Incentive-Compatible Adaptation of Internet Real-Time Multimedia

Xin Wang, *Member, IEEE,* and Henning Schulzrinne, *Member, IEEE*

***Abstract*— The rapid deployment of new applications and the inter-connection of networks with increasing diversity of technologies and capacity make it more challenging to provide end-to-end quality assurance to the value-added services, such as the transmission of real-time multimedia and mission critical data. 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 multimedia applications to adapt their sending rates according to network conditions. We have developed an intelligent service architecture that integrates resource reservation, negotiation, pricing and adaptation in a flexible and scalable way. In this paper, we present a generic pricing structure that characterizes the pricing schemes widely used in the current Internet, and introduce a dynamic, congestion-sensitive pricing algorithm that can be used with the proposed service framework. We also develop the demand behavior of adaptive users based on a physically reasonable user utility function. We introduce our multimedia testbed and describe how the proposed intelligent framework can be implemented to manage a video conference system. We 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 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 to achieve significant gains in network availability, revenue, and user-perceived benefit relative to the fixed-price policy. 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 congestion-based adaptive policy further improves as the network scales and more connections share the resources. Finally, we complement the simulation with experimental results demonstrating important features of the adaptation process.**

***Index Terms*— Adaptation, incentive, multimedia, pricing, resource allocation, congestion control.**

I. Introduction

Many new applications begin to be widely used in the Internet. These applications include real-time audio, video, and mission-critical financial data. The new value-added services provide new business opportunities, but also present new challenges. The Internet's lack of control over quality of service (QoS) has slowed down the deployment of these value-added services. Even though the capacity of the backbone networks has been considered enough and the average link utilization is always reasonably low, the traffic statistics released by several ISPs [1][2][3][4] indicates that every network always has some busy links (particularly, access links at network access points and peering points) that have long lasting high bandwidth utilization. The rapid deployment of new applications and the inter-connection of networks with increasing diversity of technologies and capacity make it more challenging to provide end-to-end quality assurance to the value-added services. On the other hand, multimedia applications on the Internet commonly employ the UDP transport protocol, which lacks a congestion control mechanism. These applications can therefore starve TCP applications (which perform congestion control) of their fair share of bandwidth.

To address these problems, one approach is to enhance the network with mechanisms such as resource reservation [5][6], admission control [7], special scheduling mechanisms [8], and differentiated services [9][10][11]. Another approach is to adjust the bandwidth used by an application according to the existing network conditions [12], relying on signaling mechanisms such as packet loss rates for feedback.

If the resource reservation is done statically (before transmission), resource allocation and provisioning have to be conservative to be able to meet QoS assurances in the presence of short- and long-term network traffic dynamics during the life of the application. Many multimedia applications are long-lived, exacerbating the problem. Allowing only static resource reservation unavoidably imposes higher resource costs and hence higher charges to the users. 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. There has been a lot of recent work that tries to address this problem - by dropping more packets to punish unresponsive applications, and by enforcing TCP like fairness [13][14][15][16]. However, these methods do not take into account the fact that some sources may not be able to reduce their transmission rate easily and TCP like rate adaptation does not work well for multimedia applications. Therefore, when congestion happens, these kinds of fairness schemes may not be appropriate for applications to meet individual QoS expectations.

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. 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. Unlike best-effort adaptive approaches, applications are guaranteed resources and there is no assumption that applications are cooperative. Our framework offers a middle ground, where resources are reserved, but resource commitments are made only for short intervals, instead of indefinitely. Prices may vary for

Manuscript received December 1, 2003; revised May 18, 2004. This work was sponsored by NSF CAREER grant.

Xin Wang is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA (e-mail: xwang8@cse.buffalo.edu)

Henning Schulzrinne is with Department of Computer Science, Columbia University, New York, NY 10027, USA (e-mail: hgs@cs.columbia.edu)

each interval, encouraging applications to adjust their resource demands to network congestion. Our model hence allows the network operator to create different trade-offs between blocking admissions and raising congestion prices to motivate the rate and service adaptation of applications to the varying network conditions, technologies and platforms. Upon congestion, the network can adjust congestion price periodically on the time scale of a minute or longer, encouraging the adaptation-capable applications to adapt their sending rates or select a different service class. Since the time period between price adjustments is relatively long, the network transmission delay has negligible impact on the system performance.

In earlier work, we presented a Resource Negotiation and Pricing (RNAP) protocol and architecture [17]. RNAP enables negotiation between user applications and the access network, as well as between adjoining network domains, and also enables the distribution and collation of price and charging information. RNAP allows the users to select from available network services with different QoS properties and re-negotiate contracted services, and allows the network to dynamically formulate service prices and communicate current prices to the users. Although dynamic re-negotiation and pricing are integral features of RNAP, it is compatible with applications with different capabilities and requirements. Applications may choose services that provide a fixed price, and fixed service parameters during the duration of service. Alternatively, if they are not constrained by a fixed user budget, they may use a service with usage-sensitive pricing, and maintain a constant QoS level, paying a higher charge during congestion. Generally, the long-term average cost for fixed-price service is higher since the network provider will add a risk premium. Applications may also be *adaptive*, that is, operate with a budget constraint, and adjust their service requests in response to price increases during congestion.

RNAP framework enables us to develop an intelligent service architecture that integrates resource reservation, negotiation, pricing and adaptation in a flexible and scalable way. However, the pricing algorithms and adaptation framework presented in this paper do not depend on any specific network architecture or protocol. It is possible to extend other existing network signaling protocols to support resource negotiation. In Section VII, we will show that our testbed implementation extended RSVP [5] to support price quote and resource negotiation. A network domain manages its own pricing scheme (which may be congestion sensitive or static) independent of other domains, and the domain electing to support congestion pricing could convey the updated prices to the end users through a signaling protocol. The deployment of the resource negotiation infrastructure, however, can be incremental. The negotiation component can be implemented as an opaque object [5] carried in the signaling protocol and left untouched when the signaling message passes by the domain not supporting service negotiation. In this case, user adaptations will only be based on the conditions of the networks which support congestion pricing and provide network statistics. On the other hand, the user that would like to adapt its applications according to network conditions can negotiate resources with the network through a user agent, located at the user site or at the network access point. A user does not need to be aware of the underlying negotiation mechanism, but only needs to provide his budget and minimum bandwidth or quality of service requirement (which can then be translated into corresponding bandwidth) for his applications. Instead of notifying the users explicitly about the bandwidth price, a network provider can sell its services as packages. A service package that supports user service adaptations can be sold at lower price, and a user may only perceive some quality degradation upon network congestion but does not need to be aware of the resource negotiation process.

In this paper, we present a generic pricing structure that characterizes the pricing schemes widely used in the current Internet, and introduce in more details a dynamic, congestion-sensitive pricing algorithm that can be used with the proposed service framework. We also develop the demand behavior of adaptive users based on a physically reasonable user utility function. We show how a set of user applications performing a given task (for example, a video conference) can adapt their sending rate and quality of service requests to the network in response to changes in service prices and subject to budget and minimum quality requirements, so as to maximize the total benefit to the user. We introduce our multimedia testbed and describe how the proposed intelligent framework can be applied to a video-conference system. 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, but we try to derive results and conclusions applicable to static and congestion-driven, dynamic pricing schemes in general. We complement the simulation with experimental results demonstrating important features of the adaptation process.

