Incentive-Compatible Adaptation of Internet Real-Time Multimedia

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

Abstract

Distributed multimedia applications impose a number of requirements on the network to ensure acceptable quality. Two areas of active research in this context are the enhancement of networks with mechanisms such as resource reservation and special scheduling mechanisms, and the adaptation of network resource usage by applications depending on resource availability. A network with enhancements for QoS support, and usage and QoS dependent pricing, can use pricing as a natural feedback mechanism to drive adaptive behavior by applications. In this work, we consider a framework in which a multimedia application or system of applications carries out adaptation of sending rate and type of network service used. The adaptation is based on user utility, defined as the (monetary) value of the current service as perceived by the user, relative to the price charged by the network. When the multimedia system consists of multiple streams (audio, video, etc.), the framework enables resource requests for individual streams to be adapted for across-the-system maximization of value to the user. The system bandwidth is dynamically re-distributed among applications in response to changes in price, as well as changes in the relative utilities with time or under different application scenarios. Mechanisms for resource negotiation between the user and the network, and for price formulation in the network, are briefly discussed. Experimental results show that perceived value based adaptation allows bandwidth to be shared among competing users fairly. When network resources are scarce, bandwidth is shown to be distributed among competing applications (and among media streams belonging to a single multimedia system) according to their relative elasticity of demand, indicated by the sensitivity of the perceived value to the bandwidth. The effect of weighting (or multiplicatively scaling) and additively shifting the utility function is examined, in the context of changing the demand elasticity, and changing the importance of maintaining an interruption-free connection.

1 Introduction

The development and use of distributed multimedia applications are growing rapidly. These applications usually require a minimum Quality of Service (QoS) from the network, in terms of throughput, packet loss, delay, and jitter. Also, 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 [1][5], admission control [6], and special scheduling mechanisms [7]. 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 resource reservation is done statically (before transmission), resource reservation and provisioning tend to be conservative due to the lack of quantitative knowledge of traffic statistics. Moreover, the resource allocation is based on initial availability of resources and does not take into account changes in availability during an ongoing transmission. Many multimedia applications are long-lived, exacerbating the problem. 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 [30, 29, 31]. 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 [14]. 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. Existing research work in this area is discussed briefly in Section 7.

In earlier work, we presented a Resource Negotiation and Pricing (RNAP) protocol and architecture [14]. RNAP enables the user to select from available network services with different QoS properties and re-negotiate contracted services, and enables the network to dynamically formulate service prices and communicate current prices to the user. 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.

In this paper, we propose an algorithm for computation of a local or incremental price for a service at a given point in a network; We then propose some approaches towards adaptation of (multimedia) application sending rate and/or choice of network services in response to the incentive provided by dynamic network pricing. We discuss how to maximize user satisfaction in such an environment, subject to the constraints imposed by the minimum and maximum QoS requirements of the application, and the available budget. We also discuss the allocation of resources to component streams (audio, video, etc.) belonging to a multimedia system, for across-the-system maximization of value to the user. We present experimental results demonstrating important features of the adaptation process.

This paper is organized as follows. In section 2 of this paper, we briefly describe the RNAP architecture, as an example of the environment in which incentive-driven adaptation takes place. We then describe the proposed pricing algorithm. In Section 3, we presents some candidate adaptation algorithms. In Section 4, we discuss influence of the dynamic resource negotiation and network dynamics on the system stability. In Section 5, we describe how this adaptation framework is implemented in a real multimedia system environment. In Section 6, we present experimental results demonstrating some of the important features of our work Section 7 contains a brief discussion of related research. Finally, we summarize our work in Section 8.

2 Resource Negotiation and Pricing

In this section, we first briefly describe the RNAP protocol and architecture [14], as a typical framework within which incentive-driven adaptation by the user takes place. We then describe an algorithm for computation of a local or incremental price for a service at a given point in a network;

2.1 Resource Negotiation through RNAP

In the RNAP framework, we assume that the network makes services with certain QoS characteristics available to the user applications, and charges prices for these services that, in general, can 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 *Quotation* message. Based on this information and the current application requirements, the HRN determines the current optimal transmission bandwidth and service parameters for each application. It re-negotiates the contracted services by sending a *Reserve* message to the NRN, and receiving a *Commit* message as confirmation or denial.

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

2.2 Network Pricing and Congestion Control

We assume that the network charges the user appropriately for different services based on the user traffic volume, the QoS characteristics of the service, and the demand on network resources. It is not necessary to assume a particular pricing model for the purposes of the multimedia system. As with the resource negotiation system, however, we propose a simple pricing algorithm to determine a price for a particular kind of forwarding service from the router based on the competitive market model [43]. The competitive market model defines two kinds of agents: consumers and producers. Consumers seek resources from producers, and producers create or own the resources. The exchange rate of a resource is called its price. Prices are set where the amount of resource demanded equals the amount of resources supplied.

The router supports multiple services. We also assume that the router is partitioned to provide a separate link bandwidth and buffer space for each service, on each port. In the discussion that follows, we consider one such logical partition. The routers are considered as the producers and own the link bandwidth and buffer space for each output port. The flows (individual flows or aggregate of flows) are considered as consumers who consume resources. 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 price computation is performed periodically, with a price update interval τ. In general, the price update interval at a router is independent of the negotiation interval of the services supported by the router. The price within each negotiation interval is kept constant, to provide predictability to users. Prices are computed locally at routers, and collated to form an end-to-end price using the RNAP protocol.

We decompose the total charge computed at a router into three components: *holding charge*, *usage charge*, and *congestion charge*.

Usage Charge:

The usage charge is determined by the actual resources consumed, 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 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. It can be represented as:

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

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

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

Holding Charge:

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

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

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

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

where τ is the duration of the period. The $R(n)$ can be estimated from the traffic specification and QoS request of the customer, for example, an effective bandwidth.

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

Congestion Charge:

Network congestion arises from a scarcity of network resources, generally bandwidth and buffer space. The congestion charge is imposed only if 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 [43]:

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

Where D and S represent the current total demand and supply respectively, and σ is a factor used to adjust the convergence rate. σ may be a function of D and S; for example, it is 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 0, to protect against further congestion. The maximum congestion price is bounded by the p_{max} parameter so that the total price for a service class does not exceed that for a higher level of service. When a service class needs admission control, all new arrivals are rejected when the price reaches p_{max}. If p_c reaches p_{max} frequently, it indicates that more resources are needed for the corresponding service and new configuration for local resources may be needed. For a period n, the total congestion charge is given by

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

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

$$session\_charge = \sum_{n=1}^{N} (c_h(n) + c_u(n) + c_c(n)) \tag{7}$$

where N is total number of intervals spanned by a session.

3 User Adaptation

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 to the user. The user must define quantitatively, through a *utility function*, the benefit or value associated with a sending rate and QoS parameter-set obtained from a particular network service. Note a user utility function is private to the user itself. The rest of this section is divided into two parts. In Section 3.1, we discuss the definition and general characteristics of an utility function. In Section 3.2, we discuss how sending rate and transmission QoS parameters are selected based on the utility function.

3.1 The Utility Function

We consider a set of user applications, required to perform a task or *mission*, for example, audio, video, and whiteboard applications for a video-conference. The *Reserve* request from the user specifies certain transmission parameters for each application. In general, the transmission parameters are the sending rate, as well as QoS parameters, usually loss and delay. The *utility* of a reservation request denotes how beneficial the corresponding network resource allocation would be 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. The object of the adaptation is to select a set of transmission parameters that gives the maximum possible utility relative to the cost of obtaining this service, subject to the user budget constraint.

3.1.1 Utility as Perceived Value

Clearly, the utility of a transmission depends on its quality as perceived by the user. However, since the user is paying for the transmission, it appears reasonable to define the utility as the *perceived value* of that quality to the user. For example, an audio transmission requiring a certain sending rate and certain bounds on the end-to-end delay and loss rate may be worth 10 cents/minute to the user. The perceptual value is strongly correlated to the perceptual quality, but is not exactly the same. For example, a pair of audio transmissions encoded identically and with the same transmission QoS parameters also have the

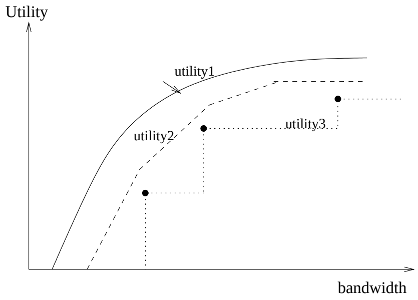

Figure 1: Some example utility functions

same perceived quality, but their perceived values may differ according to the application requirements. For example, the requirement of a weekly meeting between native speakers will probably have a lower quality requirement than a conferencing system teaching a foreign language and hence the users will see a higher value for using high quality transmission for the latter.

