Pricing Network Resources for Adaptive Applications in a Differentiated Services Network

Xin Wang and Henning Schulzrinne
Columbia University
{xinwang, schulzrinne}@cs.columbia.edu

Abstract— **The Differentiated Services framework (DiffServ) has been proposed to provide multiple Quality of Service (QoS) classes over IP networks. A network supporting multiple classes of service also requires a differentiated pricing structure. In this work, we propose a pricing algorithm in a DiffServ environment based on the cost of providing different levels of quality of service to different classes, and on long-term average user resource demand of a service class. We also integrate the proposed service-dependent pricing scheme with a dynamic pricing and service negotiation environment by considering a dynamic and congestion-sensitive pricing component. Pricing network services dynamically based on the level of service, usage, and congestion allows a more competitive price to be offered, allows the network to be used more efficiently, and provides a natural and equitable incentive for applications to adapt their service requests according to network conditions. We develop a simulation framework to compare the performance of a network supporting congestion-sensitive pricing and adaptive service negotiation to that of a network with a static pricing policy. Adaptive users adapt to price changes by adjusting their sending rate or selecting a different service class. We also develop the demand behavior of adaptive users based on a physically reasonable user utility function. Simulation results show that a congestion-sensitive pricing policy coupled with user rate adaptation is able to control congestion and allows a service class to meet its performance assurances under large or bursty offered loads, even without explicit admission control. Users are able to maintain a stable expenditure, and allowing users to migrate between service classes in response to price increases further stabilizes the individual service prices. When admission control is enforced, congestion-sensitive pricing still provides an advantage in terms of a much lower connection blocking rate at high loads.**

I. Introduction

The Differentiated Services framework (DiffServ) [1] has been proposed to provide multiple Quality of Service (QoS) classes over IP networks. Two types of Per-Hop-Behavior (PHB) are proposed: Expedited Forwarding (EF) [2] and Assured Forwarding (AF) [3]. The EF PHB is defined as a forwarding treatment where the departure rate of an aggregate's packets from any DiffServ node must equal or exceed a configurable rate. For AF service, four classes with three levels of drop precedence in each class are defined for general use.

A network supporting multiple classes of service also requires a differentiated pricing structure to allocate traffic to different classes, rather than the flat-fee pricing model adopted by virtually all current Internet services. While network tariff structures are often dominated by service providers' policies and marketing arguments rather than costs, we believe it is worthwhile to understand and develop a cost-based pricing structure as a guide for actual pricing. In economically viable models, the difference in the charge between different service classes would presumably depend on the difference in performance between the classes, and should take into account the average (long-term) demand for each class. In general, the level of forwarding assurance of an IP packet in DiffServ depends on the amount of resources allocated to a class the packet belongs to, the current load of the class, and in case of congestion within the class, the drop precedence of the packet. Also, when multiple services are available at different prices, users should be able to demand particular services, signal the network to provision according to the requested quality, and generate accounting and billing records. One of the two main goals of our work is to develop a pricing scheme in a DiffServ environment based on the cost of providing different levels of quality of service to different classes, and on long-term demand. Instead of purely relying on either engineering or economic mechanism to manage the network, we integrate our pricing strategy with the network provisioning. Our pricing scheme reflects the differentiation in resource provisioning for different forwarding assurances and also serves as a means to distribute user traffic to different service classes. In our framework, the network can also enforce QoS and ensure the expected performance of a service class by restricting the load of the class to its target level through admission control.

DiffServ supports services which involve a traffic contract or service level agreement (SLA) between the user and the network. If the agreement, including price negotiation and resource allocation are set statically (before transmission), pricing, resource allocation and admission control policies (if any) have to be conservative to be able to meet QoS assurances in the presence of network traffic dynamics. Pricing network services dynamically based on the level of service, usage, and congestion allows a more competitive price to be offered, and allows the network to be used more efficiently. Differentiated and congestion-sensitive pricing also provides a natural and equitable incentive for applications to adapt their service contract according to network conditions. A number of adaptation schemes have been proposed for multimedia applications to dynamically regulate the source bandwidth according to the existing network conditions (a survey of this work is given in [4]).

