation of User Requirements

We can make some general assumptions about the utility function as a function of the bandwidth, at a fixed value of loss and delay. A user application generally has a minimum requirement for the transmission bandwidth. He also associates a certain minimum value with a task, which may be regarded as an "opportunity" value, and this is the perceived utility when the application receives just the minimum required bandwidth. The user terminates the application if its minimum bandwidth requirement can not be fulfilled, or when the price charged is higher than the opportunity value derived from keeping the connection alive. Also, user experiments reported in the literature [17][1] suggest that utility functions typically follow a model of diminishing returns to scale, that is, the marginal utility as a function of bandwidth diminishes with increasing bandwidth. Hence, a utility function can be represented in a general form as:

$$U(x) = U_0 + w \log \frac{x}{x_m} \tag{12}$$

where x_m represents the minimum bandwidth the application requires, w represents the sensitivity of the utility to bandwidth, and U_0 is the monetary "opportunity" that the user perceives in the application. When the utilities of all the applications are represented in the format of equation 12, the optimization process for a system with multiple applications as described in Section 4.2 can be represented as:

$$\begin{aligned} max \sum_j [U_0^j + w^j \log \frac{x^j}{x_m^j} - p^j \times x^j] \\ \text{s. t.} \sum_j p^j \times x^j \leq b \\ \text{and} \quad x^j \geq x_m^j, \forall j \end{aligned} \tag{13}$$

The Lagrangian for this problem is :

$$L(x^j, p^j, b) = \sum_j [U_0^j + w^j \log \frac{x^j}{x_m^j} - p^j \times x^j] + \lambda[b - \sum_j (p^j \times x^j)] + \sum_j \mu^j (x^j - x_m^j) \tag{14}$$

The first order conditions are thus:

$$L_{x^j} = \frac{w^j}{x^j} - (1+\lambda)p^j + \mu^j \leq 0, if <, x^j = 0$$

$$L_\lambda = b - \sum_j p^j * x^j \geq 0, if >, \lambda = 0 \tag{15}$$

$$L_{\mu^j} = x^j \geq 0, if >, \mu^j = 0 \tag{16}$$

Now suppose $x^j > 0$, therfore $\mu^j = 0$. If the user can obtain the optimal bandwidth for the system at a cost below its budget, then $\lambda = 0$, and

$$L_{x^j} = \frac{w^j}{x^j} - p^j = 0$$

$$\text{therefore,} \quad x^j = \frac{w^j}{p^j} \tag{17}$$

Hence, w^j represents the money a user would spend based on its perceived value for an application.

If the total bandwidth a system can obtain is bounded by the budget, then optimal solution for the system becomes:

$$L_{x^j} = \frac{w^j}{x^j} - (1+\lambda)p^j = 0 \tag{18}$$

$$b - \sum_j p^j \times x^j = 0 \tag{19}$$

From the first equation, we can get $p^j x^j = w^j/(1+\lambda)$, and substitute this into the second equation, yielding $(1+\lambda) = \sum_j w^j/b$. Therefore the demand function is

$$x^j = \frac{b \times \frac{w^j}{\sum_l w^l}}{p^j} \tag{20}$$

Therfore, when the budget is a constraint, each application in a system receives a share based on the user's perceived value of this application. Note that the prices applicable different applications in a system (e.g. video conference) can be different, since each application may require different class of service and get a different price quotation from the network.

4.4 Scaling of the Utility Function

In this section, we consider how changes in utility function may influence the resource distribution. The utility function represents the relative preference of the user for different bandwidths. Changes in the opportunity U_0, result in a constant (bandwidth-independent) offset to the utility function, and does not influence the resource distribution as long as the valuation of a bandwidth is higher than its cost.

On the other hand, since U_0 represents how much the user is willing to pay to just keep the application alive, lowering U_0 allows the application to be terminated more readily during congestion. If a user values an uninterrupted service very highly, he increases U_0.

A multiplicative scaling-up of the bandwidth dependent portion of the utility function (by increasing w) tends to increase its bandwidth share, since it results in a bigger additive increase in perceived surplus at higher bandwidth than at lower bandwidths. Effectively, the demand elasticity of the application is reduced. The opposite effect is observed when w is reduced.

5 Simulation Model

In this section, we describe our simulation model for the CPA policy and also the FP policy which is used as a reference. The policies are simulated at the call level, that is, we consider user resouce contention due to the total user requested bandwidth exceeding the provisioned system bandwidth, rather than due to the burstiness of user traffic. Depending on the service type and network infrastructure, the network may learn user resource requirements explicitly through a signalling protocol, or implicitly by traffic measurement. We simulate explicit resource reservation and price signalling through RNAP.

5.1 Experimental Topology and Parameters Setup

We used the *network simulator* [] environment to simulate two different network topologies, shown in Fig. 1 and Fig. 2 Topology 1 contains 2 backbone nodes, 6 access nodes, and 24 end nodes. Topology 2 contains 5 backbone nodes, 15 access nodes, and 60 end nodes. Topology 2 was also used in **??**. All links are full

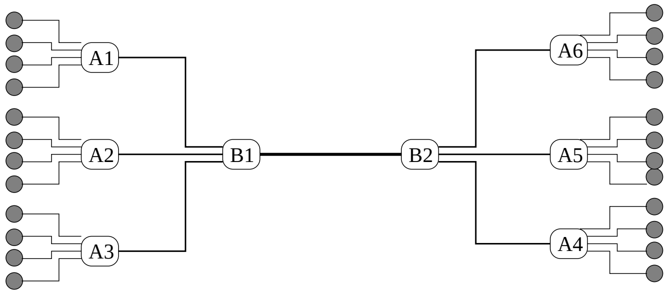

Figure 1: Simulation network topology 1

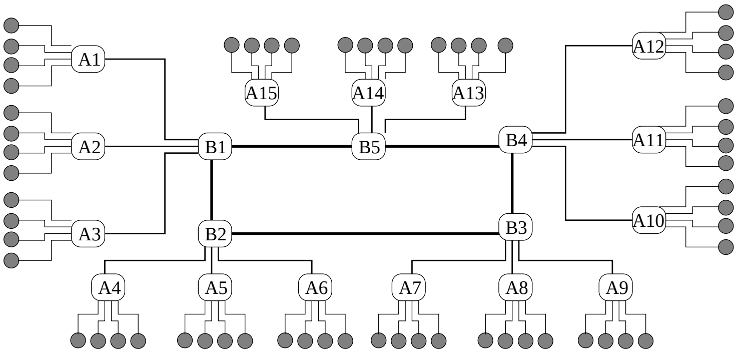

Figure 2: Simulation network topology 2

duplex and point-to-point. The links connecting the backbone nodes are 3 Mbit/s, the links connecting the access nodes to the backbone nodes are 2 Mbit/s, and the links connecting the end nodes to the access nodes are 1 Mbit/s. At each end node, there is a fixed number N_u of sending users. In topology 1, all the users from the sender side independently initialize unidirectional flows towards randomly selected receiver side end nodes. In topology 2, all the users initialize unidirectional flows towards randomly selected end nodes.

Generally, congestion arises due to excessive resource requests from users sharing a common node on a path. Under the CPA framework, congestion pricing is invoked only at this bottleneck node. We use topology 1 in most of our simulations to allow us to simulate congestion from a single bottleneck node, and only use topology 2 to illustrate the CPA performance under a more general network topology in Section 6.8.

User requests are generated according to a Poisson arrival process and the lifetime of each flow is exponentially distributed with an average length of 10 minutes. When a flow leaves, it will spawn a successor that starts after an exponentially distributed idle time; the average length of this idle time is varied to obtain varying loads in the experiments.

When using topology 1, at most $12N_u$ flows may be active in the system, all in the same direction. Unless otherwise specified, we set the number of users N_u at an end node to 4, and hence at most 48 sessions can run simultaneously in the whole network. When using topology 2, at most $60N_u$ users may be active in the system. We set N_u to 6 in topology 2, and allow 360 sessions to run simultaneously. The users are assumed to have the general form of the utility function shown in Section 4.3. w, the elasticity factor, (and also the user's willingness to pay) is uniformly distributed between 0.125 and $0.375/min. The opportunity cost U_0 is set to the amount a user is willing to pay for its minimum bandwidth requirement, and is hence given by $U_0 = p_{high} \times x_{min}$, where p_{high} is the maximum price the user will pay before for his connection is dropped. The maximum possible bandwidth a user requests is set randomly between 60 kb/s and 160 kb/s. The minimum bandwidth requirement is set to roughly half of the maximum bandwidth. Users re-negotiate their resource requirements with a period of 30 seconds in all the experiments.

The unit bandwidth price charged by the FP policy, and the unit bandwidth usage price charged by CPA, p_u, are both set to 0.23 cents/kb/min. (The holding price p_h in the CPA policy is assumed to be zero, since all simulations are curently performed within a single service class, and interactions between service classes is not considered). Accordingly, in the absence of congestion, the CPA and FP policies have identical revenue under the same traffic flows. After the onset of congestion, the CPA policy applies a variable congestion price p_c, and adaptive users adjust their requirements. In general, the two policies generate unequal revenues in this case. The targeted link utilization of the CPA policy is 90% unless otherwise specified, and congestion pricing is applied when instantaneous usage exceeds this threshold. The price adjustment procedure is also controlled by a pair of parameters, the price adjustment step σ from equation 5 and the price adjustment threshold parameter θ, defined in Section 3.2.3. Unless otherwise specified, values of $\sigma = 0.06$ and $\theta = 0.05$ are used.

5.2 Implementation of RNAP Protocol

We have given a brief overview of the negotiation process through RNAP in Section 2. Two alternative architectures to support resource negotiation through RNAP have been proposed in [26]: a centralized architecture (RNAP-C), and a distributed architecture (RNAP-D). In the centralized architecture, the NRN is a centralized entity in charge of a network domain; in the distributed architecture, the NRN is implemented in a distributed manner at routers on the domain boundary and/or inside the domain.

In this paper, we consider the distributed architecture. An RNAP agent is implemented at each router, in the form of a Local Resource Negotiator (LRN). RNAP messages propagate hop-by-hop along the same path as customer data flows, from the first-hop LRN to the egress LRN, and in the reverse direction. We consider briefly how a sending customer reserves resources for a flow or group of flows to a particular destination address.

1. The HRN (the Host Resource Negotiator through which the sender negotiates with the network) sends a *Query* message to the first hop LRN (FHL). The FHL forwards the *Query* message downstream to the last-hop LRN (LHL). The LHL determines local service availability and a local price for each service, and initiates a *Quotation* message and sends it upstream. Each intermediate LRN verifies local availability of each service, and increments the price by the local price that it computes. The FHL returns the *Quotation* message to HRN. Periodic *Quotation* messages are also sent by the LHL hop-by-hop upstream, as above.