This paper is organized as follows. In section II, we briefly describe the RNAP architecture, as an example of the environment in which incentive-driven adaptation takes place. In Section III, we discuss various network pricing models and their suitability. We discuss in detail a volume-based, congestion-sensitive pricing strategy, which was introduced briefly in [17]. In Section IV, we consider user adaptation in response to congestion-dependent pricing. We present a physically reasonable form of user utility function, and derive a specific demand function for a given network price based on this utility function. In Section V, we describe how this adaptation framework is implemented in a real multimedia system environment. In Section VI, we first introduce the simulation topology and parameters, and performance metrics. We then discuss simulation results in detail. In Section VII, we introduce our test-bed set up for a multimedia system, and show the experimental results. In Section VIII, we describe some related work. We summarize our findings in Section IX.

II. Resource Negotiation through RNAP

The pricing algorithms and adaptation framework presented in this paper do not depend on any particular network architecture or protocol. However in this paper, we simulated our results in an environment supporting dynamic service negotiation through the Resource Negotiation and Pricing protocol (RNAP) [17][**?**], using a distributed (RNAP-D) network management architecture. We first briefly review the basic RNAP framework, and then describe the aggregation of RNAP messages for scalability.

We assume that the network provides services with certain QoS characteristics to user applications, and charges prices for these services. The service prices may 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 through a *Quotation*

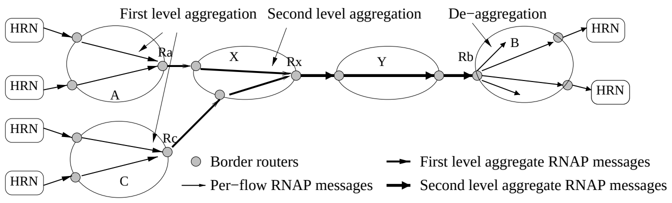

Fig. 1. Example RNAP-D message aggregation.

message. Based on this information and current application requirements, the HRN determines the 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 towards the HRN.

If end-to-end RNAP reservation is carried out for each customer flow, RNAP agents in the core network may potentially need to process RNAP messages for hundreds of thousands of flows, and maintain state information for each of them. To reduce the overhead due to per-flow RNAP message processing and storage, we consider the aggregation of RNAP messages belonging to senders sharing the same destination network address, forming a "sink tree" as shown in Fig. 1. Sink tree based aggregation has also been discussed in [18][19]. RNAP messages which request the same or similar services and have similar negotiation intervals are merged by the source domain and split again for each individual HRN at the border router of the destination domain. The merging point in the HRNs home network forwards two messages: one that travels directly to the destination network, without visiting any of the RNAP agents in between, and an aggregated-resource message that reserves resources and collects prices in the "middle" of the network.

The merged resource message have a resource request which is equal to the sum of all the branch resource requests further up in the sink tree. At each merging point, upstream flow arrivals, departures and reservation changes will trigger the update of the downstream merged request. To avoid frequent re-negotiation, the merging point may decide to reserve more resources than the sum of the upstream requests and add resources in larger increments if the current downstream allocation has been reached or is about to be reached.

III. Pricing Strategies

Each network must generate enough income to cover its costs, and is free to set its own policy to do this. A few pricing schemes are widely used in the Internet today [20]: 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. With volume charging, the user pays by the megabytes for traffic in one or both directions. Since an AC charging scheme is usually dependent on the user's access speed, it can be considered a coarse form of volume charging. 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 not used in the current Internet.

User experiments [21] 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 appears to be more appropriate for this purpose.

In this paper, we study two kinds of volume-based pricing: a fixed-price (FP) policy with a fixed unit volume price, and a congestion-price-based adaptive service (CPA) in which the unit volume price has a congestion-sensitive component. We first introduce the fixed pricing policy, and then describe the latter system in more detail, and also present a generic pricing framework to accommodate the different pricing models.

A. Fixed Pricing

In the fixed price model, the network charges the user per volume of data transmitted, independent of the congestion state of the network. The per-byte charge can be the same for all service class ("flat", FP-FL), depend on the service class or priority (FP-PR), depend on the time of day (FP-T) or a combination of time-of-day and service class (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.

B. Congestion-based Pricing

In estimating the normal load of Internet, one cannot rely on statistical sharing. There is a growing body of research showing that network traffic is self-similar in nature [22]. An effect of this self-similarity is that one does not see a smoothing of traffic peaks as the number of users sharing a link increases; instead the aggregate traffic remains bursty with peak increasing in proportion to the number of users. This suggests that we can never completely avoid network congestion. Instead, the provider must plan to keep it at a level acceptable to network users. There are two ways to approach this. First, one can apply technology to share the available bandwidth, e.g., TCP's exponential backoff, or use service provision policy to hold the promise of providing users with the level of service they require for a session. Second, one may use economics to influence users' behavior. We have discussed the tradeoff between different schemes in Section I, and their applicability to the multimedia applications.

If the price does not depend on the congestion conditions in the network, customers with less bandwidth-sensitive applications have no motivation to reduce their traffic as network congestion increases. 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 scarcity, 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. The total price of CPA will be composed of a fixed volume-based charge and a component that depends on congestion. Thus, with four variations on the fixed volume-based charge outlined above, we have the pricing models CP-FL, CP-PR, CP-T, CP-PR-T. This is summarized in Table 1.

We assume that routers support multiple service classes and that each router is partitioned to provide a separate link bandwidth and buffer space for each service, at each port. We consider one of the classes. We use the framework of the competitive market model [23]. 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. 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. The congestion price will be levied once demands exceeds a provider-set fraction of the available bandwidth or buffer space. We now discuss the formulation of the fixed charge, which we decompose into *holding charge* and *usage charge*, and the formulation of the *congestion charge*.

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

$$c_u(n) = p_u V(n). \tag{1}$$

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 (buffer space or bandwidth) 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) is therefore set to reflect the difference between the usage price for that class (e.g., i) and the usage price for the next lower service class (e.g., i - 1) and can be represented as:

$$p_h = p_u^i - p_u^{i-1}. \tag{2}$$

The holding charge $c_h(n)$ when a customer reserves bandwidth $R(n)$ during time period n is given by:

$$c_h(n) = p_h(R(n)\tau - V(n)), \tag{3}$$

where τ is the length of a negotiation interval, $V(n)$ is the traffic sent by user over the period n, and $R(n)\tau - V(n)$ is the bandwidth not used by the user. $R(n)$ 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 a 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) 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 [23]:

$$p_c(n) \quad = \quad \min[\{p_c(n-1) + \sigma(D,S)(D-S)/S, 0\}^+, p_{max}], \tag{4}$$

where D and S represent the current total demand and supply respectively, and σ is a factor used to adjust the convergence rate. The parameter 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 calculated price increment exceeds $\theta p_c(n-1)$. The maximum congestion price is bounded by p_{max}. 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 class. For a period n, the congestion charge is given by

$$c_c(n) = p_c(n)V(n). \tag{5}$$

Based on the price formulation strategy described above, a router arrives at a cost for a particular flow or flow-aggregate at the end of each price update interval. The total charge for a session is given by

$$c_s = \sum_{n=1}^{N} [p_h(R(n)\tau - V(n)) + (p_u + p_c(n))V(n)], \tag{6}$$

where N 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 reserving resources 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.