The second main goal of our work is to integrate the proposed service-dependent pricing scheme with a dynamic pricing and service negotiation environment. This increases network resource utilization and avoids high call blocking rate. In this environment, service prices have a congestion-sensitive component in addition to the long-term, relatively static price. 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.

A user has a choice of adapting its service request only at session setup or periodically during a session. With both static and dynamic pricing components, our proposed model allows the

This work was sponsored by NSF CAREER grant.

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. Applications may choose services that provide a fixed price, and fixed service parameters during the duration of service. Generally, the long-term average charge for fixed-price service is higher since the network provider will add a risk premium. Users with stringent bandwidth and QoS requirements can maintain a high quality by paying more, while adaptation-capable applications can adjust their service requests in response to price increases during congestion. In this paper, we develop the demand behavior of adaptive users based on a physically reasonable user utility function.

In our simulations, prices and services are negotiated through a Resource Negotiation and Pricing (RNAP) protocol and architecture, presented in earlier work [5]. RNAP enables the user to select from available network services with different QoS properties and re-negotiate contracted services, and enables the network to dynamically formulate service prices and communicate current prices to the user. In RNAP, resource commitments can be made for short "negotiation" intervals, instead of indefinitely, and prices may vary for each interval. In the simulation work of this paper, we focus on the study of the performance of adaptive applications in a differentiated service environment, and assume all applications are adaptation-capable and will adapt their requests periodically if congestion-sensitive pricing is enforced. Our earlier studies [6] indicate that the system performance can be improved significantly even if only a small proportion of the users adapt. We also showed in [6] that if users adapt the service requests at session setup instead of every smaller time interval, the system performance reduces only slightly.

Using RNAP and an extended version of an existing DiffServ implementation, we develop a simulation framework to compare the performance of a network supporting congestion-sensitive pricing and adaptive service negotiation to that of a network with a static pricing policy. We also study the stability of the dynamic pricing and service negotiation mechanisms. We evaluate the system performance and perceived benefit (or value-for-money) under the dynamic and static systems. We also study the relative effects on system performance of rate adaptation, dynamic load balancing between service classes and admission control. Although the simulation framework is based on the RNAP model, we try to derive results and conclusions applicable to static and congestion-driven dynamic pricing schemes in general.

This paper is organized as follows. Section II develops a physically realistic user utility function to represent user demand behavior in response to price changes. Section III discusses our proposed pricing model in detail. Section IV summarizes our earlier work on RNAP and explains how it could support the proposed pricing algorithms in a DiffServ environment. In section V we describe our simulation model, and in section VI we discuss simulation results. We describe some related work in section VII, and summarize our work in section VIII.

II. User Application Adaptation

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

Although we focus on adaptive applications as the ones best suited to a dynamic pricing environment, our framework does not require adaptation capability. 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 efficiently. Alternatively, applications may use a service with usage-sensitive pricing, and maintain a high QoS level, paying a higher charge during congestion.

Adaptation can be performed at different time scales. 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 [7] 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, with 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 have not been studied extensively. A lot of recent research on multimedia adaptations is based on best-effort service, with signaling mechanisms such as packet loss rates for feedback [4]. For example, loss rates can be determined from RTP information [8], 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 this paper, we consider the adaptation on time scales of a minute or longer, i.e. once a session.

We consider a set of user applications, required to perform a task or *mission* (for example, a video conference, consisting of audio and video). The user would like to determine a set of transmission parameters (sending rate and QoS parameters) from which it can derive the maximum benefit towards completing the mission, subject to his budget. Consumers in the real world generally try to obtain the best possible "value" for the money they pay, subject to their budget constraints 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 a 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 charge a user has to pay for obtaining that service. The goal of the adaptation is to maximize this surplus, subject to the budget constraint and the minimum and maximum QoS requirements of the user.

We now consider the simultaneous adaptation of transmission parameters of a set of I 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. We can think of the adaptation process as the allocation and dynamic re-allocation of a finite amount of resources between the applications. For example, in order to maximize the overall value of a video conference, the total amount of resources need to be distributed between audio and video. The audio application may be given higher preference and weight, and when there is a scarcity of resources, the bandwidth and QoS parameters for video may be reduced.