2. The HRN periodically sends a *Reserve* message to the FHL, and receives a *Commit* message in an identical manner to the *Query-Quotation* pair. As the *Commit* message is forwarded upstream, in addition to the committed price being incremented at each router, the incremental charge for each service at that router is added on as well. *Query* and *Reserve* messages may also be sent asynchronously at any time in a similar manner.

The RNAP agent LRN periodically computes a set of prices based on resource requests through a link. It also maintains state information for each RNAP session at the node. Since we study the resource negotiation at call level, we do not simulate actual data traffic. The state information is based only on customer requests and not on actual traffic. The state information is mainly the bandwidth request of each user belonging to a single service class, and the available link bandwidth for that service class.

The RNAP framework allows us to stuy the feasibility of several features: the periodic RNAP negotiation process including resource negotiation and pricing and charging; the stability of the usage-sensitive pricing algorithm and its effectiveness in controlling congestion; the adaptation of user applications in response to changes in network conditions and hence in the service price; and the effect of user utility functions on user adaptation and resource allocation.

5.3 Performance Measures

In the simulation, we show the performance of the system for a range of *offered load*. The offered load is defined as the ratio between the total user resource requirement at the bottleneck, and the bottleneck capacity. Under the FP policy, the total user resource requirement is also the actual resource demand from all the users. Under the CPA policy, the total user resource requirement is what the total resource demand would be if there was no resource contention at the bottleneck and the network did not impose an additional congestion-dependent price.

Both economic and engineering performance metrics are of interest in our study. We define the following engineering performance metrics:

Bottleneck bandwidth utilization: The average bandwidth utilization at the bottleneck node is measured by averaging the reserved bandwidth (expressed as a ratio of the link capacity) over all negotiation periods.

User request blocking probability: The user request blocking probability is the percentage of user reservation requests being denied by the system. When the total resource requirement for a service class exceeds its provisioned capacity, the bandwidth request for that service class will be blocked. In the simulation, the whole network capacity is dedicated to one service class, and hence a new session request is blocked when the total bandwidth requests in the system exceeds the total network capacity. Unsuccessful re-negotiation during an ongoing session is not considered as a block, and the old resource reservation will be maintained upon failure of re-negotiation.

We also define the following economic performance metrics:

Average and total user benefit: The user benefit is the perceived value a user obtains through a transmission of a certain bandwidth (which may vary during the transmission due to adaptation by the user) and of a certain duration, calculated using the user's utility function. Clearly, the user obtains no benefit if its connection request is blocked. The average user benefit is the average of perceived benefits obtained by all the users, and the total user benefit is the sum of perceived benefits obtained by all the users.

Network revenue: Network revenue is the total charge paid to the network for all the admitted requests during a simulation.

System Price: Average price of the sytem is quoted differently along different paths, and is decided by the system parameter setup and conditions. In this paer, it is gained by average all the price quoted from the system throughout a simulation time.

User Charge: A user is charged based on its bandwidth requirement and price quoted by the system. Average charge is what a user would be charged in average per minute.

5.4 Design of Experiments

We conducted experiments to study the effect of different network conditions and different types of user behavior. For each experiment, we looked at two sets of results. The first set of results pertained to the performance of the network under the CPA policy was compared with performance under the FP policy. The key objectives were to find if the CPA policy provides performance improvement, and how the improvement is influenced by the operating conditions. The second set of results pertained to the stability of the network, and nature of network dynamics, under the CPA policy. The followings experiments were performed:

Experiment 1: In Section 6.1, we compare the performance of a system using CPA policy and a system using FP policy, under the default conditions and parameter values specified above.

Experiment 2: In Section 6.2, we evaluate the effect of different system control parameters on system performance.

Experiment 3: In Section 6.3, we evaluate the effect of user demand elasticity on the system performance.

Experiment 4: In section 6.4, we vary the number of customers sharing a system and evaluate the effect of the increased multiplexing of session requests under both policies as the number of sessions is increased..

Experiment 5: In section 6.5, we study the system performance when different fractions of the user population are bandwidth-adaptive, and the remaining users maintain fixed bandwidth requests independent of the network price.

Experiment 6: In Section 6.6, we compare the system performance when users only adapt at session set-up, with the default adaptive behavior during the session.

Experiment 7: Generally, a new connection request will be blocked if the total requested resource exceeds the available resource. However, a session can indicate its minimum bandwidth requirement along with the preferred bandwidth. In section 6.7, we study the effect of a *partial admission* policy in which the connection is admitted by the network even if the requested bandwidth cannot be allocated, provided the available bandwidth is greater than the minimum bandwidth requirement of the user.

Experiment 8: In Section 6.8, we study the performance of the CPA and FP policies in a more general network topolgy in which multiple bottlenecks can occur simultaneously, and interact with each other.

6 Results and Discussion

In this section, we show simulation results from the set of experiments described in section Section 5.1.

6.1 FP Policy versus CPA Policy

We first compare the performance under the FP policy and the CPA policy, with the default conditions specified in section 5.-. Figs. 3 (a)-(d) depict the results of the simulations:

1. Fig. 3 (a) shows the variation of the utilization at the bottleneck link of topology 1, as a function of the offered load, expressed as a fraction of the link capacity. The network utilization under FP policy increases continuously with the increase of offered load. The utilization of CPA policy initially increases with the increase of the offered as expected, and then saturates at the targeted reservation level of 0.9 as the offered load increases beyond a threhold, 1.1 in this experiment. This is as expected, since the objective of the CPA policy is to provide the users the incentive to back off their individual resource requirements in period of resource contention so that the total resource demand remain within the targeted level. We note that the network utilization is lower than the offered load after request-blocking starts at a load of 0.7, and that this corresponds to a progressive increase in the unit-bandwidth price charged by the CPA policy beyond an offered load of 0.7 (Fig. 4 (c)).

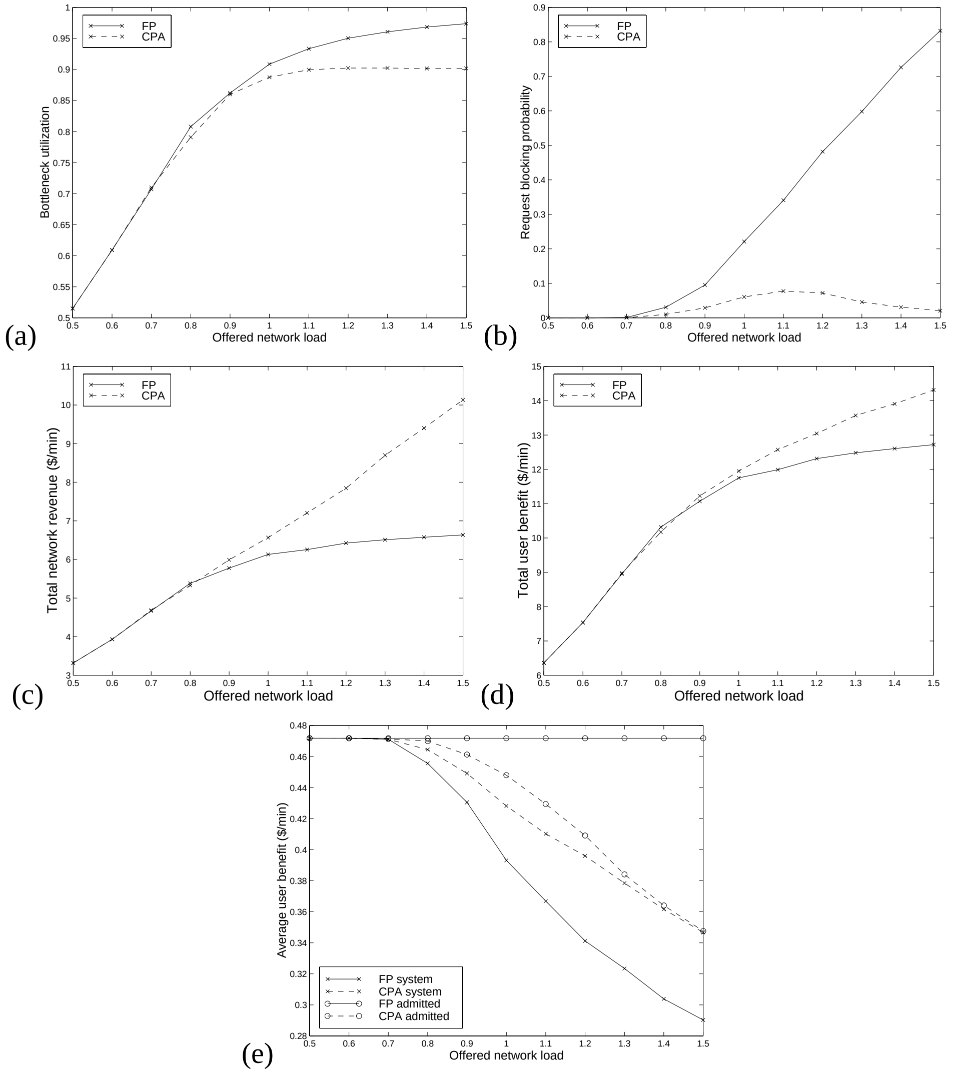

Figure 3: Performance metrics of CPA and FP policies as a function of offered load: (a) bottleneck utilization; (b) blocking probability; (c) total network revenue; (d) total user benefit; (e) average user benefit.

2. Both policies admit all connections until the total link capacity is saturated. Fig. 3 (b) indicates that the blocking probability of FP scheme increases almost linearly as the offered load increases beyond offered load 0.9, while the blocking rate of CPA increases initially and then starts to decrease after reaching a maximum at offered load 1.1. This is because the price adjustment step is proportional to the excess bandwidth above the targeted utilization and increases progressively faster with offered load at higher loads, and the user bandwidth request decreases proportionally with the price according to the general utility function of Section 4.3. The blocking probability of FP policy is almost 40 times bigger than that of the CPA policy at the heaviest load.

3. Fig. 3 (c) compares the network revenue gained under both FP and CPA policies as a function of the offered load. The FP policy flattens out after the onset of request-blocking, indicating that the average number of accepted connections increases slowly beyond this point. With the CPA policy, the revenue increases more than linearly after the network utilization saturates at the targeted level. The loss of revenue due to the scaling down of individual bandwidth requests is more than offset by gains due to the admission of more connections and the increase in the congestion price.

4. Fig. 3 (d) shows the total user benefit gained under the two policies. The user benefit flattens out for both policies after the onset of request blocking. The total benefit gained under CPA is higher than that under FP beyond this point, and the difference increases as the offered load increases. As illustrated in Section 4.3, and discussed in Section 5.1, there is a potential opportunity cost associated with a request being blocked. The decrease in perceived benefit per connection of CPA due to the reduction of bandwidth is offset by the increase in the number of admitted connections, each of which receives an “opportunity”. In effect, the CPA policy allows the network bandwidth to be used more efficiently under high loads.

