Comparative Study of Two Congestion Pricing Schemes - Auction and Tâtonnement

Xin Wang *

Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260, USA

Henning Schulzrinne

Department of Computer Science, Columbia University, New York, NY 10027, USA

Abstract

We evaluate two mechanisms for setting prices in a QoS-enhanced network with congestion-sensitive pricing and resource allocation based on users' willingness to pay. In the first model, the congestion-sensitive component of the price is calculated by tâtonnement, with the price adjusted gradually to drive down the user demand to the supply level of network resources. In the second model, each user submits multiple bids for different bandwidths, each bid expressing its willingness to pay a certain premium for the corresponding bandwidth during congestion. Simulations show that both approaches provide greater network availability, revenue, and perceived user-benefit than a congestion-independent policy. Both approaches have generally similar performance in terms of perceived user-benefit. The tâtonnement-based model obtains higher network revenue than the auction-based model. The auction-based model achieves higher network utilization at a given level of perceived user-benefit, but has higher implementation complexity and longer set-up delay for new connections. The proposed auction-based model serves more users than comparable auction-based schemes, and has less signaling overhead and greater certainty of service availability.

Key words:
Pricing, Congestion, Auction, Internet, QoS, Adaptive

* Corresponding author. Email address: xwang8@cse.buffalo.edu

1 Introduction

The Internet's lack of control over quality of service (QoS) has slowed down the deployment of many new Internet applications, such as multimedia services. Many statistics show that access networks get congested frequently, and the cost of bandwidth (especially leased bandwidth) is not dropping rapidly [1]. It is also difficult to predict user demand, especially due to the rapid deployment of new applications and increasing access speed. Consequently, it makes economic sense to study and develop models that provide adequate QoS assurance and utilize resources more efficiently than a simple best-effort, flat-rate network. Two approaches have been studied recently, one based on providing QoS support through classes of services, and the other based on user adaptation of sending rates in response to network congestion.

Many existing multimedia applications allow the media rate and quality to be adjusted over a wide range in response to network congestion [2]. Generally, the literature assumes best-effort service from the network, which provides no incentive to users to adapt their rates, and allows selfish users to benefit at the expense of cooperative ones. In earlier work [3], we had proposed a pricing model in which service prices are based on QoS (resources consumed) and long-term user demand, and also have a congestion-sensitive component to motivate rate and service adaptation by applications with elastic demand. The network is provisioned to provide multiple services, with short-term, dynamic configuration of network resources. Users with less elastic bandwidth and QoS requirements can maintain their QoS by paying more during congestion, while non-adaptive users can use services with static pricing which will be less efficient and hence more expensive.

Congestion pricing schemes in the network literature fall into two basic categories: tâtonnement and bandwidth auction. In tâtonnement, the price is updated iteratively until the aggregate user demand meets bandwidth supply. Tâtonnement-based pricing algorithms have been explored in a number of papers [4][5][6][7], where the basic idea is to signal a user the marginal cost it imposes on other users (which is also the price in a market-based resource allocation scenario) as an incentive for adaptive applications to adapt their sending rates. Auctioning has been proposed by researchers in the literature [8][9][10] as another model that allocates scarce bandwidth efficiently, though there is generally no explicit signaling for price before users send their requests. Even though there is a lot of literature on both types of congestion-pricing models, as far as we know, there has been no work comparing these two schemes in the same environment.

The main goal of this paper is not to propose new congestion pricing algorithms, but to develop comparable pricing models based on the above two

approaches and compare their performance. We study the pricing models in the environment where the network adjusts pricing periodically in the time unit of minute, and users make short-term reservations, and may adapt their demands in response to congestion-sensitive pricing. Since the time period between price adjustments is relatively long, the network transmission delay has negligible impact on the system performance.

The users and the network communicate through a Resource Negotiation and Pricing protocol (RNAP) [1], described in earlier work [11]. Under a tâtonnement-based model, network agents at each node periodically compute a local price for each service class based on resource availability, and an end-to-end message accumulates a total price and communicates it to the user. In response, the user may modify its resource reservation by an end-to-end message. Under an auction-based model, a user sends a set of bids in an end-to-end message, each bid indicating its willingness to pay for a certain bandwidth. Network agents carry out periodic auction to allocate bandwidth at a node, based on bids received from users. An end-to-end message communicates the smallest allocated bandwidth of all the nodes along the path to the user. There is no requirement for the synchronization of price updates or auction process in the whole network. An end-to-end price along a path is calculated based on the most recently updated price at a node.

We address some important practical issues related to making the tâtonnement- and auction-based schemes work in such an environment. We compare the schemes with respect to network utilization, connection blocking rate, user satisfaction and network revenue, and draw some general conclusions about the relative benefits of the two approaches.

The rest of this paper is structured as follows. Related work is discussed in Section 2, mainly pertaining to various auction models and their implementation. In Section 3, we describe a pricing model, which incorporates congestion-sensitive pricing. In section 4, we describe two alternative resource allocation strategies, corresponding to the two congestion-pricing strategies. The simulation set-up is described in Section 5, followed by a discussion of the simulation results in Section 6. Finally, we discuss and summarize the important features of our work in Section 7.

[1] Apart from the end-to-end messaging briefly described here, RNAP is intended to generally support resource negotiation in a network with multiple network services, and usage and congestion-dependent pricing. In particular, it supports the administration of pricing by local network agents at each node, as also by a centralized entity over an entire domain. To reduce processing and storage overhead and hence allow for better scalability, RNAP also supports the aggregation of messages and states.

2 Related Work

Theoretical frameworks of congestion pricing have been discussed thoroughly by several authors [4][5][6][7]. Kelly et al [4] and Low et al [7] show how selfish users, seeking to maximize their own net benefit, can be given the right incentives so as to globally optimize the social benefit. ECN-based marking has been proposed in [5][6] to convey congestion information back to the end systems, and the resulting system converges to a system optimal state as long as all utility curve are strictly concave. These schemes assume network services are best-effort, and rely on a pure market mechanism to maximize social benefit.

In [12][13][14][15][16], the resources are priced to reflect demand and supply. The methods in [13][14][16] 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 [15] is more similar to our work in that it takes into account network dynamics (session join or leave) and source traffic characteristics. It also allows different equilibrium prices over different time periods. However, congestion is only considered during admission control, and the study is restricted to a single service class.

Several auction-based mechanisms have been studied to elicit truthful user utility functions and encourage the efficient utilization of scarce network resources. In the "smart market" model [8], each packet header contains a bid field, and the packet is admitted if the bid exceeds the current cutoff amount, determined by the marginal congestion costs. The user pays the cutoff amount, instead of her own bid. The optimal strategy for the user is to bid her true valuation. The mechanism only provides a priority relative to other users, and is not an absolute promise of service. Issues that need to be addressed include accounting complexity, service interruptions during traffic peaks, and user response to fluctuations in price.

The model in [9] supports multiple levels of QoS guarantees. The implementation scheme is again called "smart market", also called "generalized Vickrey auction (GVA)". The central idea in the well-known Vickrey auction [17] is to award the item on auction to the highest bidder, but charge the second highest bid as the price. Bidding one's true valuation is a dominant strategy for each agent, and the resulting allocation is Pareto-efficient [17]. GVA extends the idea to allow agents to have preferences over more than one item, and more than one unit of the item. The "second-price" analogue is to charge each agent the total social surplus that would be possible if that agent did not participate in the auction. The optimal solution requires substantial computation, which increases polynomially with the number of users, and the number of optimizations increases linearly with the number of users. The "second-price" model is

also used in [10], and the proposed auction scheme is called progressive second price auction (PSP). PSP extends the traditional single non-divisible object auction to the allocations of arbitrary shares of the total available resource.

Inspired by [8], the work of Shu and Varaiya [18] generalized the idea in Vickrey auction by supporting auctions for different sevice levels. In DiffServ environment, the in-profile traffic is charged flat fee, while out-profile traffic is charged congestion price based on an auction-based admission control algorithm. The pricing algorithm for in-profile traffic is left undefined. The out-profile traffic will be treated differently based on the users' bids.

A number of practical issues are not addressed in the above auction models. Other than computation and sorting complexity, potential problems include signaling bursts, set-up delay, and uncertainty of connection availability. Auctions taking place at intervals may cause signaling bursts around each auction moment. Set-up delays arise because new users have to wait until the next auction to receive requested resources. Finally, for the purpose of congestion control, the end-to-end connection may need to be refreshed from time to time based on new auctions. At each refresh, a user risks losing its connection.