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 of a task as the sum of individual application utilities:

$$\hat{U} = \sum_i [U^i(X^i(T_{spec}, R_{spec}))], \tag{1}$$

where X^i is the function of the transmission (T_{sepc}) and quality of service parameter (R_{spec}) tuple for the i_{th} application. The optimization of surplus can be written as

$$\max \sum_i [U^i(X^i) - Co^i(X^i)]$$
$$\text{s. t.} \sum_i Co^i(x^i) \leq b, \quad X^i_{min} \leq X^i \leq X^i_{max}, \tag{2}$$

where X^i_{min} and X^i_{max} represent the minimum and maximum transmission requirements for stream i, Co^i is the cost of the service selected for stream i at requested transmission parameter X^i, and b is the budget of the user over a time unit (e.g., per minute or per second).

In practice, the application utility is likely to be learnt and indicated by users at discrete bandwidths, at one or a few levels of loss and delay, possibly corresponding to a subset of the available services. At the current stage of research, some possible services are guaranteed [9] and controlled-load service [10] under the IntServ model, and Expedited Forwarding (EF) [2] and Assured Forwarding (AF) [3] under DiffServ. In this case, it is convenient to represent the utility as a piecewise linear function of bandwidth (or as a set of such functions, one for each level of loss and delay). A simplified algorithm is proposed in [11] to search for the optimal service requests in such a framework.

At a fixed value of loss and delay, we can make some general assumptions about the utility function as a function of the bandwidth (can be equivalent bandwidth [12]). A user application generally has a minimum requirement for the transmission bandwidth. It also associates a certain minimum value with a task, which may be regarded as an "opportunity" value, and this is the perceived utility when the application receives just the minimum required bandwidth. A user may decide to terminate the application if it can not obtain the minimum bandwidth, or when the price charged is higher than the opportunity value derived from keeping the connection alive. User experiments reported in the literature [13][14] suggest that utility functions typically follow a model of diminishing returns to scale, that is, the marginal utility as a function of bandwidth diminishes with increasing bandwidth. Also, as shown in [15], the optimal solution is proportionally fair[1] when all user utilities are logarithmic. As users' preferences are normally concave functions of bandwidth when the loss and delay are fixed, and also to track the opporunity value a user reserves for minimum transmission, a utility function can be represented as:

$$U(x) = U_0 + w \log \frac{x}{x_m}, \tag{3}$$

where $U(x)$ denotes the utility at a particular bandwidth x, 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 at the lowest bandwidth level ($x_{\min}$). Even though we assume a logarithmic form for the utility function in this paper, similar results are obtained with other concave forms [16].

The utility function is also sensitive to network transmission parameters such as loss and delay. In our work, we rely on the experimental results in [17] which show that users' perceived quality for interactive audio decreases almost linearly with either delay or loss, with a minimum acceptable quality requirement. More subjective tests are needed for other application types. Currently, we assume a similar linear dependence for all applications. Accordingly, we represent the utility function of an application as:

$$U(x) = U_0 + w \log \frac{x}{x_m} - k_d d - k_l l, \quad \text{for} \quad x \geq x_m, \tag{4}$$

where k_d and k_l represent respectively the user's sensitivity to delay and loss. In some cases, the user's perceived sensitivity may depend on the bandwidth used. For example, tolerance to delay and loss will be different for different speech codecs. Since we are not assuming any particular application model, we assume users' delay and loss sensitivity are bandwidth independent in our simulations. A user with a higher sensitivity to delay or loss will tend to select a higher service class rather than request more bandwidth. If the utilities of all the applications are represented in the format of equation 4, the optimization process for a user task with multiple applications can be represented as:

$$\max \sum_i [U^i_0 + w^i \log \frac{x^i}{x^i_m} - k^i_d d^i - k^i_l l^i - p^i x^i]$$
$$\text{s. t.} \sum_i p^i x^i \leq b, \quad x^i \geq x^i_m, \quad d^i \leq D^i, \quad l^i \leq L^i, \quad \forall i, \tag{5}$$

where p^i is the price (which is the summation of the congestion price and usage price to be described in Section III) of the service class selected by the application i , D^i and L^i are respectively the loss and delay bound of an application i, above which the application no longer functions usefully. Note that the loss l^i and delay d^i of a service class i can be obtained from the service announcement of the provider. The optimization of equation 5 will only be performed over the service classes that

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

can provide the level of loss and delay below the QoS bounds of a corresponding application.