5. Fig. 3 (e) shows the average perceived benefit per user against offered load. For the FP policy, individual user requests do not depend on the offered load, and consequently, the average benefit per *admitted* user is independent of offered load. However, a progressively smaller fraction of users is admitted by the FP policy as offered load increases. Therfore, the average perceived benefit across all users decreases sharply with the load. The CPA has a much smaller blocking probability at high loads, and therefore gives a higher average perceived benefit at these loads. This should serve as an incentive for users to choose the CPA policy over the FP policy.

We now consider the dynamics of the system price, user bandwidth demand, and user expenditure during the simulation. The results are shown in Figs. 4 (a)-(e).

1. Figs. 4 (a) and (b) show the dynamic variation of the system price and user bandwidth demand respectively at three different levels of offered load. The bandwidth demand is shown for an “average” user, that is, one whose minimum and maximum bandwidth requirements are averages of the corresponding requirements of the user population. The price and bandwidth are nearly static at a load of 0.8, and are adjusted more frequently at higher offered loads, due to the more frequent arrival and departure of users.

   Figs. 4 (c) and (d) show the average and standard deviations of the system price and user bandwidth demand as a function of the offered load. The standard deviation in both figures shows the same trend as the blocking rate of Fig. 3 (b), an increase to a certain level and then a decrease. Initially, the price and demand deviations increase as load increases due to the more aggressive congestion control. At heavy loads, more sessions are admitted, though with smaller average bandwidth. The increased multiplexing of user demand smooths the total demand, and therefore reduces fluctuations in the price

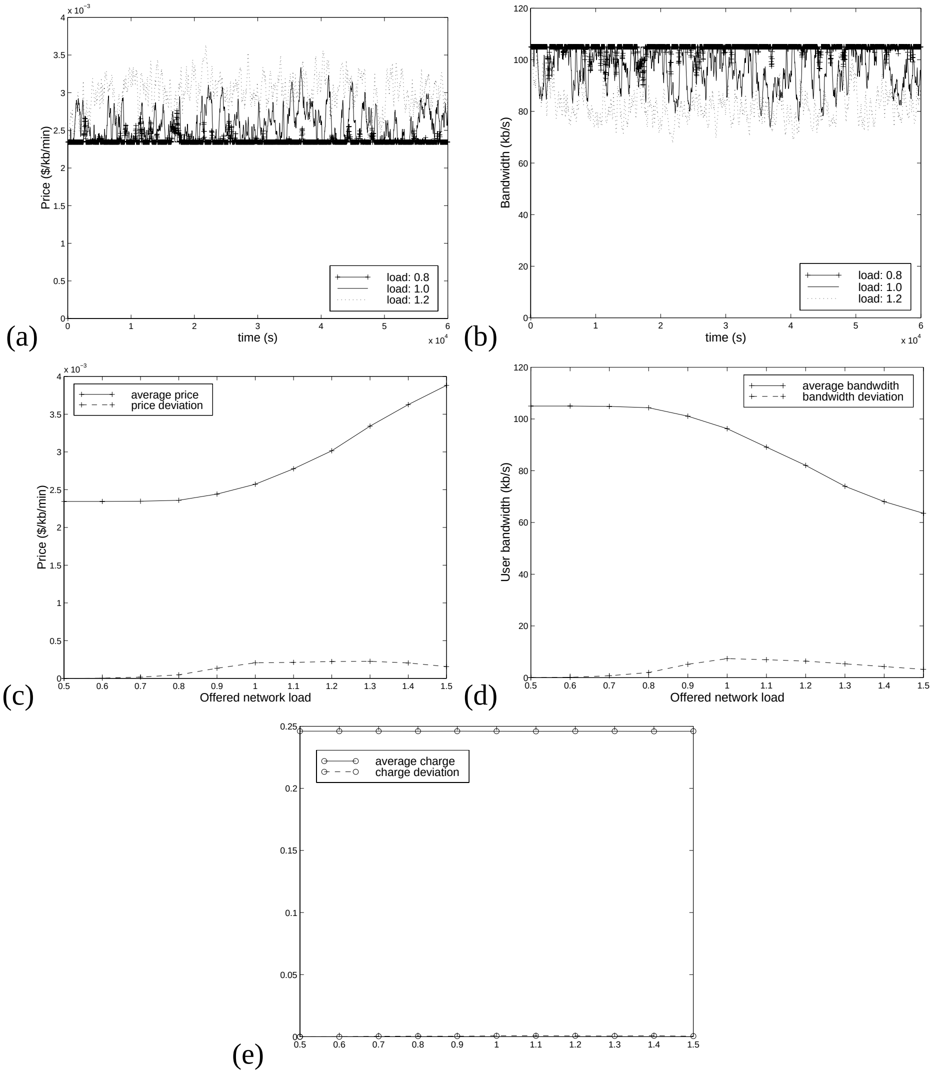

Figure 4: System dynamics under CPA: variation over time of system price (a), and average user demand (b), at on offered load of 1.2; time-average and standard deviation of system price (c), average user demand (d), and average user expenditure (e), plotted against offered load.

and correspondingly in the user demand. Thus, the dynamics actually improve after the total offered load increases to a certain level.

From the perspective of the user, the session cost (expenditure) and application level QoS performance are the most significant metrics. Fig. 4 (e) shows the average expenditure over a session, and the standard deviation, for the "average" user. Evidently, when the users adapt under the example utility function of Section 4.3, the user can operate at a stable cost, and therfore under a fixed budget, meeting one of the fundamental goals of demand adaptation.

The total variation in price over a range of loads also depends on the basic usage price and holding price values. In practice, the usage price and holding price should be set to reflect the long term user demand for different service classes, so that demand fluctuations above the congestion threshold are short-term and infrequent, and congestion pricing is only occasionally employed to smooth out traffic peaks. We are still studying the interaction of long term network resource provisioning with the short term network resource negotiation.

The results in this section indicate that the CPA policy takes advantage of application adaptivity for significant gains in network availability, revenue, and perceived user benefit, relative to the fixed-price policy. The congestion-based pricing is stable and effective. If the nominal (uncongested) price is set to correctly reflect long-term user demand, the congestion-based pricing should effectively limit short-term fluctuations in load.

6.2 Variations of Network Control Parameters

In this section, we study the impact of certain network control parameters on the network and user metrics. The parameters are: the congestion control threshold (or targeted link utilization) beyon which the congestion-dependent price component is imposed; the price adjustment factor σ in equation 5, used to control the rate at which a congested link is brought back to the targeted utilization; and the price adjustment threshold parameter θ, defined in Section 3.2.3. The parameters are varied one at a time in the following three subsections, with the other two parameters set to the default values defined in Section 5.1.

6.2.1 Effect of Congestion Control Threshold

As shown in Figure 5 (a), for three different values of the threshold parameter ρ, the CPA policy limits the average link utilization to ρ.

Figure 5 (b) shows that the blocking rate depends strongly on the targeted utilization, decreasing by a factor of 140 when ρ decreases from 0.95 to 0.85. In general, the total network revenue does not depend strongly on the target utilization (Figure 5 (c)). A lower target utilization causes a lower user demand (Figure 6 d), but also a correspondingly higher system price (Figure 6 c) at the same offered load.

Figure 5(c) shows that at moderate loads, maximum user benefit is obtained for the middle value of the target utilization, 0.90. At a lower congestion threshold, the user bandwidth demand is driven too low by the congestion pricing mechanism, and at a higher congestion threshold, the blocking rate increases, and fewer users are admitted. At very high offered loads, the effect of the reduced user demand dominates, and the user benefit is highest at the highest target utilization.

As expected, Figs. 6 (a) and (b) indicate that both the price and user demand are adjusted more frequently with the decrease of the target utilization. Figs. 6 (c) and (d) show that a lower target utilization also results in a larger standard deviation of the system price and bandwidth demand due to the more aggressive congestion control.

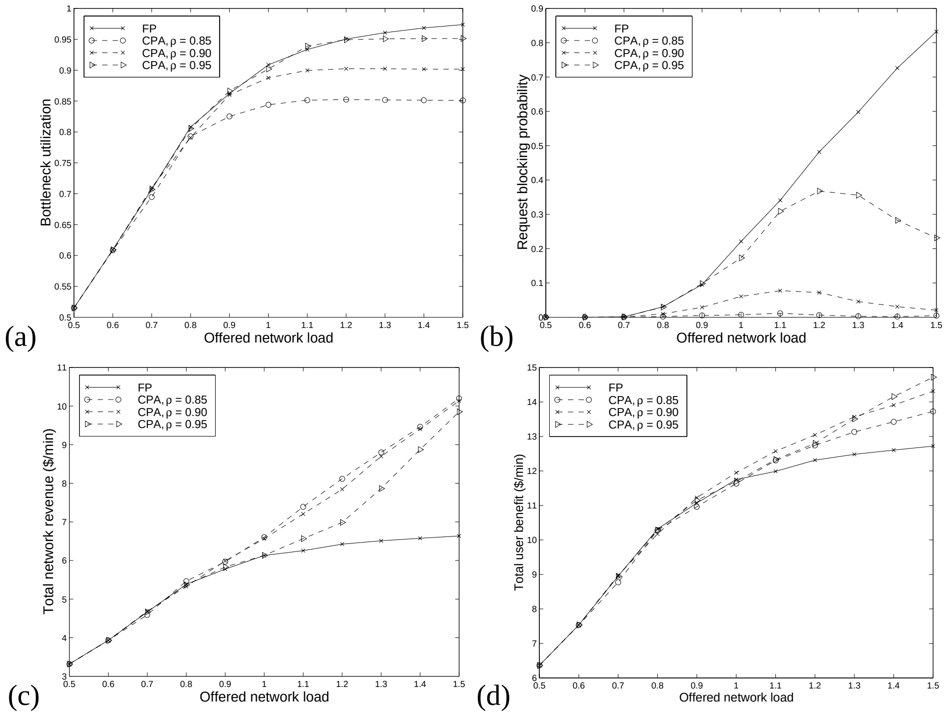

Figure 5: Performance of CPA and FP policies at different values of target congestion control threshold ρ: (a) bottleneck utilization; (b) blocking probability; (c) total network revenue; (d) total user benefit.