To address some of these concerns, Delta Auction (DA) [19] was proposed to allow auctions to take place continuously. Arriving bids that are too low are refused right away. Sufficient bids are accepted provisionally, but there is the possibility that bids arriving later may exceed those that are admitted and thus oust them from the auction. The advantage of this scheme is that the signaling traffic is distributed uniformly over time, and a user is informed quickly about refused reservations.

A Connection-Holder-is-Preferred-Scheme (CHiPS) [20] was proposed to resolve the uncertainty of the connection. Current connection-holders are preferred by allowing them to submit a second bid if their first bid is rejected. This still does not eliminate the uncertainty completely. It also appears to be inconsistent with the objective of the second-price model, which is to elicit the true user valuation of a service, since the only choices for a user is either to withdraw, or to submit a second-chance bid with a price different from the user's willingness to pay. If some users withdraw, other users may end up paying an unfairly higher price, compared to the price they would have paid if the withdrawals were taken into account at the outset.

3 Pricing Strategies

User experiments indicate that usage-based pricing is perceived as a fair way to charge people and allocate network resources [21]. Simple usage-based charg-

ing schemes charge the users a fixed price per unit bandwidth. Without any incentive, customers have no motivation to reduce their traffic as network congestion increases. 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.

Different algorithms can be used for computation of a local or incremental price (including the congestion-dependent component) for a service at a given point in a network. In earlier work [3], we proposed a pricing structure in a DiffServ environment. In addition to a congestion-dependent component, the service price has two time-invariant components, a *holding price* and a *usage price*, based on the cost of providing different levels of services, and on long-term demand. In this section, we summarize the main features of this pricing structure, and then describe in detail how tâtonnement and auction can compute the *congestion* price component.

3.1 Usage Charge

The usage charge is determined by the long-term user demand, the level of service guaranteed to the user, and the elasticity of the traffic. The model we consider is a network supporting J classes of service. The usage price for class j is p_u^j, and the long-term user bandwidth demand of class j can be estimated based on statistics and represented as x^j. The provider's decision problem is to choose the optimal prices for each class that optimize its total profit:

$$\max_{p_u^j}[\sum_{j}^{J} x^j p_u^j - f(C)],$$
$$\text{subject to:} \quad \sum_{j}^{J} x^j \leq C, \tag{1}$$

where C is the bandwidth availability of the network, and $f(C)$ is the network bandwidth cost during one unit of time.

In section 4, we discuss the representation of user preferences through an *user utility function*. Based on the discussion, we assume a general form for the utility function (Equation 10). Given this utility function, the user demand for a service class j is a constant elasticity function: $x^j(p_u^j) = A^j/p_u^j$, which varies inversely with the price of the class. A^j reflects the total willingness to pay of users for service class j, as estimated by the provider. Note that even though the objective function of equation 1 no longer depends on usage price if a constant demand function is assumed, the impact of usage price is reflected through the constraint function. The service selection of a user will be

affected by the prices and service quality of different service classes. However, once a user selects a service level, we assume the amount of resources the user requires only depends on the price of the selected class. The usage price and resource provisioning of a class will be adjusted over longer time scale, depending on the long-term total average user demand of different classes. Denoting the equilibrium unit bandwidth price at a node under full utilization by p_{basic}, and the expected utilization of service class j by ρ^j, the usage price for differentiated service classes was developed in [3] and is given by:

$$p_u^j = \frac{p_{basic}}{\rho^j}, \quad \text{where} \quad p_{basic} = \frac{\sum_j^J A^j}{C}. \tag{2}$$

Therefore,

$$p_u^j = \frac{\sum_j^J A^j}{C\rho^j}. \tag{3}$$

Using this model, the bandwidth provisioned for a service class j will be given by A^j/p_{basic}. The usage charge $c_u^{ij}(n)$ of customer i for class j over a period n in which $v^{ij}(n)$ bytes are transmitted is given by:

$$c_u^{ij}(n) = p_u^j v^{ij}(n) \tag{4}$$

3.2 Holding Charge

If admission control is enforced, the applications admitted into the network will impose an opportunity cost by depriving other applications of the opportunity to be admitted, even if the resources are not actually being used. If a particular flow or flow-aggregate does not utilize completely the resources set aside for it, the scheduler generally allows the resources to be used by excess traffic from a lower level of service. The holding charge reflects the cost imposed by users not utilizing resources set aside for them. It is determined based on the revenue lost by the provider because instead of selling the allotted resources at the usage price of the given service level (if all of the reserved resources were consumed) it sells the unused part of the resources at the usage price of a lower service level. The holding price (p_h^j) of a service class j is therefore set to reflect the difference between the usage price for that class and the usage price for the next lower service class and can be represented as

$$p_h^j = p_u^j - p_u^{j-1}. \tag{5}$$

The holding charge $c_h^{ij}(n)$ when a customer i reserves bandwidth $r^{ij}(n)$ from class j during time period n is given by

$$c_h^{ij}(n) = p_h^j(r^{ij}(n)\tau^j - v^{ij}(n)), \tag{6}$$

where τ^j is the length of negotiation interval for class j, $v^{ij}(n)$ is the traffic sent by user i over the period n, and $r^{ij}(n)\tau^j - v^{ij}(n)$ is the bandwidth not used by the user. $r^{ij}(n)$ can be a bandwidth requirement specified explicitly by the customer i, or estimated from the traffic specification and service request of the customer.

3.3 Congestion Charge

The general network resources considered to compute the congestion charge are bandwidth and buffer space. The congestion price is levied when demand exceeds a provider-set fraction of the available bandwidth or buffer space. The congestion price is re-computed periodically at some 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. In this paper, we only consider bandwidth scarcity in calculating the congestion price. The proposed pricing schemes can be applied similarly to buffer space scarcity.

With congestion price for class j over a period n represented as $p_c^j(n)$ and the volume transmitted as $v^{ij}(n)$, the total congestion charge for customer i is given by

$$c_c^{ij}(n) = p_c^j(n)v^{ij}(n). \tag{7}$$

The impact of congestion charge on user adaptation and network performance depends on how the congestion price $p_c(n)$ for a period n is determined and the associated resource allocation strategy. In this section, we describe how the congestion price is determined by two methods, tâtonnement and auction. For convenience of presentation, we refer to congestion-price-based adaptation using tâtonnement as CPA-TAT, and refer to congestion-price-based adaptation using auction as CPA-AUC.

3.3.1 Congestion Pricing based on Tâtonnement Process

Tâtonnement implements the welfare theory in a competitive market [17]. The price change, upward when aggregate user demand exceeds resource sup-

ply and downward when demand is lower than the supply, drives the demand and supply towards equilibrium. Congestion pricing through an iterative tâtonnement process can be represented as

$$p_c^j(n) = \min[\{p_c^j(n-1) + \sigma^j(x^j - \rho_*^j)/\rho_*^j, 0\}^+, p_{max}^j], \tag{8}$$

where x^j and ρ_*^j represent the current total offered network load and target bandwidth utilization for service class j respectively, σ^j is a factor used to adjust the convergence speed [2], and p_{max}^j is the highest congestion price that can be applied. Equation 8 follows the integral control law [22] to drive the user demand towards the target bandwidth utilization. The router begins to apply the congestion charge only when the offered network load exceeds a certain target bandwidth utilization. After the congestion is removed, the congestion charge is gradually reduced to zero to protect against network traffic oscillation. In our simulations, we also use a price adjustment threshold parameter θ^j to limit the frequency with which the price is updated. The congestion price is updated only if the calculated price increment exceeds $\theta^j p_c^j(n-1)$.

For predictable service for long-lived applications that are intolerant of disruption (such as VoIP or media streaming), we need some form of session-level admission control. The maximum congestion price p_{max}^j sets the admission control threshold - that is, all new arrivals of a class j are rejected when the congestion price reaches p_{max}^j. If p_c^j reaches p_{max}^j frequently, it indicates that more resources are needed for the corresponding class, or that the long-term usage price for the class needs to be increased to reflect the current demand.

3.3.2 Congestion Pricing based on Auction

As discussed in Section 2, the auction schemes proposed in the literature tend to be theoretical, and leave practical issues open. Also, most of them do not consider short-term reservation. The auction price either varies packet-by-packet [8], or remains constant throughout a flow's life time [9][10]. Only the DA and CHiPS models [19][20] address short-term resource reservation and periodic resource auction. However, neither model addresses users with elastic service requirements and their responses to price fluctuations. In practice, the user-perceived value per unit bandwidth generally decreases as the total

[2] For an integral controller, higher control gain σ leads to a faster response of the congestion price p_c. However, large values of σ can cause excessive oscillation or instabilities. Also, if minimum and maximum limits are set on the congestion price (say, zero and $p_{\max}$ respectively), setting σ too high can force p_c into one of the limit states. Assume ϵ is the largest error that occurs in closed-loop operation; to avoid forcing p_c into a limit state, σ should be set no higher than $\frac{p_{\max}}{\epsilon}$.

bandwidth increases. Consequently, when an adaptive application learns of an increase in service price under CPA-TAT, it tends to reduce its bandwidth request in order to maximize its perceived value. However, in a periodic auction scheme in which a user submits a single bid in each period, the user may not be able to learn about increased competition for scarce bandwidth until its bid is rejected. The CHiPS auction scheme [20] gives current connection holders a second chance. But this does not eliminate completely the uncertainty of service availability, and has other problems discussed in Section 2.