C. 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

Charging Scheme	Access	Connection Time	Holding	Usage	Congestion	Class-Based	Time-dependent
AC	yes	yes					
FP-FL	optional		yes	yes			
FP-PR	optional		yes	yes		yes	
FP-T	optional		yes	yes			yes
FP-PR-T	optional		yes	yes		yes	yes
CP-FL	optional		yes	yes	yes		
CP-PR	optional		yes	yes	yes	yes	
CP-T	optional		yes	yes	yes		yes
CP-PR-T	optional		yes	yes	yes	yes	yes

TABLE I

THE CHARGING STRUCTURE OF DIFFERENT SCHEMES

generic equation represents the charge incurred by a customer for a single billing cycle in all these cases:

$$\begin{aligned} cost &= c_{ac}(R_{ac}) + p(R_{ac})(t - T_m)^+ \\ &+ \sum_{i=1}^{I} \sum_{n=1}^{N_b} [p_h^i(n)(R^i(n)\tau - V^i(n)) \\ &+ (p_u^i(n) + p_c^i(n))V^i(n)]. \end{aligned} \quad (7)$$

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, and other parameters have the same meaning as in Section III-B. 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 system). Generally, p_h and p_u vary only slowly, on the order of hours, while p_c changes much more rapidly. For the different charging modes discussed so far, equation 7 contains different items shown in table I.

As equation 7 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 $R_m \times \tau$. 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 that the results from the CPA and FP are applicable to all the CP and FP pricing models as well as the access charge inclusive models, since the most important and influential feature of the models is the presence or absence of congestion-dependent pricing.

IV. 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.

A. 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 monetary 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.

B. 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, 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:

$$\hat{U} = \sum_i [U^i(x^i)], \tag{8}$$

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

$$\begin{aligned} max \sum_i [U^i(x^i) - C^i(x^i)] \\ \text{s. t.} \sum_i C^i(x^i) \leq b \\ x^i_{min} \leq x^i \leq x^i_{max}, \end{aligned} \tag{9}$$

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 [25] under the int-serv model, Expedited Forwarding (EF) [10] and Assured Forwarding (AF) [11] 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 [26] to search for the optimal service requests in such a framework.

C. 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 [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. Hence, a utility function can be represented in a general form as:

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

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 10, the optimization process for a system with multiple applications as described in Section IV-B can be represented as:

$$\begin{aligned} max \sum_j [U_0^j + w^j \log \frac{x^j}{x^j_m} - p^j \times x^j] \\ \text{s. t.} \sum_j p^j \times x^j \leq b \\ \text{and} \quad x^j \geq x^j_m, \forall j. \end{aligned} \tag{11}$$

The Lagrangian for this problem is :

$$\begin{aligned} L(x^j, p^j, b) \quad = \quad & \sum_j [U_0^j + w^j \log \frac{x^j}{x^j_m} - p^j \times x^j] \\ & + \lambda[b - \sum_j (p^j \times x^j)] + \sum_j \mu^j (x^j - x^j_m) \end{aligned} \tag{12}$$

The first order conditions are thus:

$$\begin{aligned} L_{x^j} = \frac{w^j}{x^j} - (1+\lambda)p^j + \mu^j \leq 0, if <, x^j = 0 \\ L_\lambda = b - \sum_j p^j * x^j \geq 0, if >, \lambda = 0 \end{aligned} \tag{13}$$

$$L_{\mu^j} = x^j \geq 0, if >, \mu^j = 0. \tag{14}$$

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

$$\begin{aligned} L_{x^j} = \frac{w^j}{x^j} - p^j = 0, \\ \text{therefore,} \quad x^j = \frac{w^j}{p^j}. \end{aligned} \tag{15}$$

Hence, w^j represents the money a user would spend based on its perceived value for an application. If the budget is not a constraint, the bandwidth allocation for an application is simply equal to the user's willingness to pay for the application over the price of the requested service for the application, i.e., equal to the optimal bandwidth of the application.

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

$$L_{x^j} = \frac{w^j}{x^j} - (1+\lambda)p^j = 0 \tag{16}$$

$$b - \sum_j p^j \times x^j = 0 \tag{17}$$

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

$$x^j = \frac{b \times \frac{w^j}{\sum_l w^l}}{p^j}. \tag{18}$$

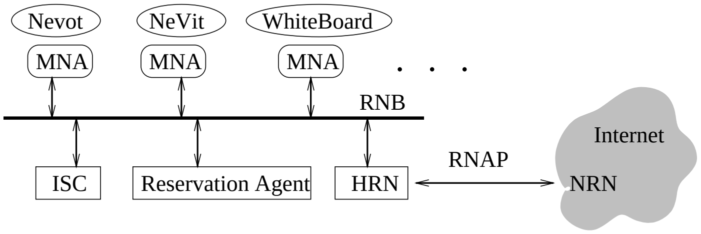

Fig. 2. The architecture of the extended MINT system

Therefore, 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 to 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.

D. 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 highly, he would increase U_0.

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

V. Resource Negotiation and Rate Adaptation in a Multimedia System

In the preceding section, we introduced the concept of application utility and system-wide utility. We explained how we define utility, and determine the sending rate and QoS parameters based on the maximization of user valuation surplus subject to budget constraints. We now consider how the above work may be applied in the context of a real multimedia system. As an example, we consider an extended version of the Multimedia Internet Terminal (MINT) [29] system, a flexible multimedia tool set that allows the establishment and control of multimedia sessions across the Internet. The various components of this extended version, and their interactions are shown in Fig. 2.

The principal application components of MINT are NeVoT and NeViT. Both NeVoT and NeViT support rate adaptation. NeVoT is an audio tool that allows the user to join different sessions simultaneously. The transmission quality of NeVoT can be changed by switching audio encoding during a transmission, with different participants being able to use different encodings at the same time. Currently the encoding algorithms used in NeVoT include LPC (5.6 kb/s), GSM (13.0 kb/s), DVI (32 kb/s), PCMU (64 kb/s), 16 bit/44.1 kHz high CD stereo (1411 kb/s). The adaptation of the audio rate in NeVoT is done by switching the coding algorithm used and in a discrete level.

NeViT is a video tool that is extended to achieve inter-media synchronization, automatic quality of service control and interaction with other media agents without being dependent on those agents. NeViT supports Sun Video card for capturing and compressing video images. The card supports JPEG, MPEG, CellB and YUV video in hardware and NeViT provides the appropriate algorithms for decompressing and displaying JPEG, MPEG, and YUV video images. Since video is more flexible in its bandwidth needs and thus lends itself more readily to adaptation, the video media agent NeViT is enhanced with a bandwidth adaptation algorithm that tunes the video frame rate to achieve different transmission data rate.

In addition to the above applications, the framework comprises of certain software agents - a Host Resource Negotiator (HRN), and a Media Negotiation Agent acting on behalf of each application. These agents exchange information over the Resource Negotiation Bus (RNB) by using a communication protocol called Pattern Matching Multicast (PMM) [30]. PMM messages are used for HRN and MNAs to exchange media parameters during a session, such as the bandwidth and frame rate of a video source, or the compression algorithm parameters for an audio. Since MINT allows decoupled media to work together, other media agents can easily be attached to the conference BUS without the necessity of changing the system structure. If a newly attached media supports rate adaptation, HRN will also send control message to inform the media to adjust its rate when necessary.

Each MNA communicates its application requirements (such as minimum bandwidth) and changes in requirements (for example, a temporary increase in application priority to accomplish a time-critical task) to the HRN. The HRN negotiates with the network through RNAP for delivery services with specific transmission bandwidths and other QoS parameters for each application. The HRN has a certain budget with which to obtain network services, and hence it can acquire a finite amount of network resources. It allocates these resources to the MNA's such that the system-wide benefit to the user is maximized. Every time the HRN receives updated prices from the network, it determines the optimal sending rate and service parameters for each application, and sends a control message on the RNB. Through this message, each MNA receives a target transmission bandwidth and certain QoS assurances. In turn, each MNA interacts with the media controller of its respective application to adjust its encoding process according to the targeted transmission rate and the QoS assurances it has received. In effect, the MNA hides the resource negotiation and allocation process from the application.