The measurement of subjective quality of multimedia transmissions has been reported by a number of researchers [19][23][24]. Generally, these experiments were intended to derive the Mean Opinion Score (MOS), which is measured as an average perceptive quality across a number of test subjects. But in our framework, perceived value very strongly reflects individual user preferences, and the application task being performed. We consider it likely, therefore, that an user application will have one or more of the following features:

- allow user to customize utility function(s)
- allow user to define “scenario”-specific utility functions; a particular scenario may be selected by the user during a session, or may be deduced by the application based on user actions
- allow user to specify a certain time-dependence of the utility function

3.1.2 Utility as a Function of Bandwidth

It is likely that only a few alternative services will be available to a multimedia application on the Internet **-** at the current stage of research, some possible services are guaranteed [4] and controlled-load service [**?**] under the int-serv model, Expedited Forwarding (EF) [10] under diff-serv, and several classes of service under Assured Forwarding (AF) [11], also under diff-serv. A particular user application would be able to choose from a small subset of the available services. Each such service would probably provide some qualitative or quantitative guarantee for loss and delay. It seems likely, therefore, that the user would develop an utility function as a function of the transmission bandwidth (which in turn would depend on specific encoding parameters such as frame rate, quantization, etc.), at different discrete levels of loss rate and delay.

We can make some general assumptions about the utility function as a function of the bandwidth, at a fixed value of loss and delay. The application has a minimum transmission bandwidth, and the utility is zero for bandwidth below this threshold. Also, user experiments reported in the literature [23][24] 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 and eventually goes to zero, defining a maximum QoS requirement.

Fig. 1 shows some possible utility vs. bandwidth curves. Utility1 is a smooth function. User experiments for deducing the utility function would be performed at discrete bandwidths, and some form of interpolation, such as a piecewise linear function (utility2), can be used to approximate the utility function. In addition, in some multimedia applications, only discrete bandwidths are feasible. For example, audio codecs can only operate at certain bandwidth points (Utility3).

3.1.3 Effect of Scaling

In this section, we consider how changes in utility function may influence the resource distribution. The operations we consider are an offset applied uniformly to the utility over all bandwidths, and multiplicative scaling of the utility function. We discuss the operation qualitatively here, and present some experimental results in section 6.2.4.

The utility function represents the relative preference of the user for different bandwidths. A constant (bandwidth-independent) offset to the utility function will not influence the resource distribution as long as the valuation of a bandwidth is higher than its cost.

On the other hand, a constant offset of the utility function changes the minimum perceived value. The minimum perceived value represents how much the user is willing to pay to just keep the application alive. Lowering this value allows the application to be terminated more readily during congestion (high cost). If a user values an uninterrupted service very highly, he increases the perceived value of the "keep alive" bandwidth.

A multiplicative scaling of the utility function by a factor greater than one tends to increase its bandwidth share, since it results in a bigger additive increase in perceived surplus at higher bandwidth than at lower bandwidths. Effectively, the demand elasticity of the application is reduced. The opposite effect is observed when the scaling factor is less than 1.

3.1.4 Time Dependence of Utility

For a particular application, the value of the information may vary with time. An user may perceive a higher value initially after the connection is established, and a lower value after a certain duration (typically, a phone call is very important to the user in the first one minute, compared to one that has lasted 30 minutes), or the reverse (for a movie, the ending is usually more interesting than the introduction). The relative importance of individual applications in a system may also evolve with time.

The evolution with time of the application utilities may be defined based on various user-defined scenarios. A simple way of representing the time evolution is to represent the multiplicative scaling and additive offset in Section 3.1.3, with a pair of time dependent parameters, α and β, so that the time-dependent utility can be represented as $\alpha_j(t) * U_j(\cdot) + \beta_j(t)$, where j represents a task performed at a time t.

3.2 Application Adaptation

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

We first consider the adaptation strategy of a single application when its utility is a function only of bandwidth (at a fixed loss and delay). We then discuss the adaptation strategy when the utility is function of multiple transmission parameters (bandwidth, loss and delay). Finally, we consider the problem of maximizing the *mission-wide* utility of a system comprising multiple applications performing a certain task. We assume the applications belong to a single user.

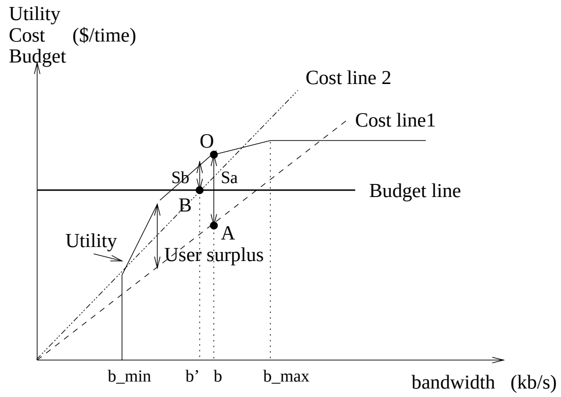

Figure 2: A perceived value based rate adaptation model

3.2.1 Adaptation of Single Application over Fixed Transmission Quality

If the quality of transmission is fixed (a particular delay and loss), the application utility (that is, the user-perceived value) increases monotonically with the bandwidth. Hence the maximization problem for the user can be written as:

$$\begin{aligned} \text{max } [U(x) - C(x)] \\ \text{s. t. } \ C(x) \leq b \\ x_{min} \leq x \leq x_{max} \end{aligned} \tag{8}$$

Where x is the bandwidth under consideration, $C(x)$ is the cost for the requested bandwidth, b is the budget of user, x_{min} is the minimum bandwidth requirement, and x_{max} represents the maximum bandwidth requirement. Note that U, b and c are in units of money/time.

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.

As mentioned earlier, the application utility is likely to be measured by user experiments and known at discrete bandwidths. In this case, it is convenient to represent the utility as a piecewise linear function, as shown in Fig. 2. The figure also assumes a constant unit bandwidth cost C, so that the cost-vs-bandwidth is a straight line with slope equal to C. The budget is shown as a horizontal line passing intercepting the cost/utility axis. From the figure, it is evident that the optimal bandwidth is

either the segment end-point with the highest surplus, if this end-point meets the budget constraint (b in Fig. 2 case A)

or else the bandwidth corresponding to the intersection point of the cost line with the budget line (b' in Fig. 2 case B).

3.2.2 Adaptation of Single Application over Multiple Transmission Parameters

We now consider the maximization of the application surplus over a set of transmission parameters (usually, the bandwidth, loss rate and delay). The objective function is as shown earlier in equation 8, but x, x_{min} and x_{max} are now vectors corresponding to the set of transmission parameters. If a complete quality of service parameter space is considered, the searching cost can be prohibitive. As briefly explained however, we believe it is likely that the application utility will take the form of a small set of utility versus transmission bandwidth functions, each at a different level of loss rate and delay, corresponding to a particular service. In this case, the optimization routine is as follows:

1. For each available service, use the corresponding utility versus bandwidth function to determine the optimal bandwidth, as in Section 3.2.1.
2. Select the service which gives the highest surplus at its optimal bandwidth.

3.2.3 Simultaneous Adaptation of Multiple Applications corresponding to Single Task

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.

A number of researchers have noted the interaction between the perception of the different component media in a multimedia system, such as a video conference [21][22][24][26]. For example, an investigation of video phone systems indicated that any increase in visual representation of the speaker increases the viewer's tolerance to audio noise [21]. To take into account the interdependencies among applications, the utility of the i_{th} application should, in general, be written as $U^i(x^1, ..x^i, ..x^n)$, where x^i is the transmission parameter tuple for the i_{th} application. The total utility function of a system consisting of n individual application streams can be represented in general as $\hat{U}(U^1(\cdot), ..., U^n(\cdot))$, where $U^i(\cdot)$ represents the utility of stream i. 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 :

$$\hat{U} = \sum_i [U^i(x^1, ..., x^i, ..., x^n)] \tag{9}$$

and the optimization of surplus can be written as

$$\begin{aligned} max \sum_i [U^i(x^1, ..., x^i, ..., x^n) - C^i(x^i)] \\ \text{s. t.} \sum_i C^i(x^i) \leq b \\ x^i_{min} \leq x^i \leq x^i_{max} \end{aligned} \tag{10}$$

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.