A way to resolve this problem is to allow a user to send multiple bids at a time indicating its willingness to pay a premium for different amounts of bandwidth during congestion. The concept of submitting multiple bids simultaneously has been used in auctions in other fields, for example, the annual England-France power interconnection auction. This approach is well-suited to a short-term reservation / user adaptation framework, and allows comparison of auction-based resource allocation with the tâtonnement-based approach. We refer to this auction scheme as ***M**-bid auction*. Each user sends an M-bid, which consists of multiple (price, bandwidth) pairs. The bid price represents the per-unit *premium* a user is willing to pay above the long-term fixed price (usage price plus holding price) during congestion to receive the corresponding bandwidth.

Users who have elastic bandwidth requirements but highly value an uninterrupted connection will ensure a high probability of receiving at least the minimum required bandwidth during congestion by bidding a high price (per unit bandwidth) on their minimum bandwidth requests, and a relatively low price on their higher-bandwidth requests. Users with higher budgets and less elastic requirements will bid a relatively high price for all bandwidths.

We assume that the elasticity of an user's preference can be expressed by an user utility function, discussed further in Section 4. The M-bids are derived by sampling the utility function. If a user's bids reflect her truthful valuations of different network bandwidth, the bids should not depend on the network conditions or other users' bids.

Each network entity periodically performs an auction to redistribute the bandwidth based on the user bids and network conditions. There is no need to synchronize the requests of all the users. All the requests arrived during the same auction period will participate in the auction at the end of the auction period. If a user does not update its bids for a new auction period, its old bids will be used. During the auction, bids from all the users are ranked based on the bidding prices. Bandwidth is allocated starting with the highest bid, until the target utilization is met. The congestion price is set to the highest rejected bid price, in accordance with the second-price auction concept [17]. If more than one bid price from a user is higher than the cutoff price, only the

one with the lowest bid price (potentially the highest bandwidth) is ultimately selected. When congestion is detected, the auction results are used to first reduce the bandwidth allocation of users with elastic bandwidth requirements, while maintaining the bandwidth allocation of users with more willingness to pay. Thus, the network tries to maximize overall user satisfaction.

Bid Price	Bid Bandwidth	Bidder	Bid Selection
5	10	1	
4	10	2	
4	15	1	←
3.5	20	3	←
3	25	2	←
2	30	3	×

Table 1
Example M-bid auction. Selected bids are marked as '←', rejected bids are marked as '×'

Table 1 shows an example of an M-bid auction. There are three users, each submitting two bids to a network entity. The total available network bandwidth is 70. All the user bids are ranked based on their bidding price. The example shows that all the users are accepted, and the total allocated bandwidth is 60. The lowest bids from users 1 and 2 are selected. The lower bid of user 3 is rejected. The per-unit bandwidth congestion-price is set as 2, the highest rejected bid price. Note that when some bandwidth is left unallocated (e.g., in table 1, the total bandwidth allocated is 60, which leads to 10 unit of bandwidth unallocated), the proposed M-bid auction strategy does not intend to re-distribute the remaining bandwidth to all the auction winners through sophisticated scheme. There is no need to allocate all the bandwidth at one time, with the dynamic change of users' requests and frequent session arrivals and departures. Also, the final bandwidth allocated to a session depends not only on the allocation at one node, but the allocation along the whole transmission path. It will be more complicated if resource re-distribution also needs to take into account the interaction of the resource allocation among multiple nodes. As we will discuss at the end of this section, the remaining bandwidth will not be wasted, but help reduce the setup delay of a new session.

For implementation, it is convenient to organize all the bids in a binary tree. New bids are inserted into the tree upon a session's arrival and all the bids from a session will be removed from the tree upon its departure. If the total number of bids in the tree is N, and a new user submits M bids, the insertion and deletion complexities are $M \log N$. The complexity of calculating the total bandwidth is $O(N)$.

The M-bid auction has a number of practical advantages in the network environment. By allowing users to submit multiple bids at the start of a session, signaling is reduced significantly, and there are no signaling bursts during auctions. By submitting all the preferences in advance, a user minimizes her uncertainty of connection availability, and reduces setup delays in subsequent refresh auctions. To reduce the setup delay of a new session, an *intermediate admission* mechanism can be used. Instead of waiting until the next auction, a network entity will allocate bandwidth to the user immediately if the resource is available, and if at least one of the user bids exceeds the service price from the preceding auction. Bandwidth may be available for immediate allocation for various reasons: due to the demand during the previous auction being less than the capacity, due to sessions having terminated since the last auction, or due to users reserving less bandwidth than they were allocated by auction (generally due to being allocated less bandwidth at other nodes along their data paths).

A disadvantage of auction-based schemes in general, including the M-bid scheme, is that when a user is making an end-to-end resource reservation, it must determine how to split its budget in bidding for resources at each hop. In this work, we assume that user's willingness to pay for each level of bandwidth is split evenly among all the hops. We will attempt a more realistic solution in later work (in practice, if the user obtains pricing or congestion information from the network, it will tend to allocate more of its budget among bottlenecked nodes).

3.4 Total Charge

Based on the price formulation strategy described, the total charge for a session i with a service class j is given by

$$c_s^{ij} = \sum_{n=1}^{N} [p_h^j (r^{ij}(n)\tau^j - v^{ij}(n)) + (p_u^j + p_c^j(n))v^{ij}(n)], \tag{9}$$

where N is the total number of intervals spanned by the session i.

Networks may set the usage charge to zero, imposing a holding charge for reserving resources only, or apply a congestion charge during resource contention. Also, the holding charge would be set to zero for services without explicit resource reservation or admission control, for example, best effort service.

4 Resource Allocation

In a network with congestion-dependent pricing and dynamic resource negotiation, *adaptive* applications with a budget constraint will tend to adjust their service requests in response to price variations. We assume that the preferences or willingness to pay of a user will be represented quantitatively through a *utility function*. The utility function represents the perceived monetary value (say, 15 cents/minute) provided by a set of transmission parameters (e.g., sending rate and QoS parameters).

Depending on the congestion pricing mechanism, the user adaptation happens in one of two ways. Under the CPA-TAT policy, an intelligent user agent uses the utility function, user budget, and network price information to determine the optimal service request and data rate. Under the CPA-AUC policy, the user agent samples the utility function to obtain a set of bids, and the network entity sets the price and allocates bandwidth through periodic M-bid auctions. The user agent sets the data rate according to the allocated bandwidth.

In practice, the application utility is likely to be learnt and indicated by users 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 [23] and controlled-load service [24] under the IntServ model, and Expedited Forwarding (EF) [25] and Assured Forwarding (AF) [26] under DiffServ. In this case, it is convenient to represent the utility as a piecewise linear function of bandwidth (or as a set of such functions, one for each level of loss and delay). A simplified algorithm is proposed in [11] to search for the optimal service requests in such a framework.

At a fixed value of loss and delay, we can make some general assumptions about the utility function as a function of the bandwidth. A user application generally has a minimum requirement for the transmission bandwidth. It 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 it can not obtain the minimum bandwidth, or when the price charged is higher than the opportunity value derived from keeping the connection alive. User experiments reported in the literature [27][28] 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. Also, as shown in [4], the optimal solution is proportionally fair [3]

[3] A vector of rates $\hat{x} = (x^k)$ is proportionally fair if it is feasible, and if for any other feasible vector $\hat{x_*}$, the aggregate of proportional changes is zero or negative: $\sum_k \frac{x_*^k - x^k}{x^k} \leq 0$.

when all user utilities are logarithmic. Based on the above considerations, we assume the following general utility function in our simulations:

$$U(x) = U_0 + w \log \frac{x}{x_{\min}}, \tag{10}$$

where $U(x)$ denotes the utility at a particular bandwidth x, $x_{\min}$ 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 at the lowest bandwidth level ($x_{\min}$). We assume a logarithmic form for the utility function in this paper. We stress that this is for convenience of analysis; similar results are obtained with other concave forms [29].

We now consider the resource allocation scenarios under CPA-TAT and CPA-AUC.