The utility for each user application is a function of the bandwidth, delay, and loss. It is possible to represent the above optimization problem as a Lagrangian and find the optimal total user surplus (gained from all the applications) directly by optimizing over different delay, loss and bandwidth for different applications. However, to avoid the computation complexity, we assume the availability of only a few different loss and delay levels corresponding to different service classes and accordingly use a more heuristic method.

The optimization involves assigning a service class and a bandwidth to each application i. For each class assignment, the corresponding loss and delay an application will experience is known and determined by the class assigned to the application. The total user surplus associated with a class assignment can hence be obtained by optimizing over the bandwidth allocations only, and the bandwidth allocated for an application i can be represented as:

$$x^i = \frac{w^i}{p^i}, \tag{6}$$

where w^i represents the money a user would spend for an application i based on its perceived value and is independent of class assignment, and p^i is the service price of the class assigned to application i and will change when a different class is assigned to the application. The above bandwidth distribution is calculated for all possible service class assignments (constrained by application requirements and budget), and the one giving the highest total surplus is selected as the optimal solution.

If there are no service class assignments for which the optimal distribution of equation 6 can be obtained at a cost below the budget, the total budget is first distributed to the component applications according to their relative bandwidth sensitivity w^i. That is, each application receives a budget share b^i such that

$$b^i = b\frac{w^i}{\sum_k w^k}. \tag{7}$$

Each application is then allocated a service i and bandwidth $x^i = \frac{b^i}{p^i}$ which maximizes its individual surplus according to equation 4.

Note that a user's decision will not be impacted by other user's individual decision, although the total demand of all the users for a service class in the previous price update cycle may modify the congestion price (to be discussed in the next section) of the service class. In this section, we have shown how a user can adapt its request in response to price change. In reality, a user may be offered a constant and cheaper price (as compared to the ones that prefer constant service rate) for a service class, but will need to adapt its sending rate at the same ratio that the price would have been increased when congestion happens. This is supported by equation 6.

The discussion so far assumes that each price p^i is the cost per unit average bandwidth. A price based on unit equivalent bandwidth [18] may be fairer since it takes into account the burstiness of user traffic. In this case, the user adaptation of the source rate is more complicated. If effective bandwidth is used, a user could calculate a new average bandwidth when the price increases. Alternatively, it could introduce additional buffering at the source to reduce its burstiness, at the cost of a higher delay, thus reducing the effective bandwidth.

III. Pricing Strategies

A few pricing schemes are widely used in the Internet today [19]: access-rate-dependent charge (AC), volume-dependent charge (V), or the combination of the both (AC-V). An AC charging scheme is usually one of two types: allowing unlimited use, or allowing limited duration of connection, and charging a per-hour fee for additional connection time. Similarly, AC-V charging schemes normally allow some amount of volume to be transmitted for a fixed access fee, and then impose a per-volume charge. Although time-of-day dependent charging is commonly used in telephone networks, it is not generally used in the current Internet.

The current predominant form of Internet pricing in the United States is access-rate-dependent flat charge. A flat rate charge, although simple to implement, is not efficient and forces light users to subsidize heavy users as shown, for example, by the INDEX experiments at UC Berkeley [20]. Due to the lack of a proper pricing model, the providers cannot generate enough revenue from the network service to sustain the enormous investment required to provide the service. And service providers cannot support service differentiation and any QoS guarantee that is required by some of the new Internet applications, such as real-time multimedia applications. User experiments [21] indicate that usage-based pricing is a fair way to charge people and allocate network resources. Although there is a preference for a cheap best effort service, users want to be able to choose a high quality of service for important tasks and are willing to pay for it [22]. Both connection time and the transmitted volume reflect the usage of the network. Charging based on connect-time only works when resource demands per time unit are roughly uniform. Since this is not the case for Internet applications and across the range of access speeds, we only consider volume-based charging.

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. 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 classes ("flat", FP-FL), depend on the service class (FP-S), depend on the time of day (FP-T) or a combination of time-of-day and service class (FP-S-T).

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 at the call admission control level, or the packet delay and dropping rate will increase 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 consists of a congestion-dependent com-

ponent and a fixed volume-based charge. The charge in CPA has the same four charging modes as in FP, giving the pricing models CP-FL, CP-S, CP-T, CP-S-T.