VI. Simulation Results and Discussion

In this section, we introduce the simulation topology and parameters, and performance metrics in Section VI-A. We then describe our simulation results that demonstrate some of the important features of our proposed adaptive reservation infrastructure in Section VI-B.

A. Simulation Model

The policies are simulated at the call level, based on the user-requested bandwidth, as opposed to packet-level. Depending on the service type and network infrastructure, the network may learn user resource requirements explicitly through a signaling protocol, or implicitly by traffic measurement. We simulate explicit resource reservation and price signaling through RNAP.

We used the *network simulator* [31] environment to simulate two different network topologies, shown in Fig. 3 and Fig. 4. Topology 1 contains two backbone nodes, six access nodes, and twenty-four end nodes. Topology two contains five backbone nodes, fifteen access nodes, and sixty end nodes. Topology two was also used in [32]. All links are full duplex and point-to-point. The links connecting the backbone nodes are 3 Mb/s, the links

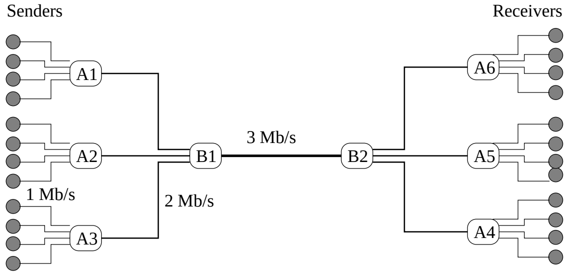

Fig. 3. Simulation network topology 1

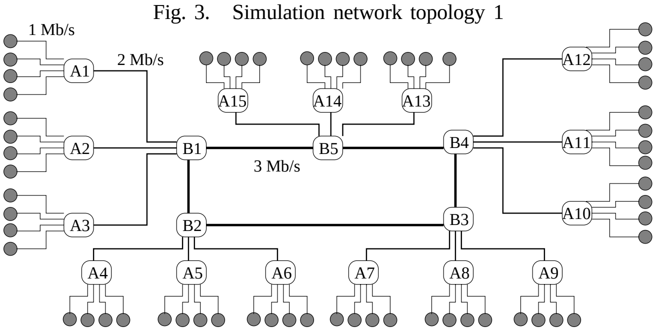

Fig. 4. Simulation network topology 2

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. 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 VI-B.7.

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, representative of a typical telephone call [33]. In topology 1, users from the sender side independently initialize unidirectional flows towards randomly selected receiver side end nodes. At most $12N_s$ flows (48 sessions with N_s set to 4) can run simultaneously in the whole network. In topology 2, all the users initialize unidirectional flows towards randomly selected end nodes. At most $60N_s$ users (360 sessions with N_s set to 6) are allowed to run simultaneously in the whole network.

The users are assumed to have the general form of the utility function shown in Section IV. w, the elasticity factor, (and also the user's willingness to pay) is uniformly distributed between \$ 0.125/min and \$ 0.375/min for a 64kb/s bandwidth. 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 his connection is dropped. Users re-negotiate their resource requirements with a period of 30 seconds in all the simulations.

The unit bandwidth price charged by the FP policy, and the unit bandwidth usage price charged by CPA, p_u, are both set to \$ 0.15/min for 64 kb/s transmission. The holding price p_h in the CPA policy is assumed to be zero, since all simulations are currently performed within a single service class, and interactions between service classes are not considered. 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 4 and the price adjustment threshold parameter θ, defined in Section III-B.3. Unless otherwise specified, values of $\sigma = 0.06$ and $\theta = 0.05$ are used.

In the simulation, we show the performance of the system for a range of *offered loads*. 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 were 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, due to insufficient provisioned resources. 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.
- *Price*: We monitor the end-to-end price quoted by the network during a simulation as a measure of the stability of the price adjustment / user adaptation process.
- *User charge*: A user is charged based on its bandwidth requirements during a user session and the corresponding price quoted by the network.
- *Network revenue*: Network revenue is the total charge paid to the network for all the admitted requests during a simulation.

B. Simulation Results

In this section, we show simulation results with the model described in Section VI-A.

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 VI-A. Figs. 5 (a)-(d) depict the results of the simulations

Fig. 5 (a) shows the variation of the utilization 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 threshold 1.1. 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.

Both policies admit all connections until the total link capacity is saturated. Fig. 5 (b) indicates that the blocking probability of FP scheme increases almost linearly as the offered load increases beyond 0.9, while the blocking rate of CPA increases initially and then starts to decrease after reaching a maximum at offered load

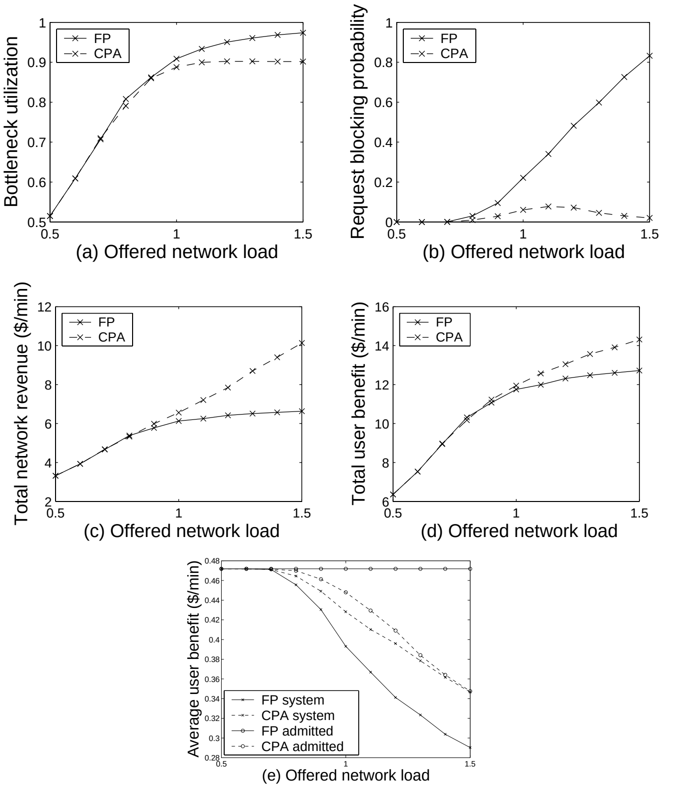

Fig. 5. Performance metrics of CPA and FP policies as a function of offered load: (a) bottleneck utilization; (b) blocking probability; (c) total network revenue; (d) total user benefit; (e) average user benefit.