4.1 User Adaptation under CPA-TAT Policy

Consumers in the real world generally try to obtain the best possible "value" for the money they pay, subject to their budget and minimum and maximum quality requirements. In our case, the "value for money" obtained by the user corresponds to the "surplus" between the utility $U(\cdot)$ and the cost of obtaining that service. In an environment with multiple services offering different delay and loss expectations, the user utility is a function of QoS metrics, such as loss and delay, as well as bandwidth. The optimization of surplus can be written as:

$$\begin{aligned} &\max(U(x,q) - C_o(x,q)) \\ \text{s. t. } \; &C_o(x,q) \le b, \quad x_{\min} \le x \le x_{\max}, \quad q_{\min} \le q \le q_{\max}, \end{aligned} \tag{11}$$

where x and q are, respectively, the bandwidth and quality of service parameters, $x_{\min}$, $x_{\max}$ and $q_{\min}$, $q_{\max}$ represent the minimum and maximum bandwidth requirements and quality of service requirements respectively, b is the user budget, and C_o is the cost of obtaining service with the bandwidth and QoS parameters x and q.

In our simulations, we compare the pricing models in a simpler environment, with a single available service and a utility function dependent only on bandwidth. We assume a fixed per-unit bandwidth cost p, so that C_o increases linearly with bandwidth. When the utility is in the form of equation 10 as a function of bandwidth and at a fixed loss and delay, the optimization process is:

$$\max(U_0 + w \log \frac{x}{x_{\min}} - px), \qquad (12)$$
$$\text{s. t. } px \leq b, \quad x_{\min} \leq x \leq x_{\max}.$$

The optimal bandwidth of a user can be obtained by solving the Kuhn-Tucker equations [17]. If the user can obtain the optimal bandwidth for the system at a cost below his budget, then the user demand can be shown to be $x = w/p$, and w represents the money a user would spend based on its perceived value for the application. Otherwise, the demand is budget-constrained, $x = b/p$. A more detailed study of user adaptation under CPA-TAT, including optimization of the user surplus with respect to both bandwidth and QoS parameters, is given in [29].

4.2 Resource Allocation under CPA-AUC Policy

The user M-bid is derived by sampling the utility function at M points, at equal utility intervals, and subtracting the corresponding time-invariant charge from the sampled utility values to derive the premium the user is willing to pay for that bandwidth during congestion. Since the marginal utility as a function of bandwidth diminishes with increasing bandwidth, the bid prices are more closely spaced initially, when the utility changes more rapidly as a function of price, and are more widely spaced at higher prices, when the utility is relatively static. Using the utility function of equation 10, the utility values at the sample points can be written as:

$$\begin{aligned} U_k &= U(x_{\min}) + \frac{(U(x_{\max}) - U(x_{\min}))k}{M-1} \\ &= U_0 + \frac{(w \log \frac{x_{\max}}{x_{\min}})k}{M-1}, \end{aligned} \qquad (13)$$

where $k = 0, ..., M-1$. The bid bandwidth and price x_k and p_k at a sample point k can be written as:

$$\begin{aligned} x_k &= x_{\min} 10^{\frac{U_k - U_0}{w}}, \quad k = 0, ..., M-1 \\ p_k &= \frac{U_k}{x_k} - p_u, \end{aligned} \qquad (14)$$

where p_u is the usage price of the bandwidth. The selected bidders are assumed to fully use up the bid bandwidth, and hence the holding charge is not considered in calculating bidding price. The network periodically updates the end-to-end price and bandwidth allocation for each user based on new auction results. A user application adjusts its sending rate correspondingly.

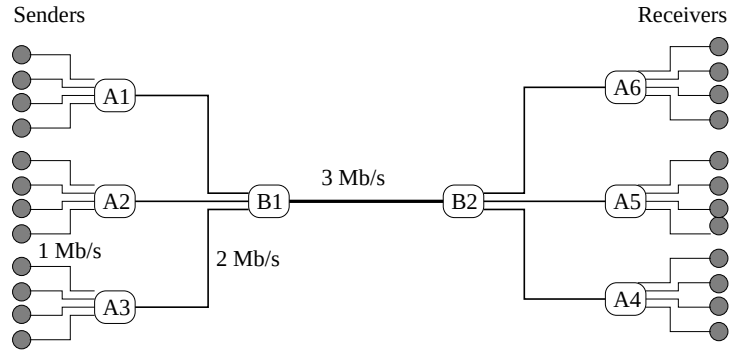

Fig. 1. Simulation network topology 1

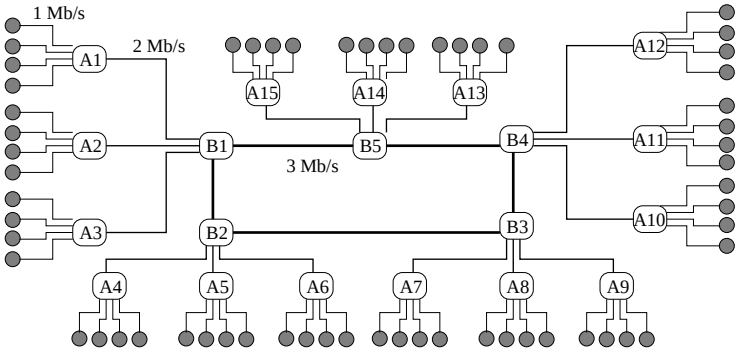

Fig. 2. Simulation network topology 2

5 Simulation Model

In this paper, we compare two congestion-based pricing schemes, CPA-AUC and CPA-TAT. For reference, we also simulate a congestion-independent policy, with a fixed price per unit bandwidth (or usage price). We refer to this fixed-price policy as "FP".

We used the *network simulator* [30] environment to simulate two different network topologies, shown in Fig. 1 and Fig. 2. Topology 1 consists of two backbone nodes, six access nodes, and 24 end nodes. Topology 2 is a more general network topology described in [31]. This topology contains five backbone nodes, 15 access nodes, and 60 end nodes. All links are full duplex and point-to-point. The links connecting the backbone nodes are 3 Mb/s, the links connecting the access nodes to the backbone nodes are 2 Mb/s, and the links connecting the end nodes to the access nodes are 1 Mb/s. At each end node, there is a fixed number N_s of sending users. In topology 1, users from the sender side independently initialized unidirectional flows towards randomly selected receiver side end nodes. At most $12N_s$ flows (48 sessions with N_s set to 4) could run simultaneously. In topology 2, all the users initialized unidirectional flows towards randomly selected end nodes. At most $60N_s$ users (360 sessions with N_s set to 6) were allowed to run simultaneously.

Experiments to study specific effects were mainly performed on network topology 1. One reason for this was simply to make simulations more tractable and convenient. Also, effects were easier to see with congestion at a single bottleneck. Experiments on topology 2 confirmed the same qualitative effects; one set of results under the parameters given in this section is shown in Section 6.5.

All user traffic was assumed to belong to one service class; interaction between service classes is outside the focus of this work, and was addressed in [3]. The policies were simulated at the call-level, based on the user-requested bandwidth, as opposed to packet-level. User requests were generated according to a Poisson arrival process and the lifetime of each flow was exponentially distributed with an average length of 10 minutes, representative of a typical telephone call [32]. The users were assumed to have the general form of the utility function of Section 4. w, the elasticity factor, (and also the user's willingness to pay) was uniformly distributed between \$0.125/min and \$0.375/min, roughly representing international phone rates. The opportunity cost parameter U_0 was set to a user's willingness to pay for its minimum bandwidth requirement.

The unit bandwidth usage price, p_u, was set to \$0.15/min for 64 kb/s transmission under all three policies. The holding price p_h was assumed to be zero, since all simulations are currently performed within a single service class. The price updating interval at each LRN (through tâtonnement or auction) and the refresh interval for Quotation messages to collect the end-to-end price were both set to 30 seconds.

For the CPA-TAT policy, the default parameter values were 0.9 for the targeted link utilization ρ, and 0.06 and 0.05 for the price adjustment parameters σ and θ. The CPA-AUC policy was simulated with the "immediate admission" feature described in Section 3 - that is, a M-bid request was admitted immediately when possible, instead of waiting for the next scheduled auction. In the CPA-AUC policy, M, the number of price-bandwidth pairs per bid, was set to 5.