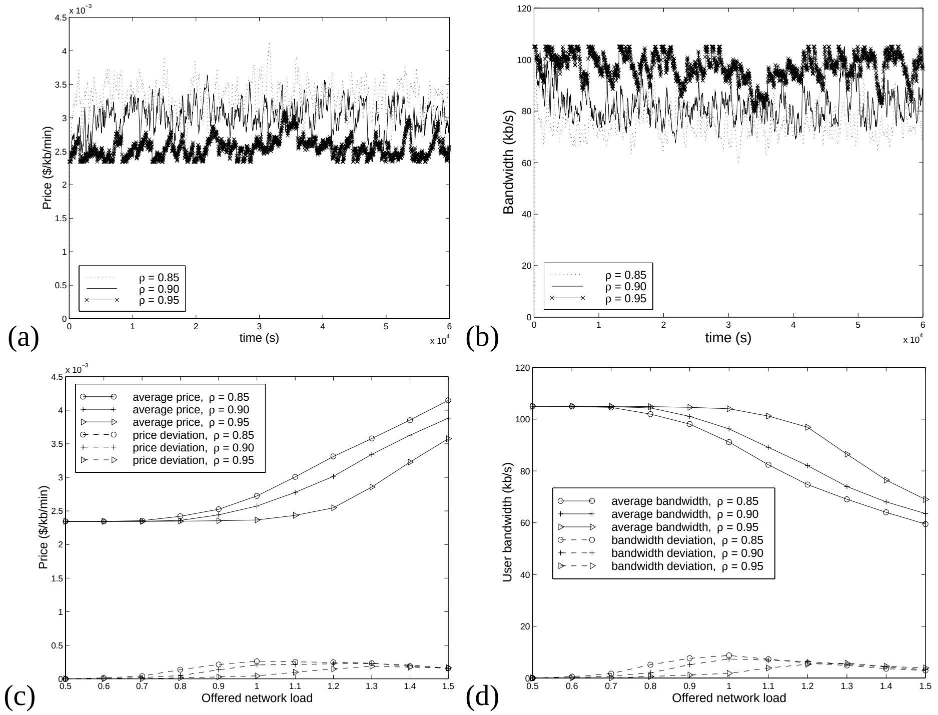

Figure 6: System dynamics at different values of the congestion control threshold: variation over time of system price (a), and average user demand (b), at an offered load of 1.2; time-average and standard deviation of system price (c) and average user demand (d), plotted against offered load.

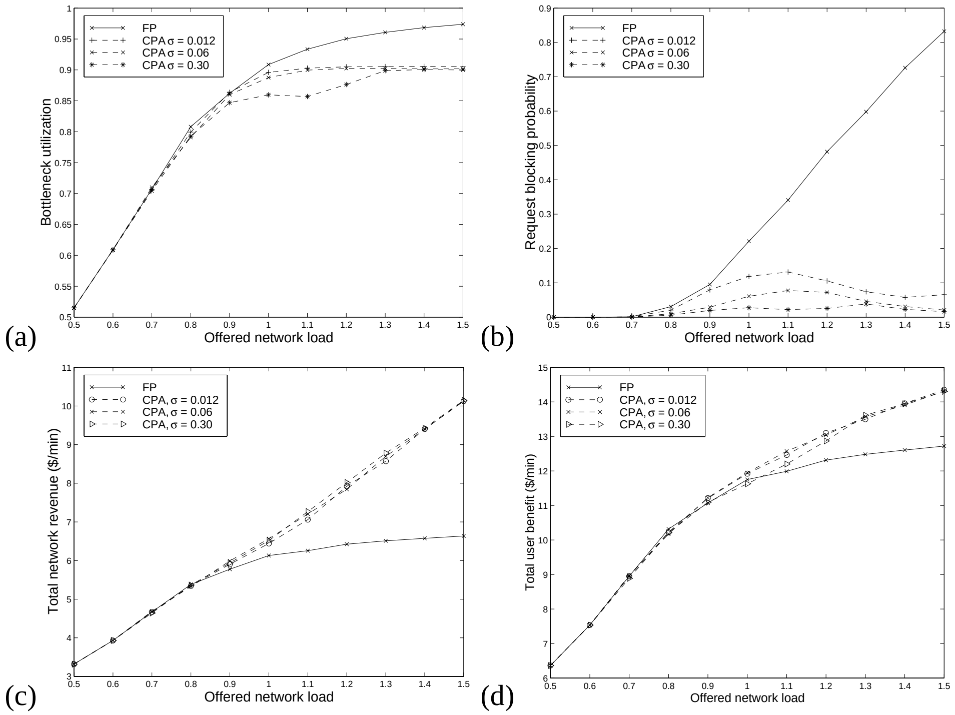

Figure 7: Performance of CPA and FP at different values of σ: (a) bottleneck utilization; (b) blocking probability; (c) total network revenue; (d) total user benefit.

The results in this section indicate that an appropriate target utilization should be selected - under the simulated conditions, 0.90 appears to be a reasonable value. The user benefit decreases if the target utilization is either too low or too high. Also, with too low a target, demand fluctuations are higher, too high a targeted level, results in a high blocking rate.

6.2.2 Effect of Price Adjustment Step

In this section, we investigate the influence of the price scaling parameter σ on the network performance, for $\sigma = 0.012$, 0.06 and 0.30.

Fig. 7 (a) indicates that all three values are effective in controling the network load to the targeted level at heavy load. However, at the highest value, 0.30, the network is significantly under-utilized at moderate to high loads, indicating that the pricing algorithm is too aggressive in driving down demand below the congestion threshold. This is evident in Fig. 8 (c) which shows the network price to be significantly higher in the same load range for σ = 0.30. Fig. 7 (b) shows that as expected, the blocking probability decreases with increasing σ. The small blocking probability at $\sigma = 0.30$ cannot compensate for the under-utilization in determining the total user benefit, however, and the total benefit is significantly smaller at $\sigma = 0.30$ than at the two lower values (Fig. 7 (d)). The opposing effects of a low utilization and high network price (Fig. 8 (c)) at moderate to high loads for $\sigma = 0.30$ results in a slightly higher revenue compared to the two smaller values of σ, as shown in Fig. 7 (c).

Fig. 8 (a) and (b) shows that the price and user bandwidth are adjusted more frequently with a larger σ. The standard deviations of the price and average user bandwidth demand also increase progressively with a larger σ (Figs. 8(c) and (d)).

At low loads (no congestion) and very high loads, all three values of σ result in similar average levels of price and user demand, but at intermediate loads, the highest value of σ results in a much higher price and

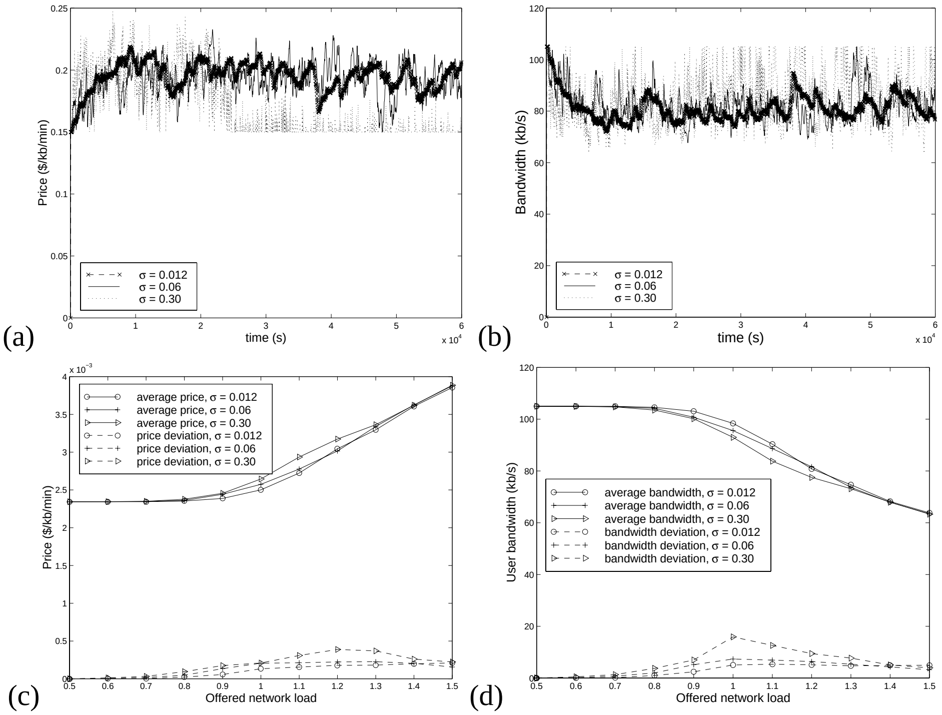

Figure 8: System dynamics at different values of σ: variation over time of system price (a), and average user demand (b), at on offered load of 1.2; time-average and standard deviation of system price (c) and average user demand (d), plotted against offered load.

lower user bandwidth demand, corresponding also to the lower utilization at these loads (Fig. 7 (a)).

From the results above, we see that increasing σ significantly reduces the blocking probability. Too large a value of σ results in network under-utilization at offered loads close to the target utilization, and also results in large network dynamics. Under our simulations, $\sigma = 0.06$ appears to be roughly optimal, and $\sigma = 0.30$ is clearly too high.

6.2.3 Effect of Price Adjustment Threshold

Figs. 9 (a)-(d) show user and network metrics against offered load with θ set to 0.5, 0.05, and 0.005, corresponding to progressively smaller excess demand thresholds before congestion control is activated. In all four figures, the two smaller values of θ correspond to very similar characteristics, except that $\theta = 0.005$ gives a slightly more well-controlled utilization at very high loads. In general, reducing θ to 0.005 does not result in significantly different performance compared to the default value of $\theta = 0.05$ used in earlier experiments. With $\theta = 0.5$, congestion pricing and user demand adaptation are barely initiated and utilization cannot be limited to the target value (Fig. 9 (a)). Therfore, the blocking probability, revenue, and total user benefit are all somewhat better than the performance obtained with FP, and much worse than that obtained with the lower values of θ.