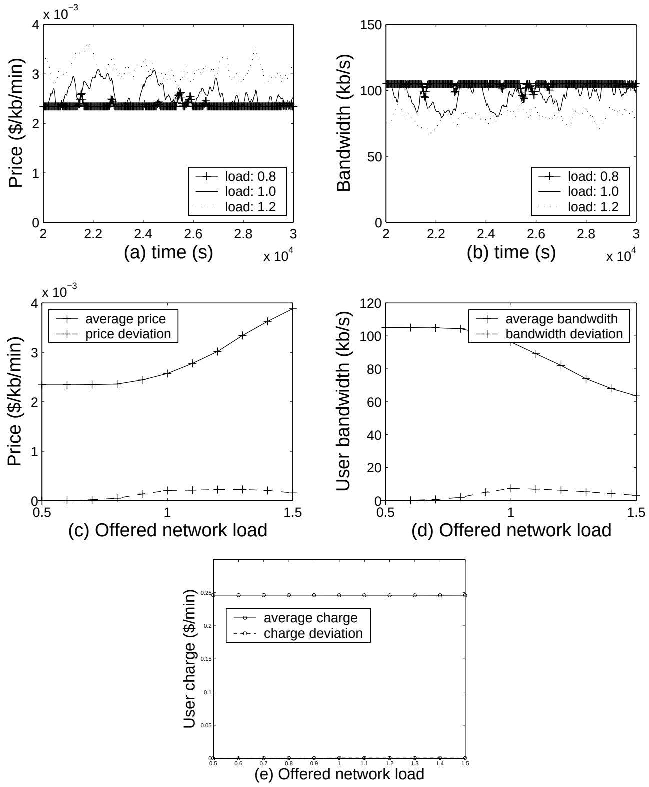

Fig. 6. System dynamics under CPA: variation over time of system price (a), and average user demand (b), at different offered load; time-average and standard deviation of system price (c), average user demand (d), and average user expenditure (e), plotted against offered load.

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 IV. The blocking probability of FP policy is almost 40 times larger than that of the CPA policy at the heaviest load.

Fig. 5 (c) compares the network revenue 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.

Fig. 5 (d) shows that 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 IV, 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.

Fig. 5 (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. Therefore, the average perceived benefit across all users decreases sharply with the load. The CPA has a much smaller blocking probability, which gives a higher average perceived benefit as load increases. 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. 6 (a)-(e).

Figs. 6 (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. 6 (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 speed of Fig. 5 (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, the increased multiplexing of user demand smoothes the total demand, and therefore reduces fluctuations in the price.

From the perspective of the user, the session cost (expenditure) and application level QoS performance are the most significant metrics. Fig. 6 (e) shows when the users adapt under the example utility function of Section IV, the user can operate at a stable expenditure, and therefore 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

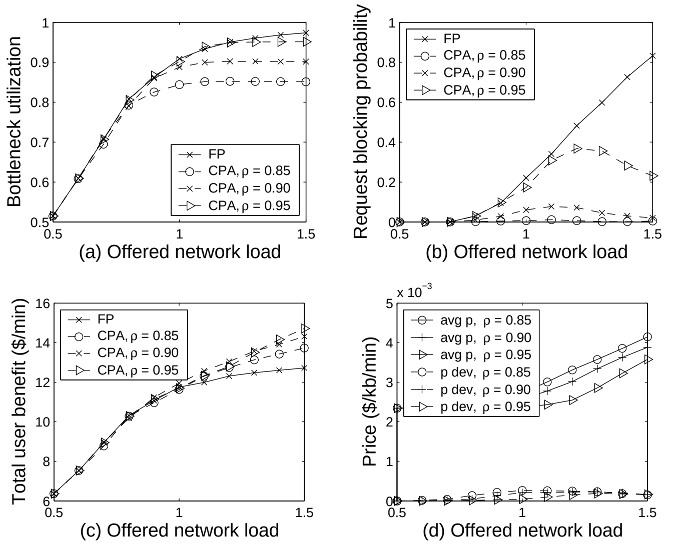

Fig. 7. Performance of CPA and FP policies at different values of target congestion control threshold ρ: (a) bottleneck utilization; (b) blocking probability; (c) total user benefit; (d) time-average and standard deviation of system price under CPA.

on the basic usage price and holding price values, which 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 (un-congested) price is set to correctly reflect long-term user demand, the congestion-based pricing should effectively limit short-term fluctuations in load.

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) ρ beyond which the congestion-dependent price component is imposed; the price scaling factor σ, used to control the rate at which a congested link is brought back to the targeted utilization; and the price adjustment threshold θ, which limits the frequency with which the price is updated. The parameters are varied one at a time.

In Fig. 7, 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 (Fig. 8). 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 (Fig. 9).

3) Effect 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 IV, w also represents a user's willingness to pay for bandwidth.

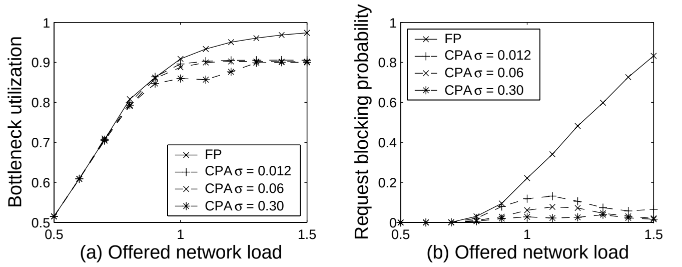

Fig. 8. Performance of CPA and FP at different values of σ: (a) bottleneck utilization; (b) blocking probability.

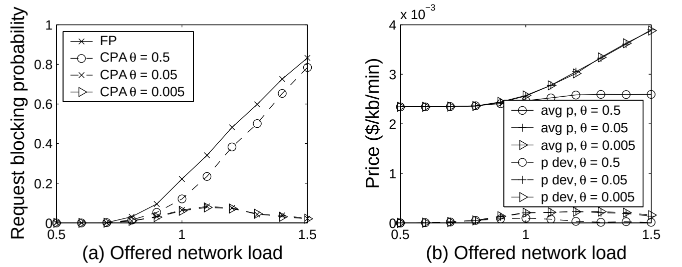

Fig. 9. Performance of CPA and FP at different values of θ: (a) blocking probability; (b) time-average and standard deviation of system price under CPA.

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 (Fig. 10). 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.

4) Effect of Session Multiplexing : We vary the number of customers sharing a system and evaluate the effect of the increased multiplexing of session requests under both CPA policy and FP policy as the number of sessions is increased. We keep the network topology and user utility distributions unchanged, but scale the link capacity proportionally with the maximum number of flows.

Fig. 11 (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. 11 (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

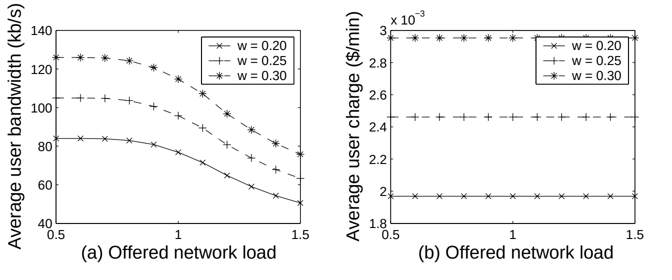

Fig. 10. Effect of the elasticity factor w on bandwidth allocation and user expenditure: (a) average bandwidth reserved by users with the three different values of w; (b) average expenditure of users with the three different values of w

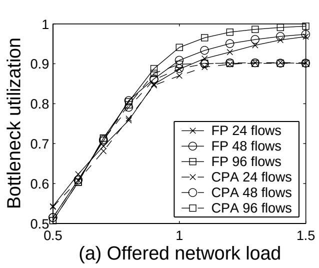

Fig. 11. Performance of CPA and FP with different number of customers sharing the system; (a) bottleneck utilization; (b) blocking probability.