6 Simulation Results

In this section, we show simulation results from a set of experiments under the conditions described in Section 5. We generally look at a number of engineering and economic metrics, as a function of *unconstrained user demand*. The unconstrained user demand is defined as what the total user demand at the bottleneck would be if there was no admission control and no user adaptation in response to congestion pricing, normalized with respect to the bottleneck capacity. In the case of CPA-AUC, the unconstrained demand corresponds to

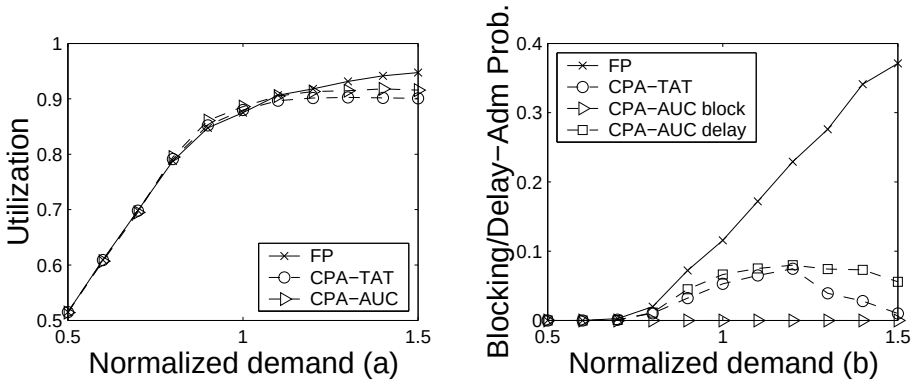

Fig. 3. Performance metrics of CPA-TAT, CPA-AUC, and FP policies as a function of normalized unconstrained demand: (a) bottleneck utilization; (b) blocking probability.

the sum of the highest bandwidth bids of all the users at a given time. The engineering metrics we look at include the average traffic arrival rate at the bottleneck, and the user request blocking probability. The economic performance metrics include the average and total user benefit (the perceived value obtained by users based on their utility functions), the end-to-end bandwidth price, and the network revenue a network provider can earn from all the admitted requests.

6.1 General Comparison of CPA-TAT, CPA-AUC and FP

We first compare the performance under the FP policy and the two CPA policies, with the default conditions specified in Section 5.

6.1.1 Bottleneck Bandwidth Utilization and Request Blocking Probability

In Fig. 3(a), CPA-TAT coupled with user adaptation is seen to maintain the network load at the targeted level ($\rho_* = 0.9$ in this simulation) as the normalized unconstrained user demands vary from 0.5 to 1.5. CPA-TAT has to be conservative because it relies on an iterative process (price adjustment leads to user adaptation) to prevent the total user demand from exceeding the available resources. Since the CPA-AUC policy has a-priori knowledge of user bids, it can allocate all the available link capacity and achieves higher utilization.

Fig. 3(b) demonstrates that, as expected, the blocking probability of the FP scheme increases sharply after the offered load exceeds 0.8. The blocking probability of CPA-TAT is up to 40 times less than that of FP and actually starts to decrease after reaching a maximum at offered load 1.2. This is because the price adjustment step is proportional to the excess demand above the tar-

geted utilization and the price increases progressively faster at higher loads. The figure shows two curves for CPA-AUC. The curve labeled "block" indicates the real blocking rate - that is, the average fraction of user requests that are rejected at auctions. Since the user utility functions were chosen such that the total minimum resource requirement is less than the network capacity, the blocking rate is almost zero in this case. The curve labeled "delay" indicates the fraction of user requests that cannot get "immediate" (auctionless) admission, and must wait until the next auction. The simulation shows that up to 9% of users are delayed. This simulation shows a fundamental difference between CPA-TAT and CPA-AUC: CPA-AUC can accommodate a certain set of adaptive users (although some of them may be delayed), while a certain fraction of the same set of the users will be rejected under CPA-TAT. Again, this is because CPA-AUC has a-priori knowledge of user demand and resource availability, while users under CPA-TAT learn about the availability indirectly (through the congestion price), and with some delay. In subsequent experiments, only the delayed admission probability of CPA-AUC is shown in the place of blocking rate. It should be noted that the real blocking rate in these cases is zero, or nearly zero.

6.1.2 Network Price and Revenue

Figs. 4(a) shows the average and standard deviations of the system price (over simulation time) as a function of the user demand. The standard deviations of both CPA-TAT and CPA-AUC show the same trend, an increase to a certain level and then a decrease. Initially, the price variations increase with the load due to the more aggressive congestion control. At heavy loads, the increased multiplexing of user demand smoothes the total demand, and therefore reduces fluctuations in the price. The average price is higher under CPA-TAT than under CPA-AUC during congestion, again reflecting the more conservative resource allocation under CPA-TAT, and correspondingly more aggressive congestion pricing at a given load. The average price of CPA-TAT reduces with the increase of target bandwidth utlization, as indicated in Section 6.2.1.

Fig. 4(b) shows that the revenue of FP flattens out after the onset of request-blocking. The revenue of CPA-AUC increases significantly under heavy load, due to the admission of higher user bids. The revenue of CPA-TAT is seen to increase faster than that of CPA-AUC, and faster than linearly after the network utilization saturates at the targeted level. CPA-TAT obtains more revenue than CPA-AUC at high loads due to its higher congestion price. The loss of revenue under both CPA policies 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.

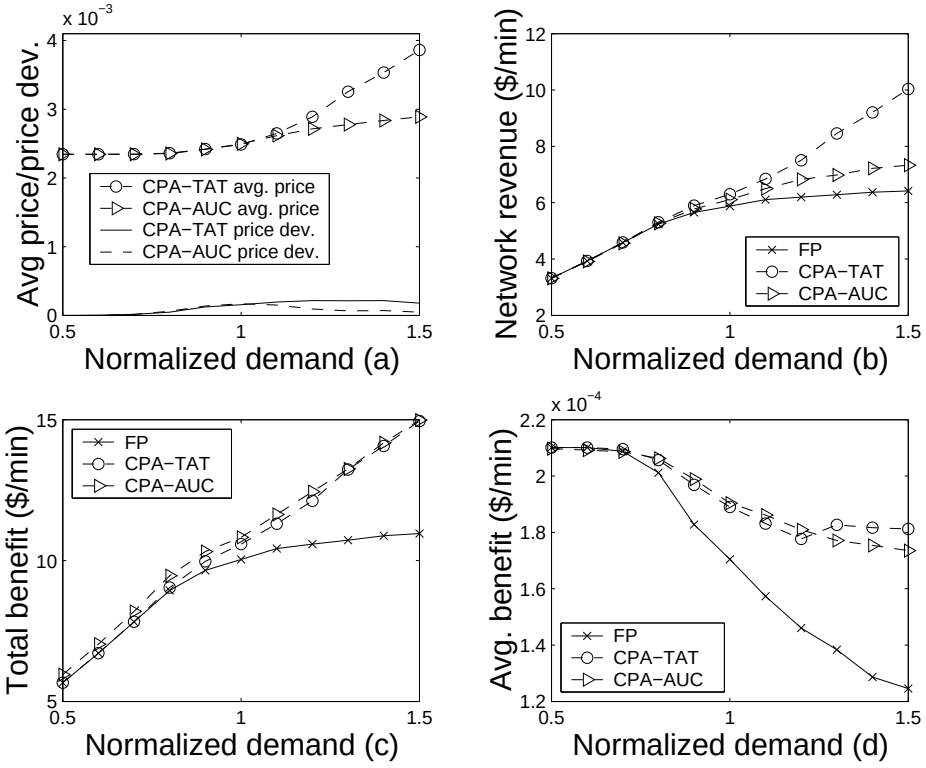

Fig. 4. Performance metrics of CPA-TAT, CPA-AUC, and FP policies as a function of normalized unconstrained demand: (a) time-average and standard deviation of system price of CPA-TAT and CPA-AUC; (b) total network revenue; (c) total user benefit; (d) average user benefit.

6.1.3 Average and Total User Benefit

Fig. 4(c) shows that the total user benefit gained under the two CPA polices are similar and increase with the load, while the user benefit of FP flattens out after the onset of request blocking. As illustrated in Section 4, 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". Therefore, the CPA policy allows the network bandwidth to be used more efficiently under high loads.