6.3 Effect of User Demand Elasticity

In the previous simulations, as mentioned in section 5, user utility functions of the form of equation **??** were used, with the elasticity factor w and the minimum and maximum bandwidth requirements uniformly distributed. In this section, we study the effect of the user demand elasticity factor w on the system performance. A smaller value of w corresponds to a more elastic demand, since the bandwidth-dependent

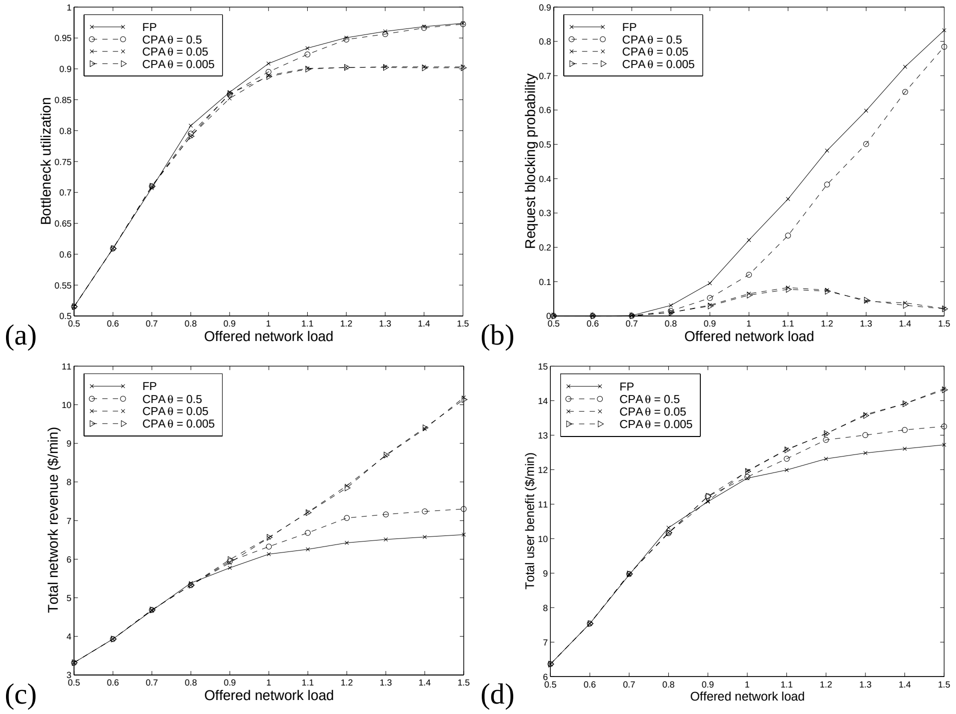

Figure 9: Performance of CPA and FP at different values of θ: a) bottleneck utilization; b) blocking probability; c) average net user benefit; (d) total net user benefit.

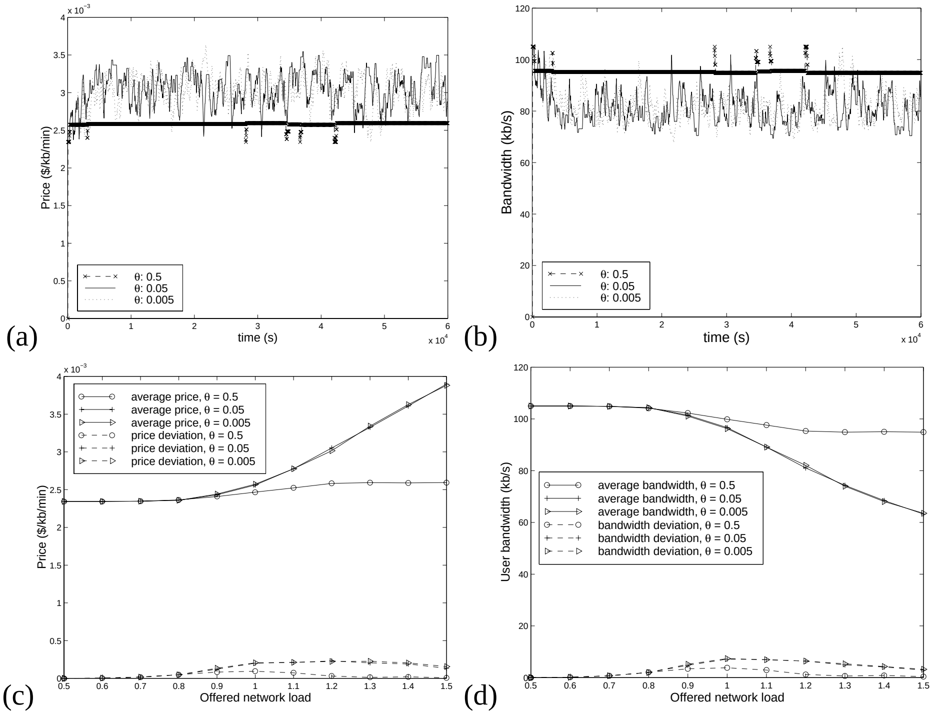

Figure 10: System dynamics at different values of θ: variation over time of system price (a), and average user demand (b), at on offered load of 0.9;

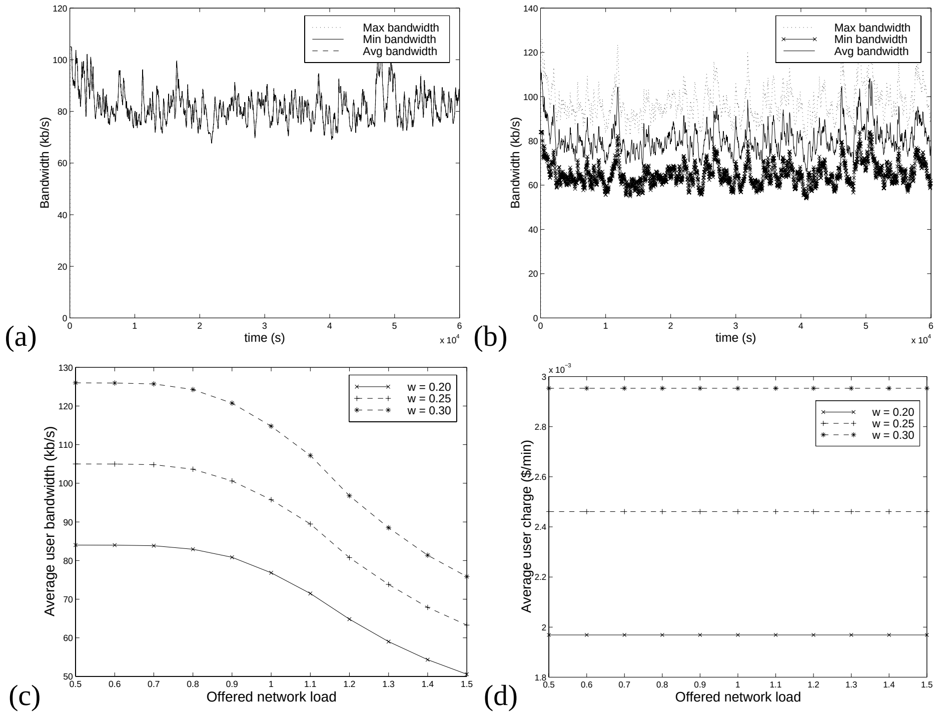

Figure 11: Effect of the elasticity factor w on bandwidth allocation and user expenditure: (a) minimum, maximum and average requested bandwidth when users have the same utility function; (b) minimum, maximum and average requested bandwidth when users have three different utility functions, with w set to 0.20, 0.25, and 0.30 $/min. respectively; (c) average bandwidth reserved by users with the three different values of w; (d) average expenditure of users with the three different values of w

component of the utility is smaller, and the user can reduce its bandwidth request in response to a price increase with only a small decrease in utility. (As explained in Section 4.3, w also represents a user's willingness to pay for bandwidth). We study the bandwidth and charge sharing among users with different utility functions, with w randomly set to the average default value of $0.25/min. times 0.8, 1.0, or 1.2, keeping the other utility function parameters constant.

1. Fig. 11 (a) and (b) shows the instaneous bandwidth sharing among users when all users have utility functions with the same elastcity, and three different elasticities, respectively. The offered load is 1.2. In both cases, we show the maximum, minimum as well as average bandwidth for all the active sessions in the system. Fig. 11 (a) indicates that when all the users have the same utility function, they share the bandwidth fairly at any time, and the maximum, minimum and average bandwidth demands coincide. Fig. 11 (b) indicates that when the users have different demand elasticities (w = 0.20, 0.25, and 0.30 $/min), their bandwidth shares span a range of approximately 50 kb/s.

2. Fig. 11 (c) shows the average bandwidth allocated to users with the three different utility functions. At a given offered load, the bandwidth users obtain is proportional to their individual demand elasticity factor or willingness to pay (that is, inversely proportional to the bandwidth demand elasticity). Fig. 11 (d) shows that average user expenditure is also proportional to w, and indepedendent of the offered load.

Evidently, users with more elastic requirements are more sensitive to price changes and reduce their resource requirements faster as price increases, therefore receiving a smaller share of the bandwidth. In effect, users with more stringent bandwidth requirements choose to pay a higher charge and "borrow" bandwidth from users with more elastic requirements when the network is congested.

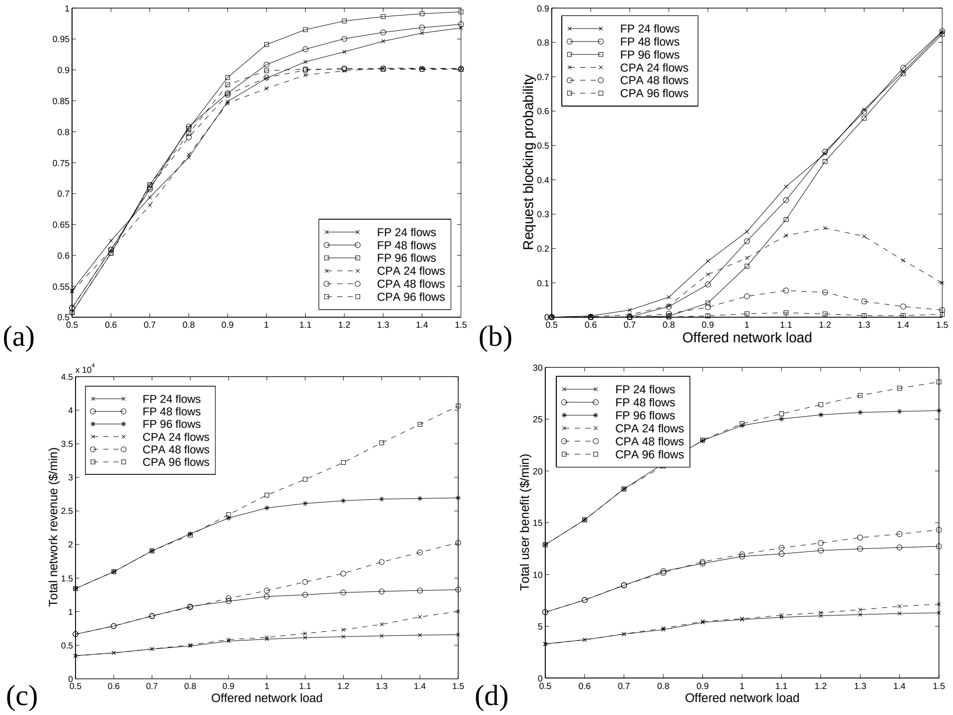

Figure 12: Performance of CPA and FP with different number of customers sharing the system: (a) bottleneck utilization; (b) blocking probability; (c) network revenue; (d) total user benefit.

6.4 Effect of Session Multiplexing

In the previous simulations, we set N_u, the number of sending users at each end node, to 4, thus restricting the maximum number of flows in the system to 48. In this section, we set the maximum number of flows in the system to 24, 48 and 96, by setting N_u to 2, 4 and 8 separately. We keep the network topology and user utility distributions unchanged, but scale the link capacity proportionally with the maximum number of flows. We compare the system and user metrics under the three cases.