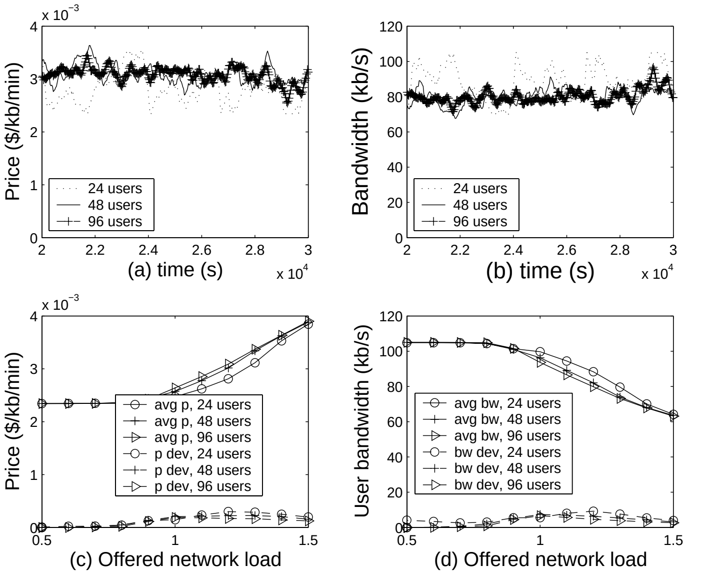

Fig. 12. System dynamics with different number of customers sharing the same bottleneck: variation over time of system price (a), and average user demand (b), at an offered load of 1.2; time-average and standard deviation of system price (c) and average user demand (d), plotted against offered load.

when the network is saturated. Under CPA, the blocking rate with 96 connections is up to 50 times smaller than that with 24 connections.

Fig. 12 depicts the price and demand dynamics as the network scales. Figs. 12 (a) and (b) show that the frequency of price and demand adjustment do not change appreciably with the 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. 12 (c) and (d).

The results in this section indicate that the performance of the CPA policy further improves as the network scales and more connections share the resources. Note that the performance improvement is due to the multiplexing of different user reservation requirements. This is different from the multiplexing of instantaneous user traffic, in which case the aggregate traffic may be self-similar.

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 requirements even at the highest price charged by the network. In this set of simulations, we restrict the maximum price so that the price does not increase without bound when all of the users are non-adaptive.

The results 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 - particularly up to a certain threshold load. Our results also indicate that the total user-perceived benefit increases with the proportion of adaptive users (not shown here).

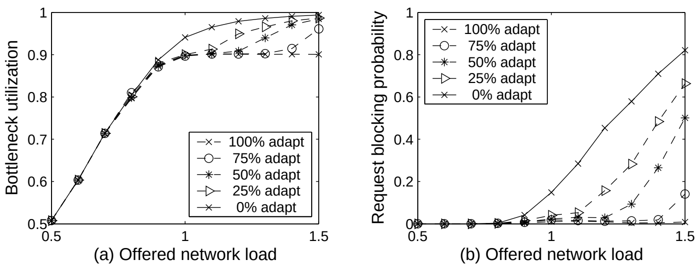

Fig. 13. Performance of CPA when only some of the users adapt their bandwidth requests; (a) bottleneck utilization; (b) blocking probability.

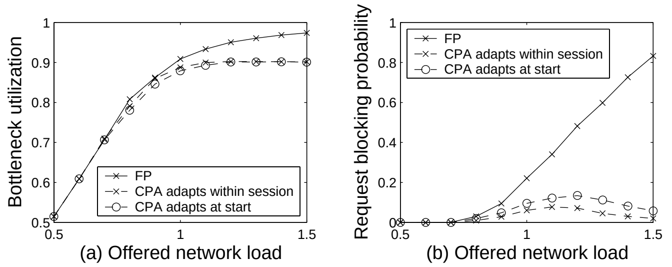

Fig. 14. Performance when CPA users select bandwidth only at session set-up, compared with performance when they continue to adapt during the session (a) bottleneck utilization; (b) blocking probability.

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 III-B 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) Session Adaptation and Adaptive Reservation: Under RNAP, applications can either pick a bandwidth when starting a session and keep that bandwidth during the session or adjust its resource demands during each negotiation interval. We refer to these modes as initial adaptation and ongoing adaptation, respectively.

Fig. 14 (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. 14 (b) shows that as expected, adaptation during a session allows for more efficient bandwidth usage and the blocking probability is reduced by half.

7) CPA Performance with Traffic Interactions from Different Paths: In the experiments above, we studied the performance of CPA when the traffic shares a common bottleneck. In this section, we assume network topology 2 in Fig. 4, 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. Note that in reality, the backbone links are normally over-provisioned. We target the backbone links to be bottlenecks only for the convenience of simulation. We monitor

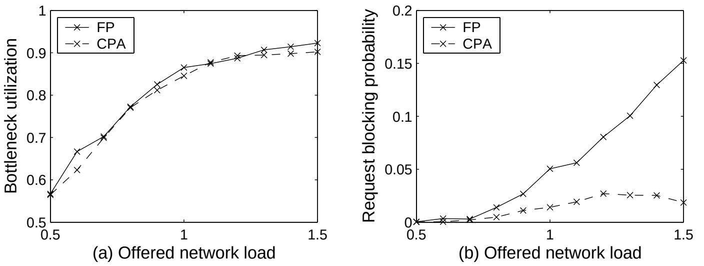

Fig. 15. Performance metrics of CPA and FP policies as a function of offered load using topology 2: (a) bottleneck utilization; (b) blocking probability.

the utilization at one of the backbone links, and calculate all the other parameters across the whole network. Fig. 15 (a) and (b) show that both the utilization and blocking probability have 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.

8) Other Mechanisms to Reduce Network Variations: The user adaptation behavior also influences the variation in bandwidth seen by application as well as the overall network behavior. A user can, for example, only request a change in bandwidth if the price change exceeds a given range. This reduces both the frequency of bandwidth adjustment and the user surplus. The initial adaptation described in Section VI-B.6 is the limit case where user reservation reflects only the price quoted at the beginning of the session.

A somewhat similar scenario can be envisioned in a core network, in which bandwidth reservation is carried out by network providers rather than by individual users. In this case, the providers can change their bandwidth requests in multiples of a large block of bandwidth, only when the user flow-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 [17].

VII. Experimental Results and Analysis

In this section, we describe our experimental results using a simplified implementation of RNAP. The implementation was based on an extension of the RSVP signaling protocol [17], and carried out on a test-bed consisting of two routers connected by a single 10 Mb/s link. An RNAP agent was implemented at each node. Two types of service were implemented - the traditional best-effort service, and the Controlled Load (CL) [34] service proposed within the int-serv model.

Although our implementation was highly simplified, it allowed us to demonstrate 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.

A. Experimental Setup and Parameters

The test-bed consisted of two routers (Ra and Rb) connected by a 10 Mb/s link, schematically represented in Fig. 16. Each interface at Ra and Rb had a capacity of 10 Mb/s, of which 4 Mb/s was configured to support the high priority CL service, and the remaining bandwidth was configured for best effort service. The congestion threshold was set to 70% of the CL capacity (2.8 Mb/s). Background traffic was also sent using best effort service.

We assumed a service roughly as expensive (per unit bandwidth) as a telephone line. Assuming a charge of 10 c/min for a capacity of 64 kb/s, the usage price is set as 2.6 c/Mb. Assuming that the next lower level of service is charged at 5 c/min, or 1.3 c/Mb, the holding price is set at 1.3 c/Mb. The price updation period was set at 30 seconds.

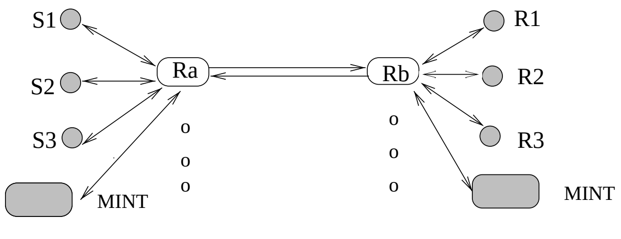

Fig. 16. Architecture of test-bed used for the experiments

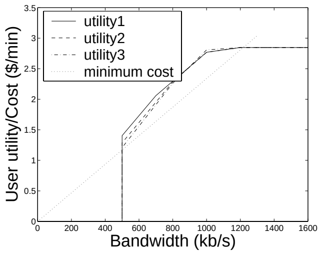

Fig. 17. Utility functions used for the three background applications

We assume that the budget available to each application is such that it can just afford the optimal sending rate when the link is uncongested. The metrics considered are: the behavior of the price in response to bandwidth demand, the influence of the price in driving adaptation of user bandwidth requirements, and the "benefit" gained by the applications in terms of the surplus (or perceived value of the service relative to its cost).