Fig. 4 (d) shows that the average user benefit under both CPA policies are similar and are much higher than under the FP policy. For the FP policy, the average benefit per *admitted* user is constant. However, a progressively smaller fraction of users is admitted due to blocking. Hence the average perceived benefit of FP across all users decreases sharply with the load. The average user benefit of CPA-TAT is seen to be higher than that of CPA-AUC at higher loads. However, this only remains true as long as the blocking rate under CPA-TAT is sufficiently low. As we will see in later simulations, the block-

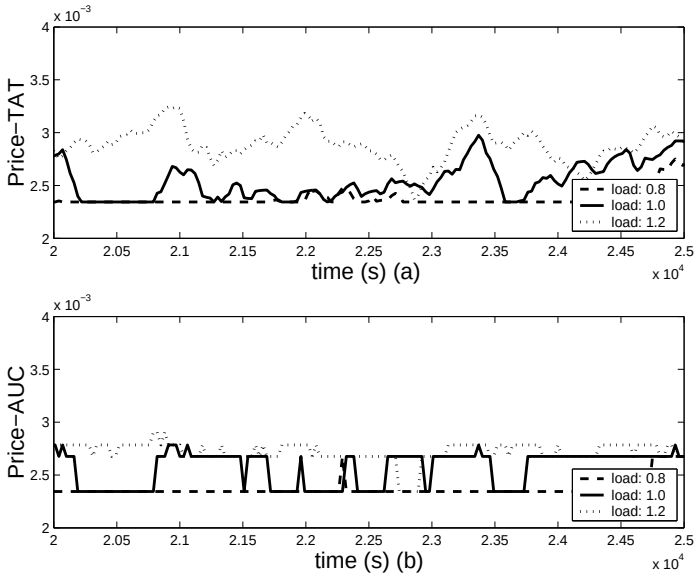

Fig. 5. Variation over time of system prices of CAP-TAT (a) and CPA-AUC (b).

ing rate of CPA-TAT increases significantly under some conditions, while the blocking rate under CPA-AUC remains close to zero. Under these conditions, the average user benefit of all users under CPA-TAT may be lower than that under CPA-AUC. The reason that CPA-TAT could gain higher average user benefit may be because the user could benefit more from multiplexing gain at higher load and obtain higher average bandwidth, while the user in CPA-AUC could not get the same amount of bandwidth allocation due to the discrete sampling of bandwdith for bidding. Also, the session setup delay reduces the average user benefit, and the delay impact becomes bigger as the network load increases and average user bandwidth reduces.

6.1.4 Dynamics of the System Price

Figs. 5 (a) and (b) show the variation of the system price of the two CPA policies at three different levels of user demand between sampled period 20000 seconds and 25000 seconds. The prices are nearly static at a load of 0.8, and are adjusted more frequently at higher load, due to the more frequent user arrivals and departures. The price variation of CPA-AUC is smaller than that of CPA-TAT, which reacts more actively with the load to drive the user demand towards supply bandwidth. Both policies have stable prices.

6.2 Variations of System Control Parameters

In this section, we study the impact of certain control parameters on the performance of CPA-TAT and CPA-AUC. For CPA-TAT, we varied the con-

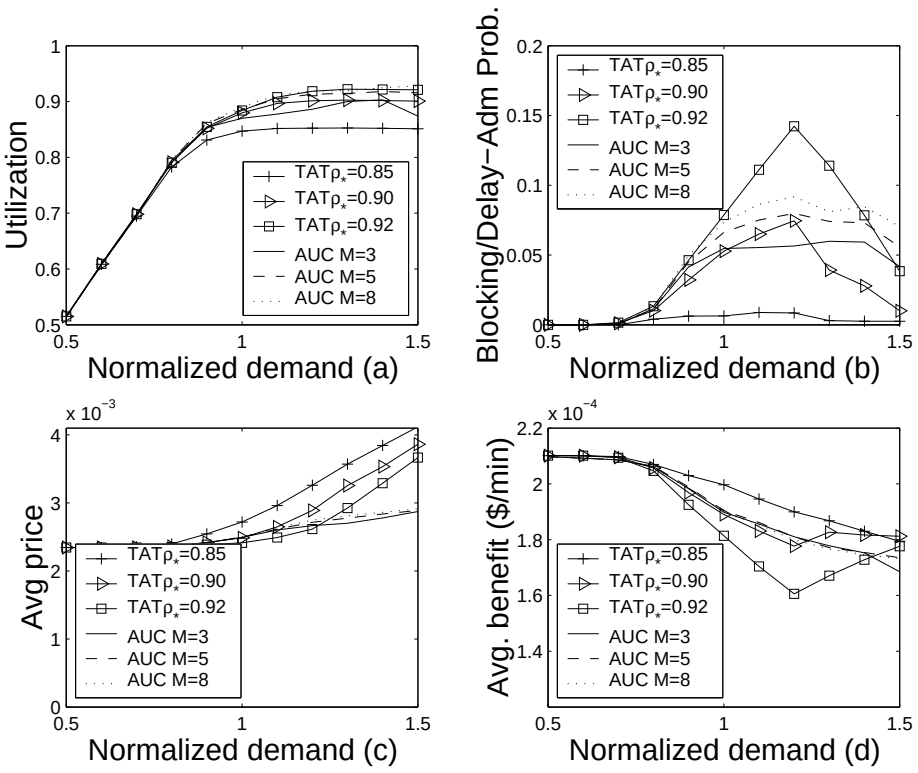

Fig. 6. Performance of CPA-TAT and CPA-AUC at different values of target bandwidth utilization for CPA-TAT, and different values of M, the number of bids in CPA-AUC: (a) bottleneck utilization; (b) probability of request blocking for CPA-TAT and delayed admission for CPA-AUC; (c) average system price; (d) average user benefit.

gestion control threshold (or targeted link utilization) ρ_*, beyond which the congestion-dependent price component was imposed. For CPA-AUC, we varied M, the number of price-bandwidth pairs per bid. We also study the impact of negotiation interval on both CPA policies. The results are shown in Fig. 6 and Fig. 7, and discussed below.

6.2.1 Effect of Congestion Control Threshold and M

Fig. 6 shows that the performance of CPA-AUC improves only slightly with increase in M, and the curves corresponding to the three values of M are almost co-incident in Fig. 6 (c) and (d). The robustness to the variation of M indicates that the bid selection scheme proposed in Section 4.2 is effective in capturing users' preferences. Fig. 6 also shows that CPA-TAT is able to maintain the utilization at the target value ρ_* in all 3 cases - $\rho_* = 0.85$, 0.90, and 0.92. At the highest utilization, the throughput of CPA-TAT approaches that of CPA-AUC. However, unlike CPA-AUC, a high throughput in CPA-TAT is obtained at the cost of performance: the blocking rate and correspondingly the

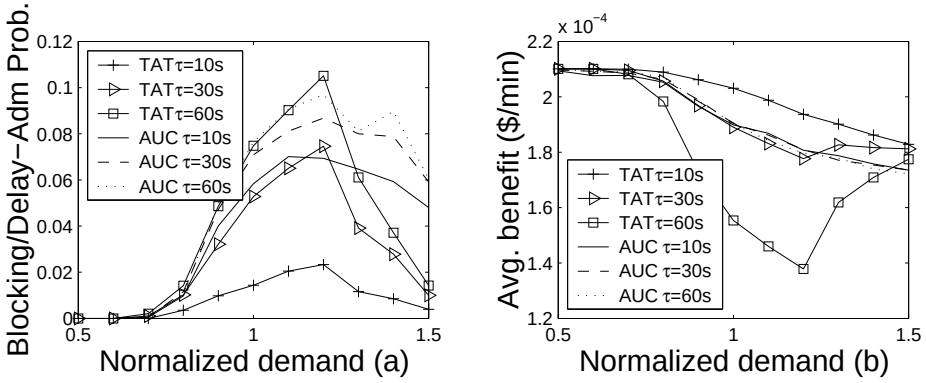

Fig. 7. Performance comparison between CPA-TAT and CPA-AUC for negotiation interval 10 s, 30 s, and 60 s: (a) probability of request blocking for CPA-TAT and delayed admission for CPA-AUC; (b) average user benefit.

average user-perceived benefit degrade significantly as the target utilization is increased (Fig. 6(b) and (d) respectively). Fig. 6(c) shows that the system price of CPA-TAT decreases as the target utilization is increased, indicating that congestion control becomes less aggressive at a given load.

6.2.2 Effect of Resource Negotiation Interval

The resource negotiation interval is the interval at which the network entities update the congestion price in CPA-TAT, and conduct auctions in CPA-AUC. Fig. 7 shows that, as expected, the performance of both policies improves as the negotiation interval is reduced. Essentially, this is because both policies are able to track changes in demand more actively when the resource negotiation interval is short. In Fig. 7 (b), we note that the average user benefit under CPA-TAT at a negotiation interval of 60 seconds decreases sharply with load initially, becoming lower than the average benefit of CPA-AUC, and then improves at high loads. This can be attributed to the high blocking rate of CPA-TAT at 60 seconds, and the sharp "knee" in the blocking rate (discussed in sub-section 6.1). As discussed in sub-section 6.1, a high blocking rate under CPA-TAT causes a corresponding decrease in the average user benefit, but changes in the delayed admission probability under CPA-AUC have little effect on the average user benefit. Since the blocking probability of CPA-AUC is almost zero for all the negotiation intervals, the admission delay of a auction request at the beginning of a session does not significantly affect the user's overall benefit.