1. Fig. 12 (a) shows that the overall link utilization under FP increases as the number of connections increases, at a given offered load. The link utilization under CPA also increases with the number of flows at moderate to high loads, but the utilization is eventually limited to the targeted level. Fig. 12 (b) shows that as the number of connections increases, the blocking probability decreases under both FP policy and CPA policies. This is because that the larger number of connections lead to better traffic multiplexing and hence more efficient use of network bandwidth.

   However, the improvement is much more pronounced under the CPA policy than under the FP policy, particularly when the network is saturated. Under CPA, the blocking rate with 96 connections is up to 50 times smaller than that with 24 connections.

2. Figs. 12 (c) and (d) show that in all three cases, both the network revenue and total user benefit under CPA are better than those under FP. The gain increases as more users share the network, due to the big improvement in blocking probability under CPA with a larger number of connections.

Fig. 13 depicts the price and demand dynamics as the network scales. Figs. 13 (a) and (b) show that the frequency of price and demand adjustment do not change appreciably with number of connections. As expected, both price and user bandwidth demand become smoother as more users share the network, and this is confirmed by the smaller standard deviations shown in Figs. 13 (c) and (d).

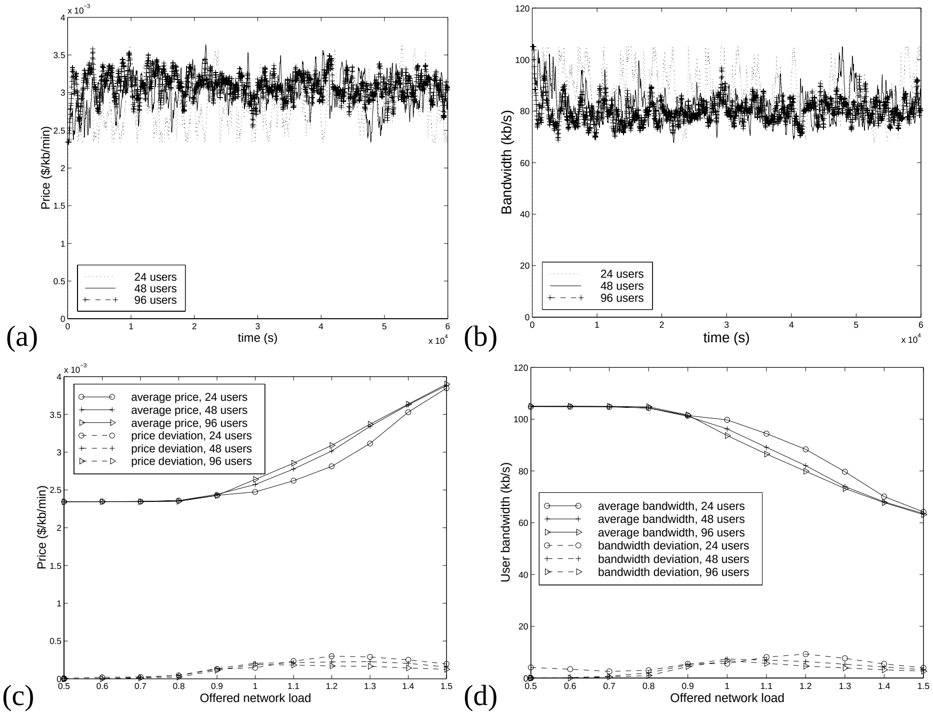

Figure 13: System dynamics with different number of customers sharing the same bottleneck: variation over time of system price (a), and average user demand (b), at an offered load of 1.2; time-average and standard deviation of system price (c) and average user demand (d), plotted against offered load.

The results in this section indicate that performance of the CPA policy further improves as the network scales and more connections share the resources.

6.5 Adaptive and Non-adaptive Users

In this section, we consider the environment where some users adapt their bandwidth requests under the CPA policy, while others maintain fixed service requests even when the congestion price is imposed. The latter group represents users with a willingness to pay that is high enough to maintain their maximum bandwidth requrements even at the highest price charged by the network. Under the CPA policy, the congestion-dependent price component increases until usage is driven down to the target utilization. We restrict the maximum price to 0.49 cents/kb/min for this set of simulations, so that the price does not increase indefinitely when 100% of the users are non-adaptive.

1. Fig. 14 (a) shows that when some users do not adapt, the network utilization can no longer be kept at the targeted level when the offered load exceeds a certain threshold. This threshold decreases as the ratio of the adaptive requests decreases. Fig. 14 (b) shows that corresponding to the sharp increase in utilization beyond a certain load threshold, the blocking rate also increases sharply. We can also see that the network blocking probability reduces significantly even when some of the flows are adaptive, compared to when none of the flows is adaptive. Therefore, even if some of the users are adaptive, all the users can receive improved perfirmance, particularly up to a certain threshold load.

2. Fig. 14 (c) shows that the total revenue increases as the proportion of non-adaptive users increases, since more users maintain their service request at high loads, and pay the higher price. Fig. 14 (d)

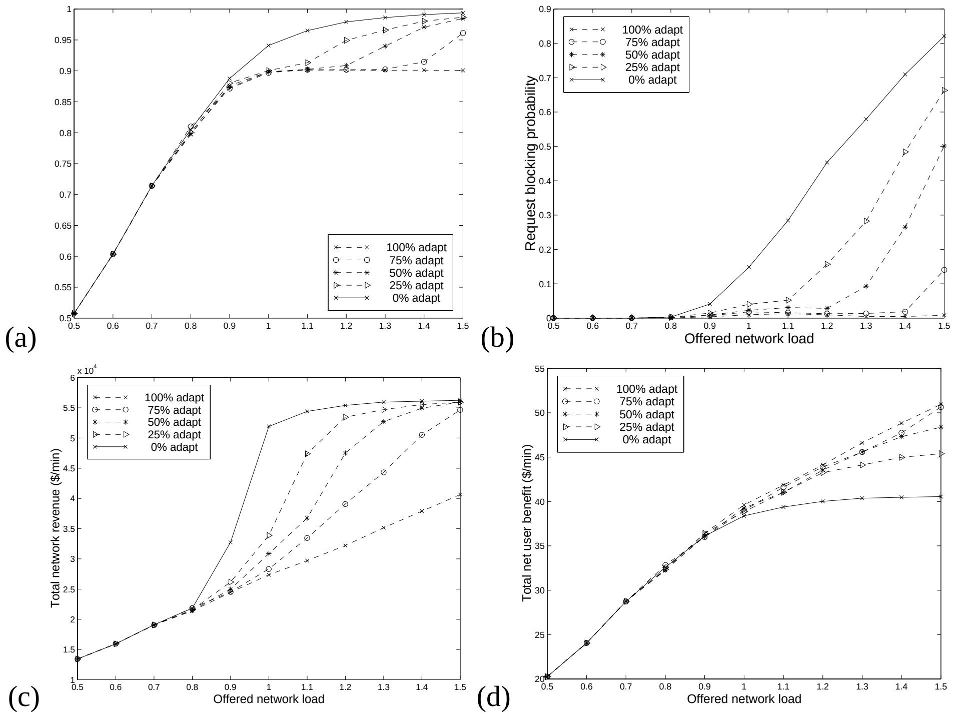

Figure 14: Performance of CPA when only some of the users adapt their bandwidth requests: (a) bottleneck utilization; (b) blocking probability; (c) network revenue; (d) total user benefit, plotted against offered load.

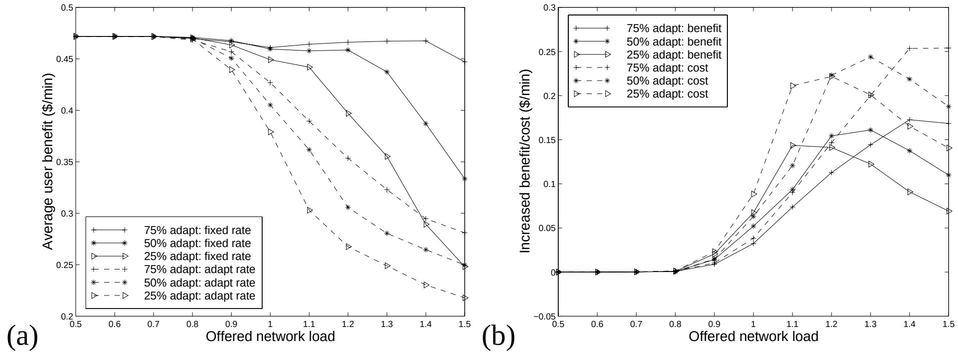

Figure 15: Relative benefits and expenditures of adaptive and non-adaptive users, under three different proportions of non-adaptive to adaptive users: (a) average user benefit for the two groups of users; (b) average benefits and average expenditures of non-adaptive users relative to average benefits and expenditures of non-adaptive users

shows that at a given load, the total user benefit increases with the number of adapative users, an outcome of the lower blocking probability.

3. In CPA policy, an adaptive user selects a request x^* to maximize its net benefit, i.e., $U(x^*) - C(x^*) \geq U(x') - C(x')$. Therefore, $U(x^*) - U(x') \geq C(x^*) - C(x')$. At equilibrium, any improvement in service quality is offset by the increased cost, and any decrease in cost obtained is offset by the resulting decrease in service quality. In this simulation, we assume that all the users (adaptive and non-adaptive) have the same utility function, and hence their perceived benefits and surplus can be compared.

   Fig. 15 (a) shows that a non-adaptive user receives a higher perceived benefit (corresponding to a higher quality of service) compared to a non-adaptive user, although the absolute benefit decreases with increasing load, and also with a decrease in the proportion of adaptive users. Fig. 15 (b) shows that for adaptive users, the lower received benefit relative to non-adaptive users ismore than compensated by a lower cost. The cost paid by users decreases proportionally with bandwidth, while the received benefit decreases less sharply because of the opportunity benefit asssocaited with simply holding a connection. Consequently, adaptive users receive a higher net than non-adaptive users. This may be regarded as a higher "value for money" and is an economic reward for user adaptation.

The above results seem to indicate that the peformance benefits of CPA decrease as fewer users adapt, which is to be expected. The results do show that even a small proportion of adaptive users may result in a significant performance benefit and better service for the entire user population.

We should also expect CPA to have an additional inherent advantage over the FP policy even when most of the users are non-adaptive. In reality, the usage price shown in Section 3.2 would reflect the estimated long-term network load. The congestion price would be only used to smooth out temporary peaks, and the general usage pattern would result in optimal utilization at the offered usage price. However, a vendor charging a static price (FP) would need to charge a certain premium above this optimal price, as a risk premium, while the CPA policy allows the vendor to operate around the optimal price and use congestion pricing to protect against demand peaks.