B. Experimental Results

We examine the adaptive behavior of the audio (NeVoT) and video (NeViT) applications in the MINT video conference system as well as three single applications referred as session 1, session 2 and session 3. As mentioned in Section IV-B, the application

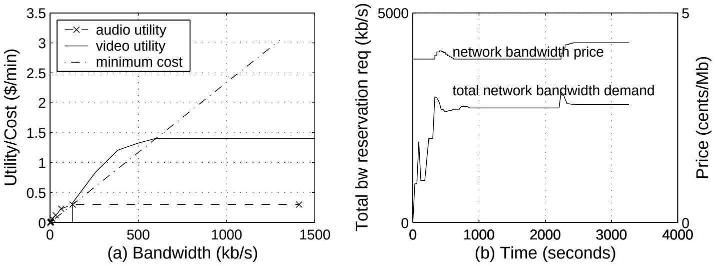

Fig. 18. (a) Audio and video utility functions used for adaptation by MINT; (b) Price and total bandwidth variation in the same experiment.

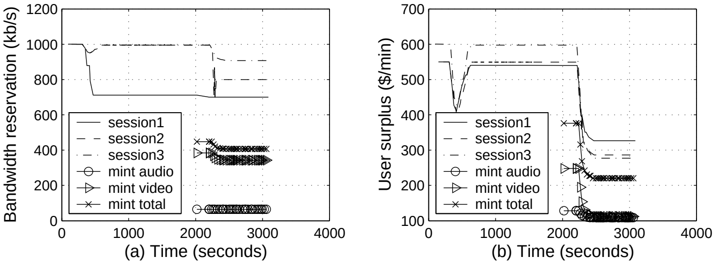

Fig. 19. Individual bandwidth reservations (a) and perceived surplus in the adaptation of Mint applications (b).

utility is likely to be measured by user experiments and known at discrete bandwidths and it is convenient to represent the utility as a piecewise linear function of bandwidth. Instead of using the example utility function described in Section IV-C, piecewise linear utility functions were used for the experiments to show the generality of our proposed multimedia adaptation framework. The simplified algorithm proposed in [26] was used by each application to search for the optimal service requests. The utility functions for the three background applications are shown in Fig. 17, and the utility functions for the audio and video applications are shown in Fig. 18 (a).

The three single user applications were started first, and shared the same output interface of the link. To create different levels of network load, a simple data source model was used in each session to continuously send UDP packets. The packet generation rate was tunable to allow user adaptation. The three applications are shown (Fig. 19) to reach stability at time 630 seconds with bandwidth allocations of 712 kb/s, 994 kb/s, and 994 kb/s respectively.

At the un-congested link bandwidth price, the optimal audio bandwidth for MINT is 64 kb/s, and the optimal video bandwidth is 384 kb/s. At time 2000 seconds, the MINT video conference system is started, and it first requests optimal bandwidth allocation (64 kb/s + 384 kb/s). The MINT applications compete for bandwidth with three single media applications belonging to different users. The total requested bandwidth exceeds the link congestion threshold, forcing the price up. It is observed the NeVoT bandwidth remains unchanged (with higher per-unit bandwidth valuation), and the NeViT bandwidth is reduced to 342 kb/s. The bandwidth share of the three competing user application drops to 700 kb/s, 800 kb/s and 907 kb/s respectively. User 1 has the most elastic bandwidth requirement between 700 kb/s and 1000 kb/s, and therefore initially gets a smaller share. But it is less elastic above 700 kb/s, and after the MINT applications are started, user 2, which has a relatively greater elasticity near its current allocation, reduces its requirement the most. The above experiment demonstrates the efficacy of the adaptation framework in allowing new sessions to join gracefully even when the network is highly loaded.

VIII. Related Work

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

A. 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) [35], and class-based *differentiated service* (diff-serv) [9]. In both architectures, implementations should include a mechanism by which the user can request specific network services, and thus acquire network resources. Current implementations of int-serv and 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 [36][37][38]. These studies were limited to a local system, and did not address the interaction of the local system with a large network. Liao [39] allocates resources to achieve equal perceived quality. In Section IV, we argued that perceived quality does not directly represent the economic value of communications.

B. Bandwidth Adaptation

In this section, we categorize approaches towards bandwidth adaptation in response to congestion, as summarized in Table II. The first row of Table II shows approaches that rely on reservation, and the second row shows approaches that do not. The columns correspond to adaptation at different time scales, decreasing from left to right. In the simplest form, the bandwidth of the application is constant and independent of the network condition. Examples include common streaming applications that simply attempt to send data or reserve a given bandwidth. Many applications can adjust their resource demand at the time of session creation. For reservation-based systems, OPWA [40] can be used to find out the available bandwidth. For best-effort systems, the end system may know its network access bandwidth and thus avoid requesting a 1 Mb/s stream when connected via a 28.8 kb/s modem.

Truly adaptive applications can adjust their resource usage on several different time-scales. In the table, we show time scales of minutes, seconds to several tens of seconds and on the order of a round-trip time. As far as we know, adjustable reservation on any time scale has not been studied extensively. A lot of recent research on adaptation is based on best-effort service, with signaling mechanisms such as packet loss rates for feedback [12]. For example, loss rates can be determined from RTP information [41], which is distributed on the order of five to several tens of seconds for modest-size receiver groups. Data applications can easily adjust their rate every round-trip time. However, adjustments more frequent than every minute or so are likely to be perceptually annoying to multimedia applications.

In earlier work, we described a Resource Negotiation and Pricing Protocol [17]. RNAP enables the network to periodically formulate service prices and communicate current prices to the user. Since RNAP focuses on dynamic re-negotiation and pricing, it allows the time scale of price updation and rate adaptation to be tailored to user requirements and service characteristics. In general, we envision a time scale of minutes for RNAP-based adaptation process.

C. Pricing and Billing in the Network

Microeconomic principles have been applied to various network traffic management problems. The studies in [42][43][44] 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.

Theoretical frameworks of congestion pricing have been discussed thoroughly by several authors [45][46][47][48]. Kelly et al [45] and Low et al [46] 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 [47] 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. Instead of only marking the packets, the authors in [48] proposed assigning each packet a price to reflect the congestion of the network. These schemes assume network services are best-effort, and rely on a pure market mechanism to maximize social benefit.

In [44][49][50], the resources are priced to reflect demand and supply. The methods in [44][49] are limited by their reliance on a well-defined statistical model of source traffic, and are

	fixed rate	adjust at conn. setup	adjust (∼min)	adjust (∼10s)	adjust (RTT∼100ms)
reservation based	telephone int-serv/diff-serv	int-serv/diff-serv, RNAP	RNAP	—	—
best effort based	current multimedia	based on access line speed	RNAP	adaptation in literature	TCP

TABLE II

COMPARISON OF ALGORITHMS FOR ADJUSTING BANDWIDTH IN RESPONSE TO CONGESTION

generally not intended to adapt to changing traffic demands. The scheme presented in [50] 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.