The general approach to solving this problem is to represent each utility $U^i(\cdot)$ as a continuous function of the entire space of transmission parameters of all n applications, and solve the Kuhn Tucker equations so as to maximize the total surplus.

A simpler method is to use a similar approach as to that for the single application case, along with some heuristics. In this case, we make the simplifying assumption that the individual application utility functions can be defined independently and is a function only of the transmission parameters of that application - $U^i(\cdot) = U^i(x^i)$. This is a reasonable assumption since $U^i(\cdot)$ would normally depend strongly mainly on the vector x^i .

As earlier, we can decompose a single utility function $U^i(x^i)$ into a set of service-specific utility functions which are functions only of bandwidth, each corresponding to a particular delay and loss provided by a particular service. Clearly, several combinations of services (and hence, service-specific utility functions) are possible. We first consider one particular combination of service-specific utility functions. Let the utility of an application i be defined at L^i bandwidth levels. The utility at each level is u_l^i ($l = 1, 2, ..L^i$), and the utility function is piece-wise linear. Segment l (the straight line between levels l and $l + 1$) has a slope k_l^i. The optimal transmission parameter set for a particular combination of service-specific utility functions is then determined as follows:

1. From the utility function for each application i, determine the segment end-point $l_{opt}(l = 1, 2, ..L^i)$, with bandwidth B_{opt}^i, at which the surplus (utility minus cost) is maximized for that application. Let the cost of the targeted bandwidth be $C_{opt}^i(B_{opt}^i)$.

2. If the total expenditure needed for the system, $\sum_i C_{opt}^i(B_{opt}^i)$, exceeds the total system budget, go to step 3, else stop.

3. From all the applications that receive service at level $l_{opt} > l_{min}$, find the application i_{victim} with the smallest slope in the surplus ($u_l^i - C_l^i$) from level l_{opt} to $l_{opt} - 1$ (this corresponds to the smallest sensitivity of application surplus to a reduction in bandwidth). Reduce the current bandwidth allocation for this application to the next lower bandwidth level ($l_{opt} = l_{opt} - 1$).

4. If the total system expenditure remains greater than the system budget, go back to step 3. If there is excess budget, allocate the excess budget to the current victim application (from step 3) to acquire as much bandwidth as permitted by the budget.

The above algorithm is repeated for each possible combination of service-specific utility functions; each time, an optimal transmission parameter set is obtained.

4 System Stability and Network Dynamics

Applications will re-negotiate network services when a price quoted by the network changes or when the media traffic format changes, resulting in different bandwidth requirements. In addition, a new application entering the network or an exisiting application leaving the network will also lead to resource re-allocation. In this section, we first consider the stability of our pricing algorithm, and then the stability of the corresponding rate adaptaion process.

4.1 Price Stability

In our proposed pricing strategy of section 2.2, three price components are considered: holding price (p_h), usage price (p_u), and congestion price (p_c). For a specific network provider, the holding price (p_h), and usage price (p_u) for a particular service are fixed, or change infrequently. Hence, only the stability of the congestion price needs to be considered.

The adaptation of the proposed congestion price follows the *tâtonnement* process for an equilibrium. The price will be quoted upward or downward, depending on whether or not demand exceeds supply, until the demand and supply reach equilibrium and a stable price p^e is located.

Since demand is a function of price, we can denote demand as $D(p)$. For a network service class, the targeted resource supply is fixed and is denoted as S. Suppose the rate of change of price moves directly with excess demand, $E(p) = D(p) - S$ as follows:

$$p' = \frac{dp}{dt} = f(D(p) - S) = f(E(p)) \tag{11}$$

where $f' \geq 0$. The price change drives the demand and supply towards equlibrium. If the *tâtonnement* process is successful, the mechanism in equation 11 will generate a path of prices which will approach p^e as t increases:

$$\lim_{t \to \infty} p(t) = p^e \tag{12}$$

If equation 11 holds for any initial price p and p^e is unique, the system is called *globally* stable. If there is more than one equilibrium-price vector, then if $p(t)$ reaches any of the p^e's, the model is called *locally stable*. We only consider local stability in our system, where equation 12 holds for all prices p in some *neighborhood* of p^e. To prove that the local price stability exists, the function $f(E(p))$ can be represented by a Taylor series expansion:

$$\frac{dp}{dt} = f(E(p^e)) + f'E'(p^e)(p - p^e) + \cdots \tag{13}$$

The higher order terms are negligible in comparison with the first-order term in equation 13, as long as only *local* stability is considered. Since $E(p^e) = 0$ by the definition of price, the equation 13 can be written as:

$$\frac{dp}{dt} = f'E'(p - p^e) \tag{14}$$

The solution of this equation is:

$$p(t) = p^e + (p^0 - p^e)e^{(f'E')t} \tag{15}$$

where p^0 is any initial price.

The assertion of stability requires that the exponential term in equation 15 approaches zero as $t \to \infty$. Since $f' > 0$, so the stability assertion requires

$$E' = D_p(p) < 0 \tag{16}$$

In a resonable network system, user demand will decrease as the price increases, so $D_p(p) < 0$. This proves that the proposed price will reach stability as times increases. However, the convergence speed of the system will depend on the convergence rate parameter $\sigma(D, S)$, or $\sigma(p)$. In our experiments, in order to obtain rapid but smooth convergence, $\sigma(D, S)$ is large when the demand is much higher than supply, and is gradually reduced as the demand approaches supply.

Since the user demand will change as users join and leave, a new stable price may be reached as the total user demand changes. In the above process, the total demand and supply are assumed to be known instaneously. For a network with delay, this assumption may not be true. Since in our proposed model, the price is only updated periodically and in the time unit of minutes, the network delay has neglible influence on the stability.

4.2 Stability of user bandwidth requests

Even though the network can reach stability for any fixed set of bandwidth requirements, the stability can be disturbed when new applications enter the network and existing applications leave the network. In addition, bandwidth adaptation by a number of users sharing the same link bandwidth can also lead to the oscillation of the system price and user requests, before the demand and supply reach equilibrium.

In the core network, oscillatory behavior can be minimized by aggregating RNAP requests, reducing the frequency with which the RNAP agent re-allocates resources and adjusts the price **??**. The resource negotiated will be incremented or decremented with some minimum granularity. When the sum of per-flow requests approaches the resources reserved for the aggregate (or earlier, at some pre-defined threshold),

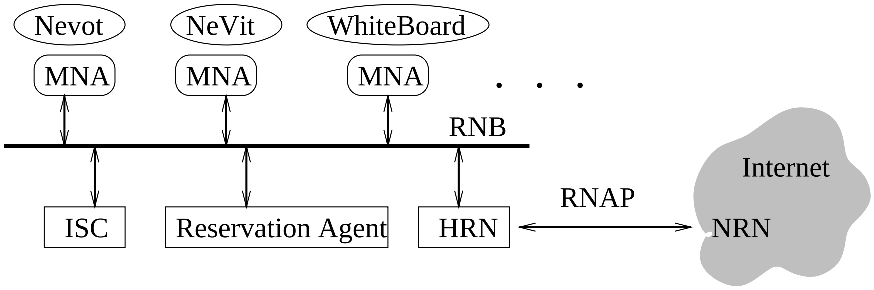

Figure 3: The architecture of the extended MINT system

the client negotiator will reserve an additional block of resources. Similarly, the requested reservation is decremented in blocks as required as the requested bandwidth decreases. The larger the block, the less frequently the aggregate session needs to be re-negotiated, but a higher holding cost is incurred for resources which may be under-utilized.

As the network price changes, users will renegotiate resources to optimize their perceived benefit (surplus) from the service. The total user requests are hence oscillatory. The piece-wise linear utility function used to simplify the optimization can sometimes result in a severe oscillation between two adaptation points far apart in bandwidth, as will be seen in the experiments in Section 6.2.1. In our experiments, we used two measures to damp out the oscillations. The first measure was to use a proportional plus derivative (PD) controller. During each negotiation period, instead of letting the requirement jump to a new optimal bandwidth, the user shifts to a bandwidth between the current one and the optimal one, resulting in temporarily sub-optimal operation. The PD control law regulates the bandwidth request as follows:

$$r_{i+1} = \begin{cases} r_i - \alpha_0(r_i - r^*) - \alpha_1(r_i - r_{i-1}), & \text{if } \frac{|SP(r^*) - SP(r_i)|}{SP(r_i)} > \theta \\ r_i, & \text{otherwise} \end{cases} \tag{17}$$

where r^* is the desired optimal rate, r_i is the rate requested for negotiation period i, and $SP(x)$ represents the surplus corresponding to bandwidth x. Quicker convergence is attained by making α_0 large, while the overshoot is minimized by making α_1 large.