6.6 Session Adaptation and Adaptive Reservation

For a given service price and quality, an application can initially determine its optimal bandwidth requirement to maximize its perceived benefit. Some applications, such as multimedia adaptive applications **??**, can also adapt their sending rate during an ongoing multimedia session. Under the RNAP framework, as mentioned in Section 2, users can negotiate and re-negotiate services at any time. In this section, we compare the performance under the two scenarios: bandwidth selection only at session set-up, and ongoing bandwidth adaptation during a session. Clearly, the first scenario is sub-optimal, since the users as a group become less adaptive. We are interested in how this sub-optimality affects the performance metrics under the simulation conditions.

Fig. 16 (a) shows that initial adaptation results in a slightly lower network utilization at moderate-to-high loads, about 3-5% smaller than the utilization under ongoing adaptation. This is because if a session arrives during a traffic peak, it will request a smaller bandwidth, which will not be scaled back after the the demand is driven down. Fig. 16 (b) shows that as expected, adaptation during a session allows for more efficient bandwidth usage and the blocking probability is reduced by half. Fig. 16 (c) shows that the total network revenue increases slightly with initial adaptation, and Fig. 16 (d) shows a decrease in the total user benefit, arising from the higher blocking rate.

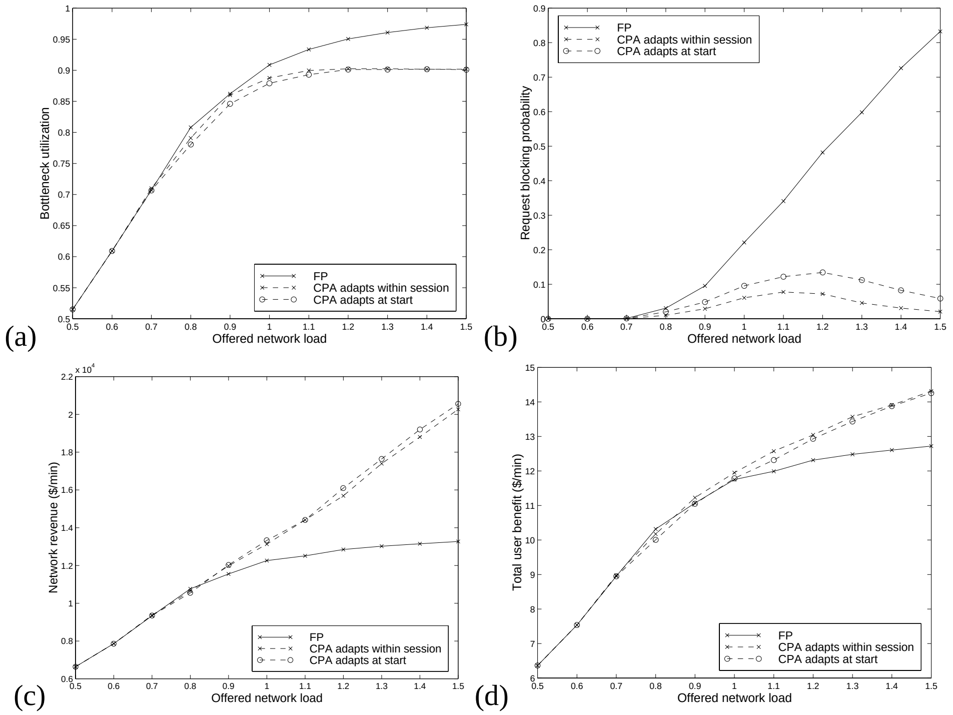

Figure 16: Performance when CPA users select bandwidth only at session set-up, compared with performance when they continue to adapt during the session (a) bottleneck utilization; (b)blocking probability; (c) network revenue; (d) total user benefit.

6.7 CPA with Partial Admission

In the CPA policy framework, a new user requests a certain bandwidth depending on its utility function and the current network price. The network either admits or denies the request depending on the availability of bandwidth at each link. As described in Section 2, RNAP also allows the users to transmit data rate information. With respect to the sender HRN, the data rates represent the minimum and maximum sending rates the sender is willing and able to transmit. With respect to the receiver HRN, these rates indicate the minimum and maximum data rates the receiver is willing and able to receive. The minimum and maximum data rate from a sender indicate its demand scalability. If the user allows the network to intercept this information, the network can admit the user reservation request when the available bandwidth is less than the current required bandwidth, but is greater than the user's minimum requirement. We call this kind of admission *partial admission*. Since the users adapt to changes in network price continuously, after a user has received a partial admission, it is likely to obtain its fair bandwidth share (based on its utility function) as part of the adaptation process.

Figs. 17 (a)-(d) show that the CPA policy with partial admission does result in a significant decrease in the blocking probability (by about 30%) relative to the default CPA policy, indicating that the available bandwidth is now used more efficiently. Consequently, there is a small but noticable improvement in utilization, network revenue and total user benefit after the onset of congestion.

6.8 CPA Performance with Traffic Interactions from Different Paths

In the experiments above, we studied the performance of CPA under different loads, with the variation of different parameters, and under different traffic behaviors, but with the traffic always sharing a common bottleneck. In this section, we assume network topology 2 in Fig. 2, with the potential for multiple bottlenecks to exist, and for these bottlenecks to interact. For instance, rate adaptation by an user due to particular

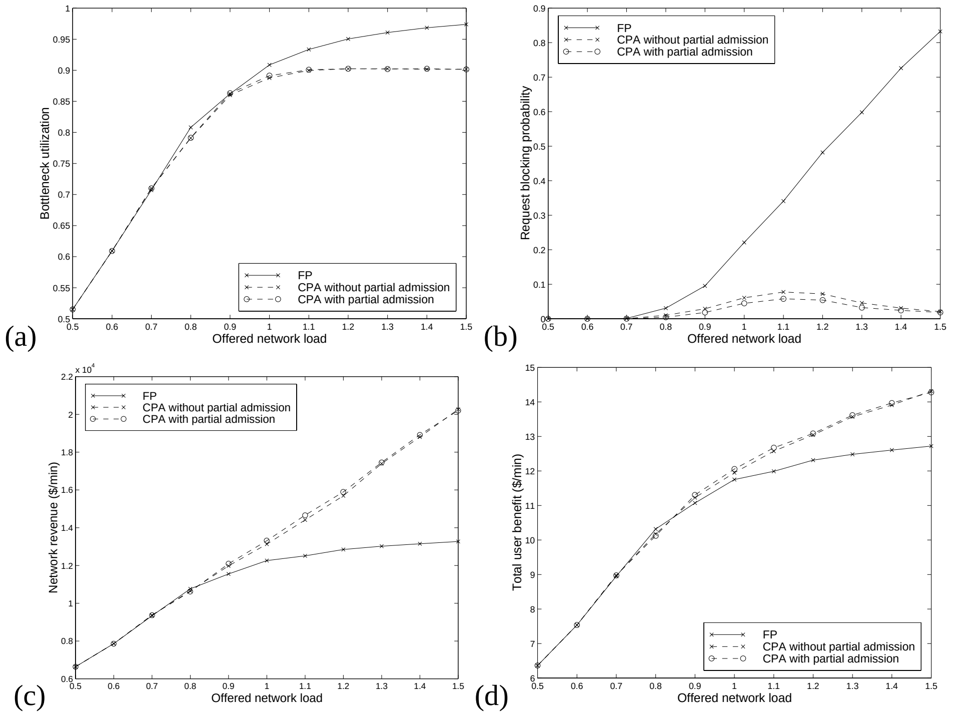

Figure 17: Performance of CPA with and without partial admission: (a) bottleneck utilization; (b) blocking probability; (c) network revenue; (d) total user benefit.]

bottleneck may influence the resource availability at another bottleneck. For example, in topology 2, when users reduce the demand on the path from A1 to A12 as result of congestion in link A5 and A4, it also reduces the demand through link B1 and B5.

In the simulation, traffic is generated symetrically from all users, as described in Section 5. The five backbone links are the potential bottleneck links Note that in reality, the backbone links are normally overprovisioned. We target the backbone links to be bottlenecks only for the convenience of simulation. We monitor the utilization at one of the backbone links, and calculate all the other parameters across the whole network. Fig. 18 (a) and (b) shows that both the utilization and blocking probability show trends similar to those for a single bottleneck, except that the variation of the utilization and blocking probability is not as smooth due to the coupling of the traffic between different paths. The utilization is still well controlled to the targeted threshold under CPA. Figs. 18 (c) and (d) show a corresponding improvement given by the CPA policy with respect to the network revenue and total user benefit.

6.9 Other Mechanisms to Reduce Network Dynamics

Other than network parameter settings, user adaptation behavior too has an effect on the traffic dynamics seen by the user, and by the network. A user can set a minimum bandwidth adaptation increment, and communicate a new bandwidth request to the network only when the new calculated bandwidth requirement changes the existing bandwidth requirement by more than the minimum increment. This reduces the frequency of bandwidth adjustment at the cost of a sub-optimal bandwidth (in terms of perceived value). The extreme case is when the adaptation only occurs at the beginning, as is shown in Section 6.6.

A somewhat similar scenario can be envisoned in a core network, in which bandwidth reservation is carried out by other network providers rather than by individual users. In this case, the customer providers can change their bandwidth requests in multiples of a large block of bandwidth, only when the user flow-

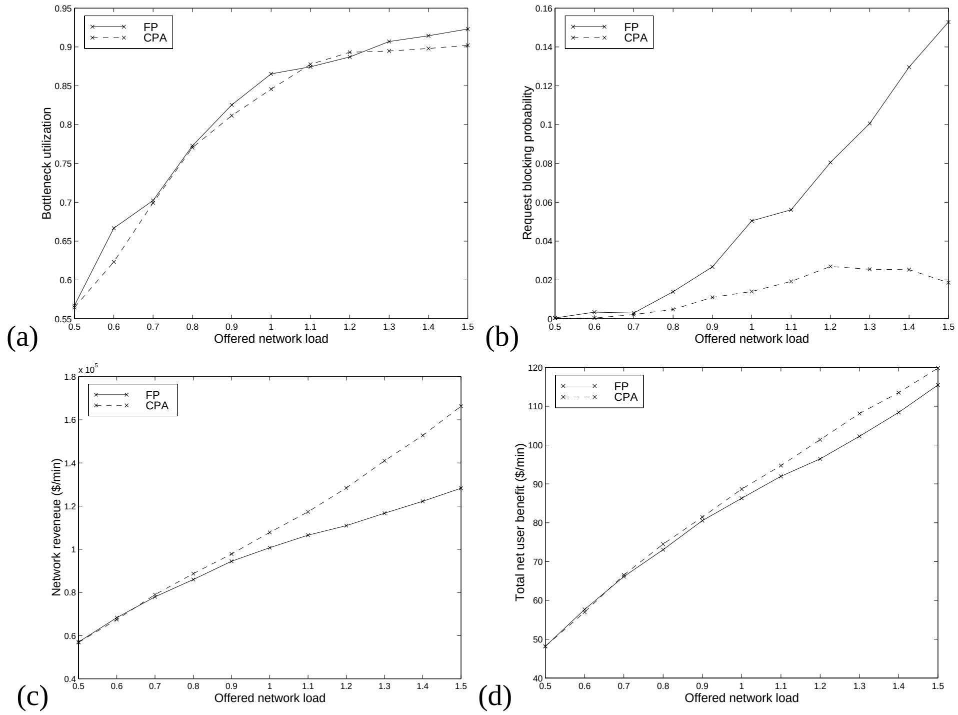

Figure 18: Performance metrics of CPA and FP policies as a function of offered load using topology 2: (a) bottleneck utilization; (b) blocking probability; (c) total network revenue; (d) total user benefit.

level demand to the customer providers changes by a certain increment. This can reduce both network dynamics and signaling overhead in the core network, and has been discussed in greater detail in [26].