The purpose of this paper is to develop pricing algorithms for efficient resource provisioning and service differentiation, and study their performance in an environment that enables network engineering control and with traffic dynamics. We therefore focus on the FP-S and CP-S models.

A. Proposed Pricing Scheme

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 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 the total demand 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*.

A.1 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 completely utilize 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^j) of a service class j is therefore set to reflect the difference between the usage price for that class and the usage price for the next lower service class and can be represented as:

$$p_h^j = p_u^j - p_u^{j-1}. \tag{8}$$

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

$$c_h^{ij}(n) = p_h^j(r^{ij}(n)\tau^j - v^{ij}(n)), \tag{9}$$

where τ^j is the length of negotiation interval for class j, $v^{ij}(n)$ is the traffic sent by user i over the period n, and $r^{ij}(n)\tau^j - v^{ij}(n)$ is the bandwidth not used by the user. $r^{ij}(n)$ can be a bandwidth requirement specified explicitly by the customer i, or estimated from the traffic specification and service request of the customer. Note that a user that does not use the resource reserved should not be charged more than the usage charge when his transmission consumes the same amount of resources. That is, the holding price should not be higher than the usage price.

A.2 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. 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 fixed costs associated with the service. In a network supporting multiple classes of service, the difference in the charge between different service classes would presumably depend on the difference in performance between the classes. The model we consider is a network supporting J classes of services, the service price for class j is p_u^j, the long time user bandwidth demand for class j is known (e.g., through statistics) and can be represented as x^j. The provider's decision problem is to choose the optimal prices for each class that optimize its profit:

$$\max_{p_u^j}[\sum_j^J x^j p_u^j - f(C)],$$
$$\text{subject to:} \quad \sum_j^J x^j \le C, \tag{10}$$

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

If we assume the users have the utility functions of Section II, the total demand of service class j can be represented as a constant elasticity model: $x^j = A^j/p_u^j$, which varies inversely with the price of the service class. A^j reflects the total willingness to pay of users belonging to service class j.

Service pricing for differentiated service

DiffServ supports SLA negotiation between the user and the network. An SLA generally includes traffic parameters, which describe the user's traffic profile, and performance parameters, which characterize the level of performance that the network promises to provide to the conforming part of the user's traffic. A widely used descriptor for a user's traffic profile consists of a peak rate, a sustainable rate, and a maximum burst tolerance. The generally considered QoS parameters are delay and loss. Mechanisms, such as weighted fair queuing (WFQ) [24][25][26] and class based queuing (CBQ) [27] can be used to provision resources for different service classes. In general, a class with lower load leads to lower delay expectation. A higher level of service class is expected to have a lower average load, and hence lower average delay. If we do not consider fixed costs due to provider's policies and business operations (which can be a fixed item amortized over time), charging services inversely proportional to their individual expected load seems to

reasonably reflect the cost of providing the services and the differences between their performance. Denoting the equilibrium unit bandwidth price at a node under full utilization by p_{basic}, and the expected utilization of service class j by ρ^j, the unit bandwidth price for service class j can then be estimated as $p_u^j = p_{basic}/\rho^j$. The effective bandwidth consumption of an application with rate x^{ij} can be represented as x^{ij}/ρ^j. For constant elasticity demand, $x^j = A^j/p_u^j$, the effective bandwidth consumption is $A^j/(p_u^j\rho^j)$. Then the price optimization problem of equation 10 can be written as

$$\max_{p_u^j}[\sum_j^J \frac{A^j}{p_u^j}p_u^j - f(C)], \quad p_u^j = \frac{p_{basic}}{\rho^j},$$
$$\text{subject to:} \quad \sum_j^J \frac{A^j}{p_u^j\rho_j} \le C. \tag{11}$$

The Lagrangian for the problem can be represented as

$$\max_{\lambda}[\sum_j^J A^j + \lambda(C - \frac{\sum_j^J A^j}{p_{basic}}) - f(C)]. \tag{12}$$

The optimal solution is:

$$p_{basic} = \frac{\sum_j^J A^j}{C}, \quad p_u^j = \frac{p_{basic}}{\rho^j} = \frac{\sum_j^J A^j}{C\rho^j}. \tag{13}$$