In addition to the PD control, the bandwidth was allowed to be adjusted only if the new bandwidth led to an increase in surplus of at least θ %. This prevented bandwidth adaptation which did not result in a significant improvement in the perceived surplus.

5 Dynamic 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) [16] 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. 3.

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) [17]. 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, as described in section 2. 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.

6 Experimental Results and Analysis

In this section, we describe preliminary experimental results demonstrating some of the important features of our work, using a simplified implementation of RNAP. The implementation was based on an extension of the RSVP signaling protocol, and carried out on a test-bed consisting of two nodes 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 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

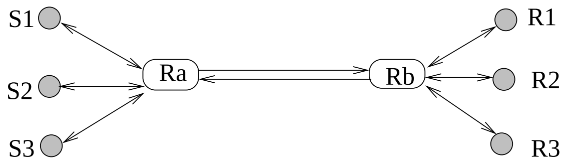

Figure 4: Architecture of test-bed used for the experiments

user utility functions on user adaptation and resource allocation.

6.1 Experimental Setup and Parameters

The test setup consisted of 2 routers (Ra and Rb) connected by a 10 Mb/s link, schematically represented in Fig. 4. 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 Controlled-Load (CL) [3] service, and the remaining bandwidth was configured for best effort service. Detailed descriptions regarding the price formulation in the network and network state maintainance are given in [14].

During the experiments, each of three application HRNs (belonging to three different users at S1, S2, and S3) requested the CL service over the link, determining its bandwidth reservation so as to optimize its own utility (perceived value). Since these experiments are intended as a preliminary demonstration of the working of the overall adaptation framework, we made the simplifying assumption that application utility is a function only of bandwidth, and other QoS parameters are relatively static or can be compensated for by adjusting the bandwidth requirement (e.g., reserving more bandwidth will reduce delay and loss).

We assumed a service roughly as expensive (per unit bandwidth) as a telephone line. Assuming a charge of 10 c/min, and 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.

While a 4 Mb/s partition of the link was set aside for CL service, the congestion threshold was set to 70% of the capacity (2.8 Mb/s). When the total reserved bandwidth on the link exceeded this threshold, the network began to apply the usage-sensitive congestion price to drive the demand down.

For each experiment, 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).

6.2 Experimental Results

We now describe a set of experiments which address the following issues: (i) the sharing of bandwidth between competing adaptive applications with identical utility functions; (ii) the sharing of bandwidth between competing applications with utility functions reflecting different amounts of elasticity in bandwidth requirements; (iii) distribution of bandwidth among applications belonging to a single-user multimedia system so as to maximize mission-wide value; (iv) the influence of specific changes in the utility function on the bandwidth adaptation; (v) adaptive behavior of audio and video applications belonging to the MINT system.

In each experiment, we study 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). We ascertain that a stable and equitable distribution of bandwidth is reached in each case.

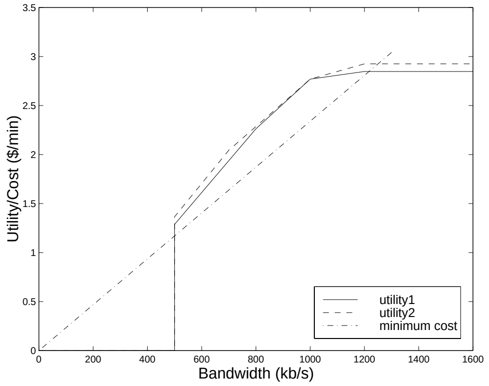

Figure 5: Utility functions used in the experiments of section 6.2.1 and 6.2.3

6.2.1 Bandwidth Sharing between Users

In the first experiment, we study the adaptive behavior when applications having the same utility function and belonging to different users compete for network resources. The same experiment is performed with two different utility functions, Utility1 and Utility2, shown in Fig. 5.

Fig. 6-a1, 6-a2, and 6-a3 show different aspects of adaptive behavior when Utility1 is used. Initially, in response to the initial price, each user determines that the optimal bandwidth (giving the maximum surplus) is 1000 kb/s. Since the total reservation of 3000 kb/s made by the three users is higher than the congestion threshold of 2800 kb/s, the network imposes an additional congestion price, resulting in a gradual increase in the price.

Fig. 6-a1 shows the initial increase in price, from 3.9 cents/Mb, until it stabilizes at 4.2 cents/Mb after about 150 seconds (corresponding to 5 negotiation periods). Fig. 6-a1 also shows the variation with time of the total bandwidth reservation, and Fig. 6-a2 shows the variation with time of the individual bandwidth reservations, and the maximum per-user bandwidth that the user budget permits. As the price increases, each user is constrained by its budget to decrease its sending rate in response. As a result, the reserved bandwidth decreases smoothly, until the link becomes un-congested, and the price stabilizes. Fig. 6 a3 shows a gradual decrease in the surplus obtained by each user until the price stabilizes. All users are observed to have nearly identical adaptation traces.

The second experiment uses Utility2 in Fig. 5. Utility2 differs from Utility1 in that the optimal bandwidth (at the initial un-congested link price) of 1000 kb/s differs only slightly from the next sub-optimal bandwidth of 700 kb/s with respect to the perceived surplus.

[Show results without PD control in 6 b, and with PD control in 6c?]

In Fig. 6 b, the adaptation traces are observed to be different from that shown in Fig. 6 a. When the price increases, the applications are constrained by their budget to reduce their bandwidths initially. When the price increases to a certain value, the optimal bandwidth requirement for all the users (calculated at slightly different times) shifts to 700 kb/s, since the increase in cost for a larger bandwidth is higher than for a smaller bandwidth. Since the two optimal points in our example are very far apart in bandwidth, and the perceived surplus of the two bandwidth are very close, an oscillation between 2100 kb/s and 3000 kb/s was observed in the total bandwidth when this experiment was performed.

To avoid this problem, the control scheme proposed in Section 4.2 was used to reduce the oscillation. In the experiment, α_0, α_1 and θ from equation 4.2 were set separately as 0.4, 0.6, and 2%. Fig. 6 c-2 shows that using the control scheme resulted in the bandwidth requirement of all three users stabilizing within seven negotiation periods. The users obtained unequal bandwidth shares inspite of having the same utility function. This is partly because of the asynchronous user negotiation behavior, and partly because