7 Related Work

In this section we briefly discuss related research work in three main areas: resource reservation and allocation mechanisms; adaptive applications; billing and pricing in the network.

7.1 Resource Reservation and Allocation

Current research in providing QoS support in the Internet is mainly based on two architectures defined by IETF: Per-flow based *integrated services* (int-serv) [7], and class-based *differentiated service* (diff-serv) [21]. In general, RSVP and the implementations of diff-serv lack integrated mechanisms by which the user can select one out of a spectrum of services, and re-negotiate resource reservations dynamically. They also do not integrate the pricing and billing mechanisms which must accompany such services.

Resource allocation schemes based on perceived-quality have been studied in [18][2][19]. These studies were limited to a local system, and did not address the interaction of the local system with a large network. Liao [6] allocates resources to achieve equal perceived quality. We argue that perceived quality does not directly represent the economic value of communications, as discussed below.

7.2 Adaptive Applications

There has been a lot of recent research on adaptation of the sending rates of multimedia applications in response to available network resources [28], which relies on signaling mechanisms such as packet loss rates for feedback. The orientation of these methods is different from ours, since they assume no QoS support and no usage sensitive pricing of network services. The frequent and passive rate adjustment can severely degrade the multimedia quality, and sometimes an application is even not able to maintain its minimum QoS requirement.

7.3 Pricing and Billing in the Network

Microeconomic principles have been applied to various network traffic management problems. The studies in [20][18][15][10][] are based on a maximization process to determine the optimal resource allocation such that the utility (a function that maps a resource amount to a satisfaction level) of a group of users is maximized. These approaches normally rely on a centralized optimization process, which does not scale. Also, some of the algorithms assume some knowledge of the user's utility curves by the network and truthful revelation by users of their utility curves, which may not be practical.

In [9][5][10][11][23], the resources are priced to reflect demand and supply. The pricing model in these approaches is usage-sensitive - it has been shown that usage-sensitive pricing results in higher utilization than traditional flat (single) pricing [9]. Some of these methods are limited by their reliance on a well-defined statistical model of source traffic, and are generally not intended to adapt to changing traffic demands.

The scheme presented in [11] is more similar to our work in that it takes into account the network dynamics (session join or leave) and source traffic characteristics (VBR). It also allows different equilibrium price over a different time period, depending on the different user resource demand. However, congestion is only considered during admission control. Our pricing algorithm has two congestion-dependent components - congestion due to excessive resource reservation (holding cost) and congestion due to network usage (usage cost).

In general, the work cited above differs from ours in that it does not enter into detail about the negotiation process and the network architecture, and mechanisms for collecting and communicating locally computed prices. Some of the work also assumes immediate adjustment of the price in response to the network dynamics, or require the user to maintain a static demand until a optimal price is found, which is not practical. Our work is concerned with developing a flexible and general framework for resource negotiation and pricing and billing, and evaluating the performance benefits of congestion-sensitive pricing and adaptation through simulations, decoupled from specific network service protocols. Our work can therefore be regarded as complementary to some of the cited work.

In [16], a charging and payment scheme for RSVP-based QoS reservations is described. A significant difference from our work is the absence of an explicit price quotation mechanism - instead, the user accepts or rejects the estimated charge for a reservation request. Also, the scheme is coupled to a particular service environment (int-serv), whereas our goal is to develop a more flexible negotiation protocol usable with different service models.

8 Conclusions

We have considered a framework for incentive-driven rate and QoS adaptation. In the framework, users respond actively to changes in price signaled by the network by dynamically adjusting network resource usage by the application, so as to maximize the perceived utility relative to the price, subject to user budget and QoS requirements. We have discussed different pricing models, and outlined a dynamic, congestion-sensitive pricing algorithm. We have also described the user demand behavior based on a physically reasonable user utility characteristic.

The main focus of this paper has been the simulation of the above framework. Through simulations, we have compared the performance of a network under the congestion price based adaptation policy (CPA) with that under a fixed price based policy (FP). We have also studied the stability of the adaptation process, and nature of network dynamics, under the CPA policy. In general, CPA policy takes advantage of application adaptivity for significant gains in network availability, revenue, and perceived user benefit (in terms of the user utility functions), relative to the fixed-price policy. The congestion-based pricing is stable and effective in limiting utilization to a targeted level. If the nominal (uncongested) price is set to correctly reflect long-term user demand, the congestion-based pricing should effectively limit short-term fluctuations in load.

We have investigated the impact of various network control parameters on the network and user metrics. The user benefit decreases if the target utilization is set either too low or too high. Also, with too low a target, demand fluctuations are higher, while too high a targeted level results in a high blocking rate. Increasing the price scaling factor σ (which affects the speed of reaction to congestion) significantly reduces the blocking probability. However, too large a value of σ results in network under-utilization at offered loads close to the target utilization, and also results in large network dynamics. If the price adjustment threshold parameter θ is set too high, there is no meaningful price adjustment and adaptive action. Below a certain level, further reductions in θ do not give performance benefits or disadvantages.

Users with different demand elasticity are seen to share bandwidth fairly, with each user having a bandwidth share proportional to its relative willingness to pay for bandwidth. The results also show that even a small proportion of adaptive users may result in a significant performance benefit and better service for the entire user population - both adaptive and non-adaptive users. The performance improvement given by the CPA policy further improves as the network scales and more connections share the resources.

References

[1] Watson A. and M. A. Sasse. Evaluating audio and video quality in low-cost multimedia conferencing systems. In *Interacting with Computers*, volume 8, page 255, 1996.

[2] T. F. Abdelzaher, E. M. Atkins, and K. Shin. Qos negotiation in real-time systems and its application to automated flight contro. In *IEEE Transactions on Software Engineering*, 1999.

[3] J. Adam. Upgrading the internet. *IEEE Spectrum*, 32(9):24–29, 1995.

[4] J. Altmann, B. Rupp, and P. Varaiya. Internet user reactions to usage-based pricing. In *Proceedings of the 2nd Berlin Internet Economics Workshop (IEW '99)*, Berlin. Germany, May 1999.

[5] N. Anerousis and A. A. Lazar. A framework for pricing virtual circuit and virtual path services in atm networks. In *ITC-15*, December 1997.

[6] G. Bianchi, A.T. Campbell, and R.R.-F. Liao. On utility-fair adaptive services in wireless networks. In *6th International Workshop on Quality of Service (IEEE/IFIP IWQOS'98)*, 1998.

[7] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. Request for Comments (Informational) 1633, Internet Engineering Task Force, June 1994.

[8] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation protocol (RSVP) – version 1 functional specification. Request for Comments (Proposed Standard) 2205, Internet Engineering Task Force, September 1997.

[9] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in computer networks: Motivation, formulation, and example. In *IEEE/ACM Transactions on Networking*, December 1993.

[10] D. F. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks.

[11] E. W. Fulp and D. S. Reeves. Distributed network flow control based on dynamic competive markets. In *Proceedings International Conference on Network Protocol (ICNP'98)*, October 1998.

[12] J. Heinanen. Assured forwarding PHB group. Internet Draft, Internet Engineering Task Force, August 1998. Work in progress.

[13] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding PHB. Internet Draft, Internet Engineering Task Force, February 1999. Work in progress.

[14] S. Jamin, S. J. Shenker, and P. B. Danzig. Comparison of measurement-based admission control algorithms for controlled-load service. In *Infocom*, page 973, Kobe, Japan, April 1997.

[15] H. Jiang and S. Jordan. A pricing model for high speed networks with guaranteed quality of service. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, March 1996.

[16] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz. An embedded charging approach for rsvp. 1998.

[17] C. Lambrecht and O. Verscheure. Perceptual quality measure using a spatio-temporal model of human visual system. 1996.

[18] C. Lee, J. Lehoczky, R. Rajkumar, and D. Siewiorek. A quality of service negotiation procedure for distributed multimedia presentational applications. In *Proceedings of the Fifth IEEE International Symposium On High Performance Distributed Computing (HPDC-5)*, 1996.

[19] C. Lee, J. Lehoczky, R. Rajkumar, and D. Siewiorek. On quality of service optimization with discrete qos options. In *Proceedings of the IEEE Real-time Technology and Applications Symposium*, June 1999.

[20] J. F. MacKie-Mason and H. Varian. Pricing congestible network resources. September 1995.

[21] K. Nichols and S. Blake. Differentiated services operational model and definitions. Internet Draft, Internet Engineering Task Force, February 1998. Work in progress.

[22] P. Reichl, S. Leinen, and B. Stiller. A practical review of pricing and cost recovery for internet services. In *Proc. of the 2nd Internet Economics Workshop Berlin (IEW '99)*, Berlin, Germany, May 1999.

[23] J. Sairamesh. Economic paradigms for information systems and networks. In *PhD thesis*, October 1997.

[24] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. Request for Comments (Proposed Standard) 2212, Internet Engineering Task Force, September 1997.

[25] Hal Varian. *Microeconomic Analysis*. W.W. Norton & Co, 1993.

[26] X. Wang and H. Schulzrinne. RNAP: A resource negotiation and pricing protocol. In *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, pages 77–93, Basking Ridge, New Jersey, June 1999.

[27] X. Wang and H. Schulzrinne. Adaptive reservation: A new framework for multimedia adaptation. In *IEEE International Conference on Multimedia and Expo (ICME'2000)*, New York, NY, USA, July 2000.

[28] Xin Wang and Henning Schulzrinne. Comparison of adaptive internet multimedia applications. *IEICE*, June 1999.

[29] J. Wroclawski. Specification of the controlled-load network element service. Request for Comments (Proposed Standard) 2211, Internet Engineering Task Force, September 1997.

[30] Hui Zhang and Srinivasan Keshav. Comparison of rate-based service disciplines. In *Sigcomm '91 Symposium – Communications Architectures and Protocols*, pages 113–121, Switzerland, September 1991. ACM. also in Computer Communication Review 21(4) September 1991.