Some of the work above 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 and experiments, decoupled from specific network service protocols. Our work can therefore be regarded as complementary to some of the cited work.

IX. CONCLUSION

The rapid deployment of new applications and the interconnection of networks with increasing diversity of technologies and capacity make it more challenging to provide end-to-end quality assurance to the value-added services, such as the transmission of real-time multimedia and mission critical data. We have considered an intelligent framework for incentive-driven rate and QoS adaptation of multimedia applications. In the framework, users respond actively to changes in price signaled by the network by dynamically adjusting network resource usage by the applications, so as to maximize the perceived utility relative to the price, subject to user budget and QoS constraints. 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. We introduce our multimedia testbed and describe how the intelligent framework can be implemented to manage a video conference system.

One of the objectives of this paper is to study the performance of the incentive driven service adaption framework. Through extensive 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 (un-congested) price is set to correctly reflect long-term user demand, the congestion-based pricing should effectively limit short-term fluctuations in load.

We have also 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. Finally, our testbed results show the effectiveness of the intelligent service architecture in managing resources for a real-time video-conference system.

In this paper, 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. However, pricing in the presence of competition or alternative paths can coexist with our scheme. At the beginning of a session, a user can select the cheapest network and the cheapest path, while a user would adapt the service request during an on-going session to maintain the quality of an application.

REFERENCES

[1] SWITCH, “Switchlan traffic statistics.” http://www.switch.ch/lan/stat/.

[2] NORDUnet, “Nordunet network statistics.” http://www.nordu.net/stats/.

[3] ABOVE.net, “Above.net’s real-time network status.” http://stats.sjc.above.net/traffic/.

[4] BBC Internet Services, “Internet link usage.” http://support.bbc.co.uk/support/mrtg/internet/.

[5] 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, Sept. 1997.

[6] P. Pan and H. Schulzrinne, “YESSIR: a simple reservation mechanism for the Internet,” *Computer commun. review*, vol. 29, pp. 89–101, Apr 1999.

[7] S. Jamin, S. J. Shenker, and P. B. Danzig, “Comparison of measurement-based admission control algorithms for controlled-load service,” in *Proc. of Infocom*, (Kobe, Japan), p. 973, Apr 1997.

[8] H. Zhang and S. Keshav, “Comparison of rate-based service disciplines,” in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Switzerland), pp. 113–121, Sept. 1991. also in Computer Communication Review 21(4) September 1991.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated service,” Request for Comments 2475, Internet Engineering Task Force, Dec 1998.

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

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

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

[13] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Trans. Networking*, vol. 7, pp. 458–473, Aug 1999.

[14] I. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A TCP-friendly rate adjustment protocol for continuous media flows over best effort networks," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, (Basking Ridge, New Jersey), Jun 1999.

[15] D. Lin and R. Morris, "Dynamics of random early detection," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Cannes, France), Sept. 1997.

[16] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Stockholm, Sweden), Aug 2000.

[17] 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)*, (Basking Ridge, New Jersey), pp. 77–93, Jun 1999.

[18] O. Schelen and S. Pink, "Resource reservation agents in the internet," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, (Cambridge, England), pp. 153–156, Jul 1998.

[19] P. Pan, E. Hahne, and H. Schulzrinne, "The border gateway reservation protocol (BGRP) for tree-based aggregation of inter-domain reservations," *Journal of Communications and Networks*, Jun 2000.

[20] 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.

[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] W. Leland, W. Willinger, M. Taqqu, and D. Wilson, "Statistical analysis amd stochastic modelling of self- similar data traffic," in *International Tele-traffic Conference 14*, (Ottawa, Canada), pp. 319–328, 1994.

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

[24] 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.

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

[26] 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), Jul 2000.

[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," *Interacting with Computers*, vol. 8, no. 3, pp. 255–275, 1996.

[29] D. Sisalem and H. Schulzrinne, "The multimedia internet terminal (MInT)," *Journal of Telecommunications*, vol. 9, pp. 423–444, Sept. 1998.

[30] H. Schulzrinne, "Dynamic configuration of conferencing applications using pattern-matching multicast," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'95)*, (Durham, New Hampshire), 1997.

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

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

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

[34] J. Wroclawski, "The use of RSVP with IETF integrated services," Request for Comments 2210, Internet Engineering Task Force, Sept. 1997.

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

[36] A. Hafid, G. V. Bochmann, and B. Kerherve, "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)*, (Syracuse, USA), 1996.

[37] T. F. Abdelzaher, E. M. Atkins, and K. Shin, "Qos negotiation in real-time systems and its application to automated flight control," 1999.

[38] 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*, Jun 1999.

[39] G. Bianchi, A. Campbell, and R.-F. Liao, "On utility-fair adaptive services in wireless networks," in *6th International Workshop on Quality of Service (IEEE/IFIP IWQOS'98)*, 1998.

[40] S. Shenker and L. Breslau, "Two issues in reservation establishment," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Cambridge, MA), Aug 1995.

[41] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan 1996.

[42] J. F. MacKie-Mason and H. Varian, "Pricing congestible network resources," *IEEE JSAC*, vol. 19, pp. 1141–1149, Sept. 1995.

[43] H. Jiang and S. Jordan, "A pricing model for high speed networks with guaranteed quality of service," in *Proc. of Infocom*, (San Fransisco, California), Mar 1996.

[44] 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.

[45] 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.

[46] 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.

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

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

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

[50] 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.

PLACE
PHOTO
HERE

Xin Wang received her BS and MS degrees in telecommunications engineering and wireless communications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1990 and 1993, respectively, and her PhD degree in electrical engineering from Columbia University, New York, NY, in 2001.

From 2001 to 2003, she was a Member of Technical Staff in the area of mobile and wireless networking at Bell Labs Research, Lucent Technologies, New Jersey. She is currently an Assistant Professor in the department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, New York. Her research interests include modeling and analysis of mobile and wireless networks, integrated network infrastructure design and performance enhancement across network layers, applications and heterogeneous networks, network and mobility management, QoS, signaling and control, as well as adaptive network services and applications.

PLACE
PHOTO
HERE

Henning Schulzrinne received his undergraduate degree in economics and electrical engineering from the Darmstadt University of Technology, Germany, his MSEE degree as a Fulbright scholar from the University of Cincinnati, Ohio and his Ph.D. degree from the University of Massachusetts in Amherst, Massachusetts. He was a member of technical staff at AT&T Bell Laboratories, Murray Hill and an associate department head at GMD-Fokus (Berlin), before joining the Computer Science and Electrical Engineering departments at Columbia University, New York. He is currently chair of the Department of Computer Science. His research interests encompass real-time, multimedia network services in the Internet and modeling and performance evaluation.

He is a division editor of the "Journal of Communications and Networks", and an editor of the "IEEE/ACM Transactions on Networking" and former editor of the "IEEE Internet Computing Magazine" and "IEEE Transactions on Image Processing". He is member of the Board of Governors of the IEEE Communications Society and the ACM SIGCOMM Executive Committee, former chair of the IEEE Communications Society Technical Committees on Computer Communications and the Internet and has been technical program chair of Global Internet, Infocom, NOSSDAV and IPtel. He also was a member of the IAB (Internet Architecture Board).

Protocols co-developed by him are now Internet standards, used by almost all Internet telephony and multimedia applications. His research interests include Internet multimedia systems, quality of service, and performance evaluation.

He serves as Chief Scientist for SIPquest Inc. and Chief Scientific Advisor for Ubiquity Software Corporation. He has received the New York City Mayor's Award for Excellence in Science and Technology and the VON Pioneer Award.