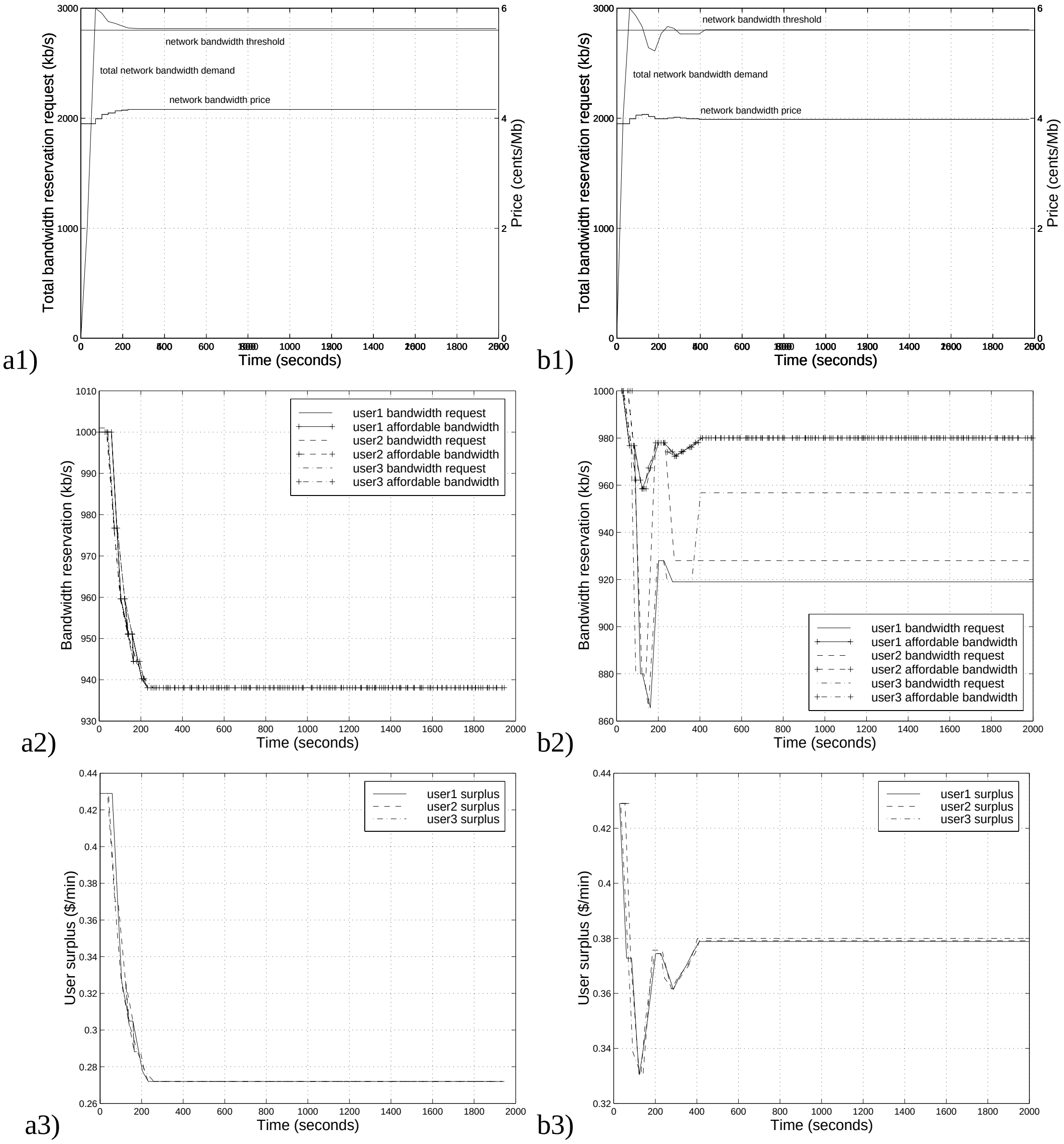

Figure 6: Allocation of bandwidth and surplus for three competing users sharing a link. a1, a2, and a3 show the results when the users all have the Utility 1 function from Fig. 5, and b1, b2, and b2 show corresponding results when the users have the Utility 2 function from the same figure

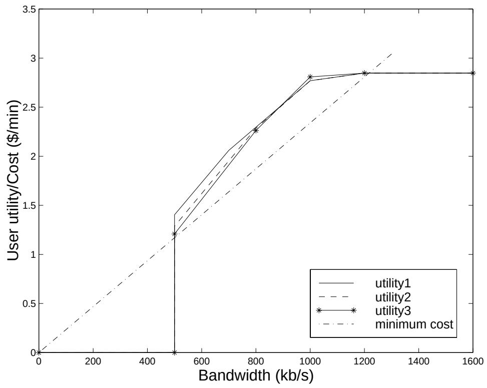

Figure 7: Utility functions with different bandwidth sensitivity

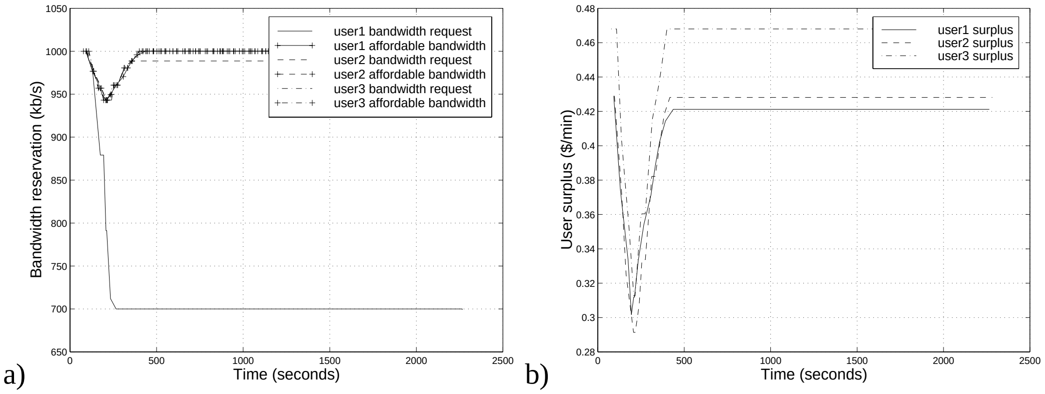

Figure 8: Bandwidth reservation a) and perceived surplus b) when all users have different demand elasticity

of the possible sub-optimal bandwidth request (within θ % of optimal) resulting from the control scheme of Section 4.2. All three users end up with final surplus values very close to each other (within 2 %). This is important since we consider the perceived surplus, rather than the bandwidth, as a measure of the user satisfaction.

6.2.2 Bandwidth Sensitivity and Demand Elasticity

In this experiment, we study the effect of different elasticities in user demand on user bandwidth sharing and adaptation, using different utility functions (Fig. 7) for different users. An utility function with a smaller slope reflects a higher elasticity in the bandwidth requirement of the user. Fig. 8-a shows that the user with the more elastic requirement is more sensitive to price changes and reduces his resource requirement faster when the network price increases. Correspondingly, Fig. 9-a shows that as a reward for elastic behavior, the average network charge for the more elastic user is lower, while the three users have similar perceived surplus (Fig. 8-b).

Thus, users with less stringent bandwidth requirements express this flexibility through a less bandwidth-sensitive utility function, and bear a greater share of reductions in bandwidth for congestion-control. Users with more bandwidth-sensitive requirements have to pay a higher charge during congestion to maintain their bandwidths at current levels.

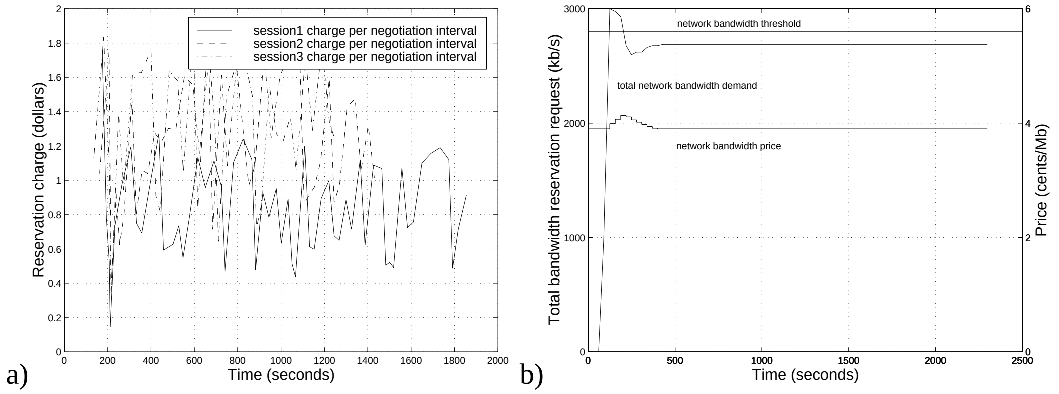

Figure 9: Network charges for different users a) and the total network bandwidth demand and price b) when the users have different demand elasticity

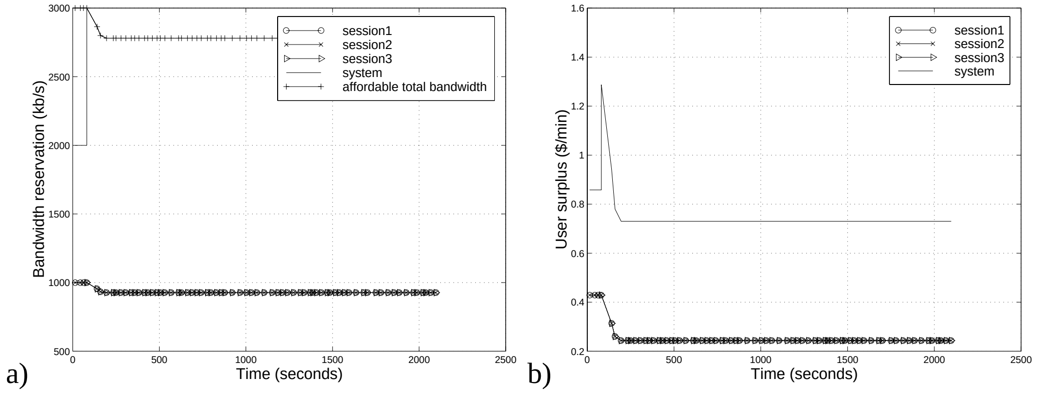

Figure 10: Bandwidth reservation a) and perceived surplus value b) for adaptation across media sessions in a system, all sessions having the same utility

6.2.3 Adaptation Across Media

The experiments so far show how a utility function serves to guide bandwidth adaptation by an application in response to congestion-sensitive pricing in the network, and how bandwidth is shared by competing applications based on individual demand elasticities. In this section, we look at how utility functions guide the distribution of bandwidth across different media which are part of a multimedia system belonging to a single user. The results of two experiments are presented.