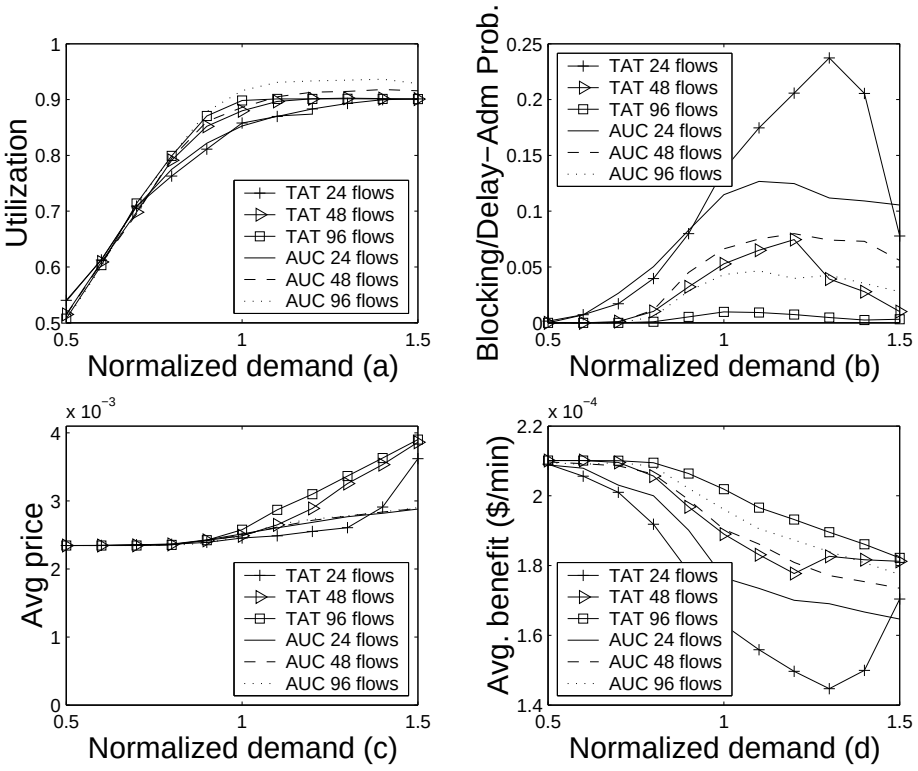

Fig. 8. Performance comparison between CPA-TAT and CPA-AUC with different numbers of user flows: (a) bottleneck utilization; (b) request blocking probability; (c) average system price; (d) average user benefit.

6.3 Effect of Session Multiplexing

To evaluate the effect of the increased multiplexing of session requests under CPA-TAT and CPA-AUC policies, we varied the number of customers sharing a system. We kept the network topology and user utility distributions unchanged, but scaled the link capacity proportionally with the maximum number of flows.

Fig. 8 (a) shows that as traffic multiplexing increases, the throughput increases. As expected, Fig. 8 (c) indicates that the average price of CPA-TAT and CPA-AUC also increases correspondingly. Fig. 8 (b) shows that as traffic multiplexing increases, the blocking rate (or delayed admission rate) decreases under both CPA policies, and correspondingly, the average user benefit increases (Fig. 8 (d)). These performance benefits are due to the more efficient distribution of resources. As the number of user flows decreases, the impact of the bandwidth request of each user becomes larger, and user requests are more likely to be blocked during congestion under CPA-TAT, and delayed under CPA-AUC. When only 24 flows share the same bottleneck link, the blocking rate of CPA-TAT is high enough that it now has a lower average user benefit than CPA-AUC.

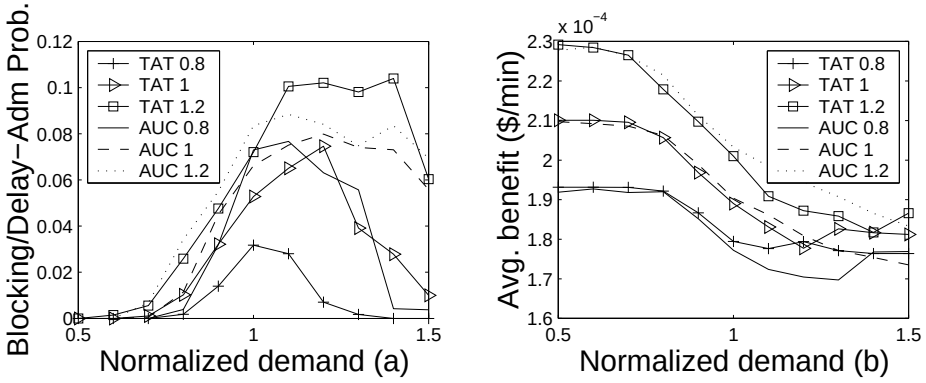

Fig. 9. Performance comparison between CPA-TAT and CPA-AUC as the elasticity factor w was scaled by 0.8, 1.0 and 1.2: (a) probability of request blocking for CPA-TAT and delayed admission for CPA-AUC; (b) average user benefit.

6.4 Impact of User Demand Elasticity

In this experiment, 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 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, w also represents a user's willingness to pay for bandwidth.

In previous simulations, the elasticity factor w was uniformly distributed between \$0.125/min and \$0.375/min. For this experiment, we did two simulations, one in which the default distribution of w was scaled upward by a factor of 1.2, and another with the distribution scaled down by a factor of 0.8. As the demand elasticity increases (i.e., a smaller w), individual bandwidth requests become smaller, and network resources can be distributed more efficiently because of the increased multiplexing. Accordingly, the request blocking probability of CPA-TAT and the delayed admission probability of CPA-AUC decrease as w is scaled down. In Fig. 9 (b), the average user benefit is seen to increase with w for both CPA policies. This is because a larger w indicates a higher user valuation of the resources and hence higher average user benefit. At a scaling factor of 1.2, the blocking rate of CPA-TAT is high enough that the average user benefit of CPA-TAT becomes smaller than that of CPA-AUC over most of the load-range.

6.5 Performance of A General Network Topology

In the experiments above, we studied the performance of CPA when the traffic shares a common bottleneck. In this section, we assume the more general

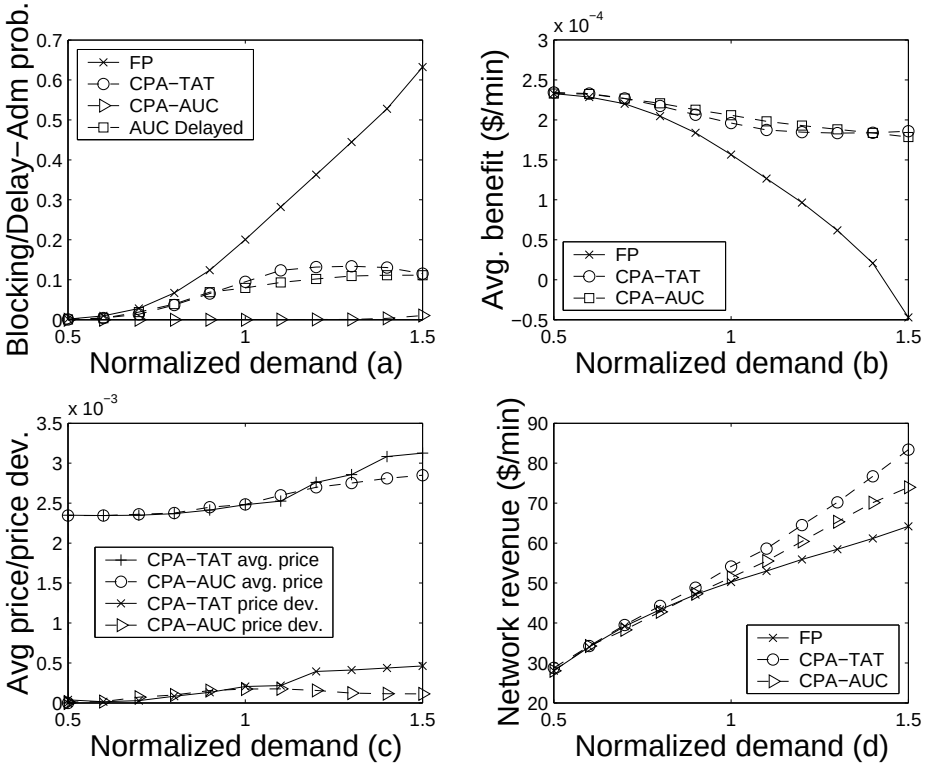

Fig. 10. Performance metrics of CPA-TAT, CPA-AUC, and FP policies as a function of normalized un-constrained demand with topology 2: (a) blocking probability; (b) average user benefit; (c) Time-average and standard deviation of system price of CPA-TAT and CPA-AUC; (d) total network revenue.

network topology of Fig. 2, with the potential for multiple bottlenecks to exist, and for these bottlenecks to interact.