In the first experiment, the system consists of three media sessions, all of which have the same utility function, Utility1 shown in Fig. 5. When the system budget is exceeded due to congestion, the HRN adjusts the application bandwidths downwards according to the adaptation algorithm described in Section 3.2.3. Since all the applications have identical utilities, the total system bandwidth is equally distributed between them at all times, as seen in Fig. 10 a).

The second experiment is similar except that the three media sessions have different utility functions shown in Fig. 7. Fig. 11 a) shows that when the total optimal bandwidth requirement for all the media sessions in the system exceeds the system budget, the media session with the more elastic resource demand will be assigned relatively less bandwidth so as to maximize the overall perceived value. This is a similar result to that obtained in section 6.2.2 for multiple competing user applications. In effect, the system regards a media session with more elastic requirements as being more able to absorb bandwidth reductions, and “borrows” bandwidth from this session to give to other sessions.

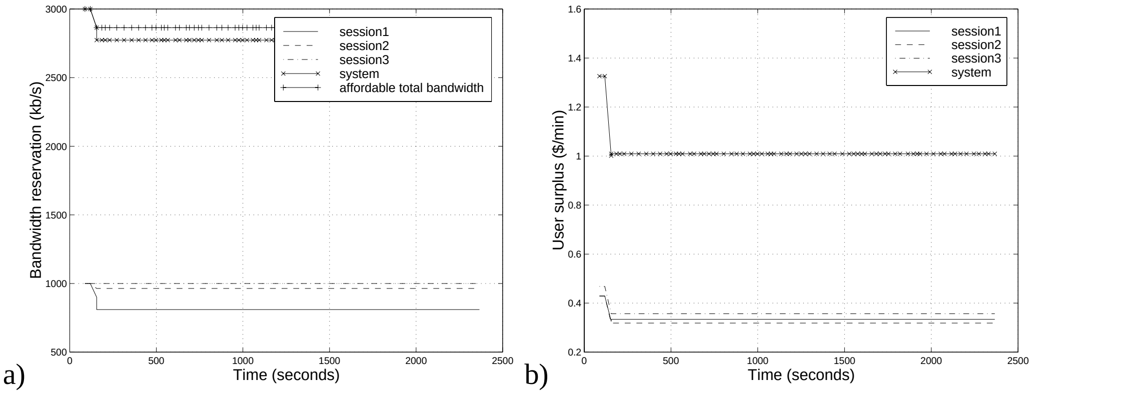

Figure 11: Resource reservation a) and perceived surplus value b) among sessions of a system with different bandwidth sensitivity

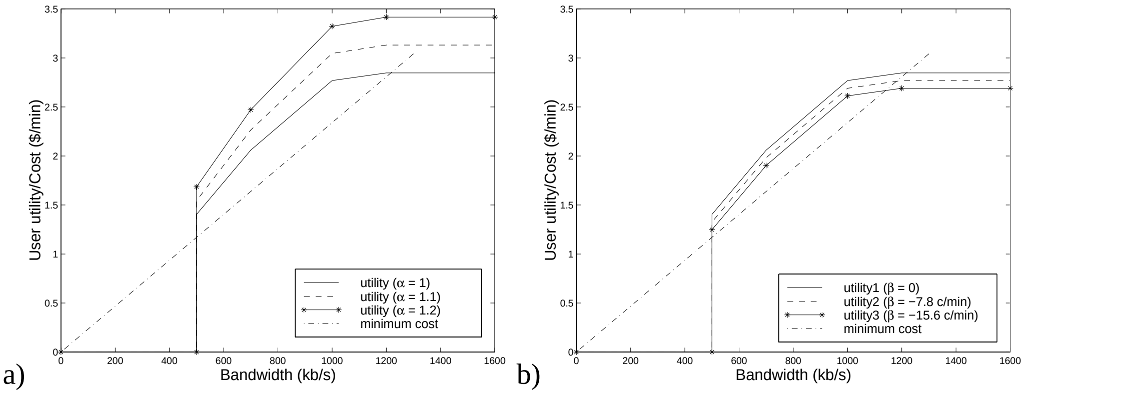

Figure 12: Equivalent utilities under multiplicative scaling a) and additive shifting b)

6.2.4 Linear Operations on Utility Functions

In this section, we show how linear operations on the utility function, namely multiplicative scaling by a weight α, and additive or subtractive shifting by an amount β, influence bandwidth selection and distribution. The experiment studies bandwidth distribution between multiple sessions in a system belonging to a single user, though similar results have also been observed with applications belonging to different users.

Consider three media sessions belonging to a system, all with the same basic (un-scaled) utility function (we use utility1 of Fig. 7). Sessions 1, 2, and 3 are assigned scaling factors of 1, 1.1, and 1.2 respectively. The resulting scaled utilities are shown in Fig. 12 a).

Fig. 13 shows the variation of individual and system bandwidth allocations and perceived surpluses. Expectedly, when the adaptation is constrained by the system budget, an application with a higher α gets a larger bandwidth share because of its lower elasticity of demand.

We now consider the effect of an offset applied uniformly to the utility over all bandwidths, so that the utility function shifts upwards or downwards. In Fig. 14 b), the utility1 function (which is the same as utility1 in Fig. 7 a) is shifted downwards and form utility2 and utility3. Three different sessions are assigned different utility functions.

The results shown on Fig. 14 a) shows that all three sessions are allocated the same bandwidth though Fig. 14 b) shows that the allocation results in different values of perceived surplus. This is because utility function represents the relative preference of the user for different bandwidths. The absolute value of the utility is not important - the adaptation algorithm only searches for the bandwidth with the maximum perceived value relative to its cost.

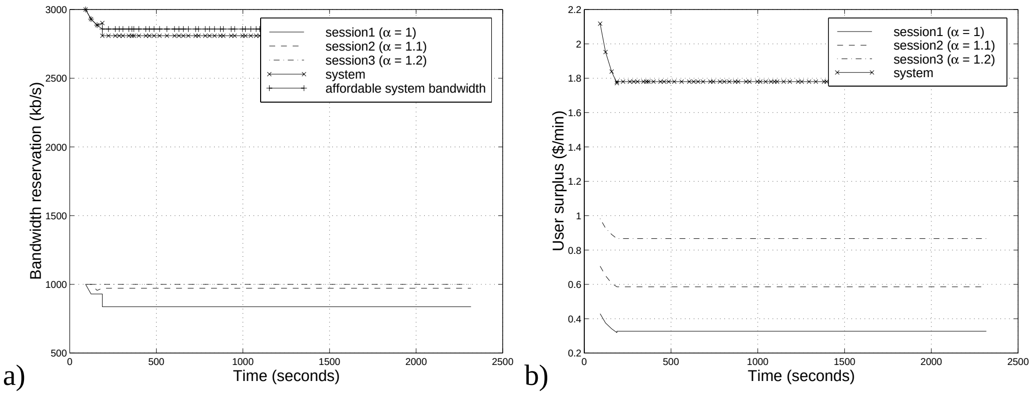

Figure 13: Bandwidth reservation and perceived surplus for utilities scaled multiplicatively by different amounts