In the simulation, traffic is generated symmetrically from all users, as described in Section 5. The five backbone links are the potential bottleneck links. We monitor the utilization at one of the backbone links, and calculate all the other parameters across the whole network. Fig. 10 shows the blocking probability, average perceived user benefit, and total network revenue as a function of normalized un-constrained demand, and the variation of the system price with time. All four metrics show trends similar to those for a single bottleneck, though the overall request blocking rate is higher than with a single bottleneck for all the policies. The variation with time of the average price under CPA-TAT is less smooth than the single bottleneck case due to the coupling of the traffic between different paths.

7 Conclusion

In this paper, we compare the performance of two key congestion pricing schemes in the network field, tâtonnement and auction, under a resource negotiation and pricing framework. To make the two pricing schemes comparable, and also to resolve some practical issues in auction models in the literature, we have developed an auction-based congestion pricing mechanism, based on a M-bid, second-price auction model.

The experimental results indicate that both pricing schemes can efficiently allocate resources during network congestion. They have generally comparable performance in terms of user-perceived benefit. CPA-AUC has a-priori knowledge of resource availability and user demand in making allocations. This allows CPA-AUC to achieve a high network utilization without increasing the blocking rate, which is effectively zero in our simulations (although some connections are delayed until the next auction). Resource allocation in CPA-TAT depends on the response of users to congestion price, which is an indirect signaling mechanism and has an inherent delay. Accordingly, CPA-TAT has to set its target utilization more conservatively than CPA-AUC in order to obtain a low blocking rate, and to obtain user-perceived benefit comparable to CPA-AUC. A conservative target utilization also makes the congestion pricing under CPA-TAT more aggressive than that of CPA-AUC, and results in higher network revenue. Setting a high target utilization allows CPA-TAT to match CPA-AUC in throughput, but at the cost of a high blocking rate, and lower user benefit than CPA-AUC. The blocking rate of CPA-TAT can also become excessive under certain other conditions (for example, a long negotiation interval, low demand elasticity, or a small number of high-bandwidth users), making CPA-AUC better in terms of overall user benefit, because of the higher number of admitted users.

Both CPA policies function effectively over the range of control parameters considered in our experiments. In particular, the performance of CPA-AUC is seen to be robust to the number of bids M. This is useful because a smaller number of bids reduces implementation complexity and overhead. Both policies generally perform better with a shorter control or negotiation interval. In choosing the length of the interval, performance has to be traded against signaling and processing overhead. The performance of both CPA policies is also influenced by the number of users and their demand-elasticity. In general, the performance will improve if the users' bandwidth requirements are more elastic, and will also improve as more connections share network resources. Both CPA policies also work in a network with multiple bottlenecks and more general topology.

Compared to other auction models used in the network literature, the M-

bid auction model provides greater predictability and availability of service, by allowing users to express their willingness to pay for a number of levels of bandwidth in advance of the auction. Submitting multiple bids simultaneously also reduces signaling traffic during auctions by avoiding multiple or more frequent bid submissions. The main drawback of CPA-AUC relative to CPA-TAT is the higher implementation complexity, particularly if auctioning is implemented per-node, instead of centralized auctioning per-domain. For CPA-TAT, only the price for each service class needs to be maintained at a node, and the user resource demands can be aggregated at network core [11]. However, it is not easy to aggregate users' bids, and auction may not be appropriate for being performed at per-user level in a large scale network. From the user perspective, one drawback is the set-up delay for new connections during congestion (this can be reduced by setting aside a fraction of available resources during auctions, at the cost of lower network utilization). A second problem inherent in this and other existing auction schemes is the difficulty in dividing the user willingness-to-pay for a particular bandwidth among multiple nodes or domains, in order to obtain end-to-end resource reservation. In our experiments, we assume that the user bids the same price at all the nodes/domains. Also, if the user receives a smaller bandwidth at the bottleneck node compared to other nodes, the network does not attempt to re-allocate the excess bandwidth at other nodes, because of the added complexity. In our experiments, all the connections share the same bottleneck node, so the equal division of the bids among different hops does not influence the comparison results between CPA-TAT and CPA-AUC. In practice, if users can determine the bottleneck node(s), they will distribute their willingness-to-pay more optimally. But this may make the resource allocation less predictable, and result in unfair resource distribution. Another drawback of auction is the need to reveal user preferences to the network. We would like to study the impact of bid division and other auction-related issues on network performance in future work.

References

[1] K. Maney, "Future not so bright for telecoms," in *USA Today*, Jul 2002.

[2] X. Wang and H. Schulzrinne, "Comparison of adaptive internet multimedia applications," *IEICE Transactions on Communications*, vol. 82, pp. 806–818, Jun 1999.

[3] X. Wang and H. Schulzrinne, "Pricing network resources for adaptive applications in a differentiated services network," in *Proc. of Infocom*, (Anchorage, Alaska), Apr 2001.

[4] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational*

Research Society, vol. 49, pp. 237–252, 1998.

[5] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[6] A. Ganesh, K. Laevens, and R. Steinberg, "Congestion pricing and user adaptation," in *Proc. of Infocom*, (Anchorage, Alaska), Apr 2001.

[7] S. H. Low and D. Lapsley, "Optimization flow control–I: basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, pp. 861–874, Dec 1999.

[8] J. F. MacKie-Mason and H. Varian, "Pricing the internet," in *Kahin and Keller (eds): Public Access to the Internet*, (Cambridge, MA), pp. 269–314, MIT Press, 1995.

[9] J. F. MacKie-Mason, "A smart market for resource reservation in a multiple quality of service information network," technical report, University of Michigan, Sept. 1997.

[10] N. Semret and A. Lazar, "The progressive second price auction mechanism for network resource sharing," in *8th International Symposium on Dynamic Games*, (Netherlands), Jul 1998.

[11] X. Wang and H. Schulzrinne, "An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications," in *IEEE Journal on Selected Areas in Communications*, vol. 18, 2000.

[12] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang, "Pricing in computer networks: Motivation, formulation, and example," *IEEE/ACM Trans. Networking*, vol. 1, Dec 1993.

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

[14] D. F. Ferguson, C. Nikolaou, and Y. Yemini, "An economy for flow control in computer networks," in *Proc. of Infocom*, (Ottawa, Canada), pp. 110–118, IEEE, Apr 1989.

[15] 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)*, Oct 1998.

[16] J. Sairamesh, "Economic paradigms for information systems and networks," in *PhD thesis, Columbia University*, (New York), 1997.

[17] H. Varian, *Microeconomic Analysis*. W.W. Norton & Co, 1993.

[18] J. shu and P. Varaiya, "Pricing network services," in *Proc. of Infocom*, (San Francisco, USA), IEEE, Apr 2003.

[19] G. Fankhauser, B. Stiller, C. Vogtli, and B. Plattner, "Reservation-based charging in an integrated services network," in *Proceedings of the 4th Informs Telecommunications Conference*, (Boca Raton), Mar 1999.

[20] P. Reichl, G. Fankhauser, and B. Stiller, "Auction models for multi-provider internet connections," in *Tagungsband zur 10. GI/ITG-Fachtagung Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB '99)*, Sept. 1999.

[21] 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.

[22] R. Vaccaro, *Digital control, a state space approach*. McGraw Hill, 1995.

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

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

[25] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," Request for Comments 2598, Internet Engineering Task Force, Jun 1999.

[26] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," Request for Comments 2597, Internet Engineering Task Force, Jun 1999.

[27] C. Lambrecht and O. Verscheure, "Perceptual quality measure using a spatio-temporal model of human visual system," in *Proc. of IS&T/SPIE*, Feb 1996.

[28] A. Watson and M. A. Sasse, "Evaluating audio and video quality in low-cost multimedia conferencing systems," in *Interacting with Computers*, vol. 8, pp. 255–275, 1996.

[29] X. Wang, *Scalable Network Architectures, Protocols and Measurements for Adaptive Quality of Service*. PhD thesis, Columbia University, New York, 2001.

[30] Virtual InterNetwork Testbed, "The network simulator - ns (version 2)." http://www.isi.edu/nsnam/ns/.

[31] M. Creis, "RSVP/NS: An implementation of RSVP for the network simulator NS-2," http://www.isi.edu/nsnam/ns/ns-contributed.html.

[32] Common Carrier Bureau, "Trends in telephone service," tech. rep., Federal Communications Commission, Washington, D.C., Dec 2000.