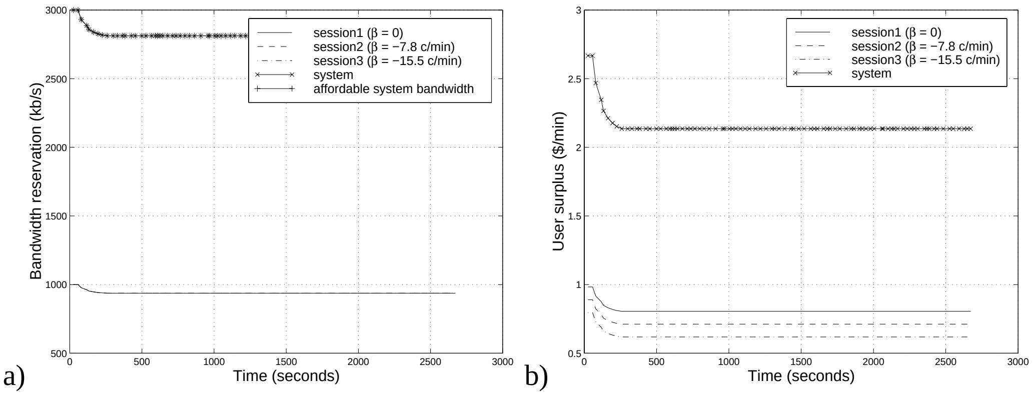

Figure 14: Bandwidth reservation and perceived surplus for utilities shifted additively by different amounts

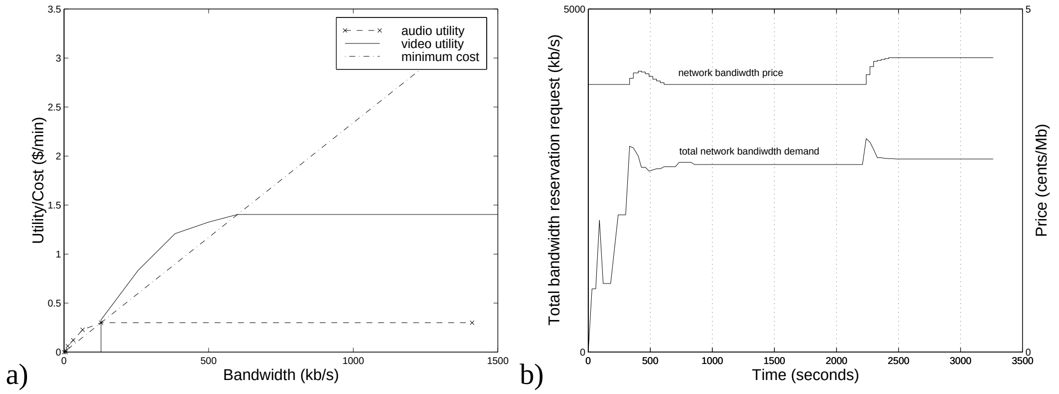

Figure 15: a) Audio and video utility functions used for adaptation by MINT b) Price and total bandwidth variation in the same experiment

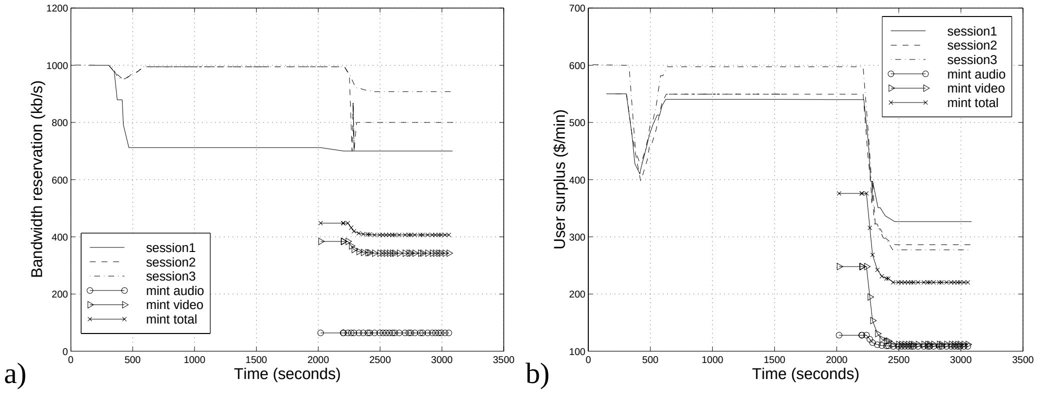

Figure 16: Individual bandwidth reservations and perceived surplus in the adaptation of Mint applications

6.2.5 Adaptation in MINT

Finally, we examine the adaptive behavior of the audio (NeVoT) and video (NeViT) applications in the MINT video conference system. The utility functions for the audio and video applications are shown in Fig. 15 a).

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. The MINT applications compete for bandwidth with three single media applications belonging to different users. The applications use the utility functions of Fig. 7. The three user applications are started first, and reach stability at time 630 seconds with bandwidth allocations of 712 kb/s, 994 kb/s, and 994 kb/s respectively.

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 total requested bandwidth exceeds the link congestion threshold, forcing the price up. It is observed the NeVoT bandwidth remains unchanged, 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.

7 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.

7.1 Resource Reservation and Allocation

Current research in providing QoS support in the Internet is mainly based on two architectures defined by IETF: Per-flow based *integrated services* (int-serv) [8], 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. Per-flow resource reservation in int-serv is generally implemented through the RSVP reservation protocol [1]. Implementation of resource reservation for diff-serv is a subject of ongoing research, and various approaches have been proposed [32]. In general, RSVP and the implementations of diff-serv lack integrated mechanisms by which the user can select one out of a spectrum of services, and re-negotiate resource reservations dynamically. They also do not integrate the pricing and billing mechanisms which must accompany such services.

Resource allocation schemes based on perceived-quality have been studied in [36][37][47]. These studies were limited to a local system, and did not address the interaction of the local system with a large network. Liao [48] allocates resources to achieve equal perceived quality. In section 2, we argued that perceived quality does not directly represent the economic value of communications.

7.2 Bandwidth Adaptation

	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 1: Comparison of algorithms for adjusting bandwidth in response to congestion

In this section, we categorize approaches towards bandwidth adaptation in reponse to congestion, as summarized in Table 1. The first row of Table 1 shows approaches that rely on reservation, and the second row shows approaches that do not. The clumns 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 [2] 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 reservations 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 [13], 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.

In earlier work, we described a Resource Negotiation and Pricing Protocol [14]. 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 the RNAP-based adaptation process.

7.3 Pricing and Billing in the Network

Microeconomic principles has been applied to various network traffic management problems. The studies in [34][36][38][39][42] 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 and truthful revelation by users of their utility curves, which may not be practical in a centralized process.

In [33][35][40][41][45], the resources are priced to reflect demand and supply. The pricing model in these approaches is usage-sensitive - it has been shown that usage-sensitive pricing results in higher utilization than traditional flat (single) pricing [33]. Some of these methods are limited by their reliance on a well-defined statistical model of source traffic, and are generally not intended to adapt to changing traffic demands.

The scheme presented in [41] is more similar to our work in that it takes into account the network dynamics (session join or leave) and source traffic characteristics (VBR). It also allows different equilibrium price over a different time period, depending on the different user resource demand. However, congestion is only considered during admission control. Our pricing algorithm has two congestion-dependent components - congestion due to excessive resource reservation (holding cost) and congestion due to network usage (usage cost).

In general, the work cited above differs from ours in that it does not enter into detail about the negotiation process and the network architecture, and mechanisms for collecting and communicating locally computed prices. Our work is more concerned with developing a flexible and general framework for resource negotiation and pricing and billing, decoupled from specific network service protocols and pricing and resource allocation algorithms. Our work can therefore be regarded as complementary with some of the cited work.

In [46], a charging and payment scheme for RSVP-based QoS reservations is described. A significant difference from our work is the absence of an explicit price quotation mechanism - instead, the user accepts or rejects the estimated charge for a reservation request. Also, the scheme is coupled to a particular service environment (int-serv), whereas our goal is to develop a more flexible negotiation protocol usable with different service models.

8 Summary

We have presented a framework for incentive-driven rate and QoS adaptation by an application or multi-application system. In this framework, the user responds actively to changes in price signaled by the network by dynamically adjusting network resource usage by the application. The adaptation is based on the user-perceived value of a given combination of transmission parameters, relative to the cost of obtaining the corresponding service from the network, taking into account constraints imposed by the minimum application requirements and the budget specified by the user. In a multi-application system such as a video-conference application, the system budget is distributed among the component media according to changes in price, as well as changes in the relative utilities with time or under different application scenarios, so as to maximize the overall perceived value relative to cost. Some heuristics are discussed to simplify this process.

Experimental results show that perceived value based adaptation allows bandwidth to be shared among competing users or applications in a system fairly. At the onset of congestion, the bandwidth share of users with more elastic demands is reduced more, but all users receive equitable levels of perceived surplus. We have discussed the stability of the dynamic pricing algorithm and bandwidth adaptation, and a PD control law is shown to minimize oscillations and abrupt transitions in the bandwidth adaptation. Multiplicative scaling and additive shifting of utility functions can be used to control the evolution of application utilities with time.

References

[1] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource Reservation Protocol (RSVP) - version 1 Functional Specification”, RFC 2205, Set. 1997.

[2] S. Shenker, and L. Breslau, “Two issues in reservation establishment”, *Proc. ACM SIGCOMM’95*, Cambridge, MA, August 1995.

[3] J. Wroclawski, “Specification of the controlled-load network element service,” RFC 2211, Sept. 1997.

[4] S. Shenker, C. Partridge, and R. Guerin, “Specification of guaranteed quality of service,” RFC 2212, Sept. 1997.

[5] P. Pan and H. Schulzrinne, “YESSIR: A simple reservation mechanism for the Internet”, In *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’98)*, Cambridge, England, July 1998.

[6] S. Jamin, S. J. Shenker, and P. B. Danzig, “Comparison of measurement-based admission control algorithms for controlled-Load service,”, *Proc. IEEE INFOCOM’97*, April 1997.

[7] H. Zhang and S. Keshav, “Comparison of rate-based service disciplines”, *Proc. ACM SIGCOMM’91*, Zurich, Switzerland, Sept. 1991.

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

[9] K. Nichols, V. Jacobson, and L. Zhang, “A two-bit differentiated services architecture for the Internet,” Internet Engineering Task Force, Nov. 1997. Work in progress.

[10] V. Jacobson, K. Nichols, and K. Poduri, “An expedited forwarding PHB,” Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[11] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured forwarding PHB group,” Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[12] X. Wang and H. Schulzrinne, “Comparison of adaptive Internet multimedia applications,” To appear at *IEICE Transactions on Communications.*, June, 1999.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” RFC 1889, e, Jan. 1996.

[14] X. Wang and H. Schulzrinne, “RNAP: A Resource Negotiation and Pricing Protocol”, to appear at *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’99)*.

[15] R. Velthuys, J. Wong, K. Lyons, G. v. Bochmann, E. Dubois, N. Georganas, G. Neufeld, and T. Ozsu, “Enabling technologies for distributed multimedia applications”, CITR Internal Report, 1995.

[16] D. Sisalem and H. Schulzrinne, “The multimedia Internet terminal (MINT),”*Journal of Telecommunications*, vol. 9, pp. 423-444, 1998.

[17] Henning Schulzrinne, “Dynamic configuration of conferencing applications using pattern-matching multicast,” *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’95)*, 1995.

[18] H. Varian, “Microeconomic Analysis,” Third Edition.

[19] ITU-T P.800, “Methods for subjective determination of transmission quality”.

[20] H. Knoche, H. De Meer, D. Kirsh, “Utility curves: Mean Opinion Scores considered biased,” To appear at *IWQoS’99*, London, June, 1999.

[21] O. Ostberg, B. Lindstrom, and P-O., Renhall, “Contribution of display size to speech intelligibility in video-phone systems”, *International Journal of Human-Computer Interaction*, 1(1), pp 149-159, 1989.

[22] A. H., Anderson, E. G. Bard, C. Sotillo, A. Newlands, G. Doherty-Sneddon, “Limited visual control of the intelligibility of speech in face-to-face dialogues,” in *Perception and Psychophysics*, 59(4), 580-592., 1997.

[23] C. Lambrecht and O. Verscheure, “Perceptual quality measure using a spatio-temporal model of human visual system”, *Proc. of IS&T/SPIE*, San Jose, Feb, 1996.

[24] A. Watson and M. A. Sasse, “Evaluating audio and video quality in low-cost multimedia conferencing systems,” *Interacting with Computers*, Vol. 8 (3), pp. 255-275, 1996.

[25] P. Usai and C. Grilli, “Subjective testing methodology to assess price vs. transmission quality of telecommunication services,” in *Proc. of 15th International Symposium on Human Factors In Telecommunications*, (Melbourne, Australia), Mar. 1995.

[26] E. A. Isaacs and J. C. Tang, “What video can and cannot do for collaboration: A case study,” *Multimedia Systems*, vol. 2, pp. 63-73, 1994.

[27] M. Podolsky, C. Romer, and S. McCanne, “Simulation of FEC-Based Error Control for Packet Audio on the Internet,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, California), pp. 505, March/April 1998.

[28] R. Vaccaro, “Digital control, a state space approach”, McGraw Hill, New York, 1995

[29] J. Padhyem, J. Kurose, D. Towsley and R. Koodli, “A TCP-Friendly Rate Adjustment Protocol for Continuous Media Flows over Best Effort Networks” , *UMass-CMPSCI Technical Report* TR 98-04, October 1998.

[30] S. Floyd, and K. Fall, “Promoting the Use of End-to-End Congestion Control in the Internet”. Submitted for publication, Feb. 1998.

[31] D. Lin and R. Morris, “ Dynamics of random early detection” . *Proc. SIGCOMM’97,*

[32] Internet 2 Bandwidth Broker Information, http://www.merit.edu/working.groups/i2-qbone-bb.

[33] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang, “Pricing in computer networks: Motivation, formulation, and example,” *IEEE/ACM Transactions on Networking,* vol. 1, pp 614-27, Dec. 1993.

[34] J. F. MacKie-Mason and H. Varian, “Pricing Congestible Network Resources,” *IEEE J. Select. Areas Commun.,*, vol. 13, no. 7, pp 1141-9, Sept. 1995.

[35] N. Anerousis and A. A. Lazar, “A framework for pricing virtual circuit and virtual path services in atm networks”, *ITC-15,* pp. 791 - 802, 1997.

[36] A. Hafid, G. V. Bochmann and B. Kerherve,“A quality of service negotiation procedure for distributed multimedia presentational applications,” *Proceedings of the Fifth IEEE International Symposium On High Performance Distributed Computing (HPDC-5),* Syracuse, New York, 1996.

[37] T. F. Abdelzaher, E. M. Atkins, and K. Shin, “QoS negotiation in real-time systems and its application to automated flight control,” To appear in *IEEE Transactions on Software Engineering,* 1999.

[38] H. Jiang and S. Jordan, “A pricing model for high speed networks with guaranteed quality of service,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom),* (San Fransisco, California), Mar. 1996.

[39] S. Low and P. Varaiya, “An algorithm for optimal service provisioning using resource pricing,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom),* (Toronto, Canada), June 1994.

[40] D. F. Ferguson, C. Nikolaou, and Y. Yemini, “An economy for flow control in computer networks,” in Proceedings of the *Conference on Computer Communications (IEEE Infocom),* (Ottawa, Canada), pp. 110-118, IEEE, Apr. 1989.

[41] E. W. Fulp, D. S. Reeves, “Distributed network flow control based on dynamic competive markets,” *Proceedings International Conference on Network Protocol (ICNP’98),* Austin Texas, Oct. 13-16, 1998.

[42] F. P. Kelly, A.K. Maulloo and D.K.H. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society* 49 (1998), 237-252.

[43] H. Varian, “Microeconomic Analysis,” Third Edition, 1993. W.W. Norton & pany.

[44] Hahn F (1982). Stability. In: Arrow KJ and Intriligator MD (eds), “handbook of Mathematical Economics”, Volumn II. Noth-Holland, Amesterdam, pp 745-793.

[45] J. Sairamesh, “Economic paradigms for information systems and networks”, PhD thesis, Columbia University, New York 1997.

[46] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz, “An embedded charging approach for RSVP,” *The Sixth International Workshop on Quality of Service (IWQoS’98),* pp 91-100, Napa, California, USA.

[47] C. Lee, J. Lehoczky, R. Rajkumar and D. Siewiorek, “On Quality of Service Optimization with Discrete QoS Options,” *Proceedings of the IEEE Real-time Technology and Applications Symposium,* June 1999.

[48] G. Bianchi, A.T. Campbell, and R.R.-F. Liao, “On utility-fair adaptive services in wireless networks, ” *6th International Workshop on Quality of Service (IEEE/IFIP IWQOS’98)* , Napa Valley, CA, May 1998.