The bandwidth provisioned for a service class j will then be given by A^j/p_{basic}, and is proportional to total user willingness to pay for that class. Note that even though the objective function of equation 11 no longer depends on usage price if a constant demand function is assumed, the impact of usage price is reflected through the constraint function. The service selection of a user will be affected both by the prices and service quality of different classes. Small price difference may not lead users to migrate between classes. Once a user selects a service level, we assume the amount of resources the user requires only depends on the price of the selected class. Also, the usage price and resource provisioning of a class will be adjusted over longer time scales, depending on the long-term total average user demand of different classes. So the long-term average user demand will mainly depend on the price of the selected class. As we will see later, in short-term, individual users can be encouraged to migrate between classes due to traffic dynamics and congestion pricing.

The usage_charge $c_u^{ij}(n)$ for class j over a period n in which $v^{ij}(n)$ bytes were transmitted is given by

$$c_u^{ij}(n) = p_u^j v^{ij}(n). \tag{14}$$

Initially, the usage prices are set based on historic data. The prices can be varied according to the usage patterns observed over a long-term period.

A.3 Congestion Charge

A simple usage-based charging scheme monitors the data volume transmitted and charges users based on their average rate. Charging according to the mean rate, although encouraging the user to use network bandwidth more efficiently, does not discourage users from selecting loose traffic contracts and sending the worst-case traffic allowed by their contract, which create problems for network traffic management. An appropriate pricing scheme should provide users incentives to select traffic contracts that reflect their actual needs. Effective bandwidth [12][28] and pricing based on effective bandwidth [18] have been proposed in a multiple-service-class environment. However, effective bandwidth normally accounts for the worst-case traffic subject to the traffic profile of the SLA. Hence, the contract for typical users has an effective bandwidth much larger than the mean rate, and provisioning based on equivalent bandwidth is not economically efficient in a DiffServ environment. In addition, performance guarantees in DiffServ are qualitative and can be very loose. This may make it difficult to evaluate the equivalent bandwidth. Since DiffServ does not allocate resources to applications based on their effective bandwidth, but only provide users an average performance reflecting the service level, it appears unfair to charge users based on their effective bandwidth which can be derived from the user profile declaration or through user traffic measurements.

To encourage users to reduce their resource requirements under network resource contention, we propose an additional congestion-sensitive price component under these conditions. The general network resources considered are bandwidth and buffer space. Two kinds of congestion pricing can be considered: pricing when the expected load bound is exceeded, or pricing when buffer occupancy reaches certain level. In the first case, when the average demand for a certain class exceeds a threshold, an additional congestion price is charged over all users of that class.

In the case of priority dropping for AF class, the dropping precedence is only considered when the buffer occupancy reaches different thresholds. When each threshold is reached, user packets with the corresponding precedence level begin to be dropped with a certain probability, and users with higher precedence levels are charged the additional buffer price. Therefore, the higher precedence users pay the sum of buffer prices corresponding to all the exceeded thresholds. During congestion, lower precedence users will suffer lost packets, or reduce their rate, or smoothen their traffic at the source (at the cost of higher delay due to buffering), or change to a higher precedence and pay a higher price.

Both kinds of congestion price for a service class can be calculated as an iterative tâtonnement process [23], which implements the welfare theory in a competitive market. The price change, upward when aggregate user demand exceeds resource supply and downward when demand is lower than the supply, drives the demand and supply towards equilibrium.

In this paper, we consider pricing based on the load level of a service class. Congestion price for an interval n can be calculated through an iterative tâtonnement process as:

$$p_c^j(n) = \min[\{p_c^j(n-1) + \sigma^j(x^j - \rho_*^j)/\rho_*^j, 0\}^+, p_{max}^j], \tag{15}$$

where x^j and ρ_*^j represent the current total offered network load and target bandwidth utilization for service class j respectively, σ^j is a factor used to adjust the convergence speed [2], and p_{max}^j is the highest congestion price that can be applied.

[2]For an integral controller, higher control gain σ leads to a faster response of the congestion price p_c. However, large values of σ can cause excessive

Equation 15 follows the integral control law [29] to drive the user demand towards the target bandwidth utilization. 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. The maximum congestion price is bounded by the p^j_{max}. When a service class needs admission control, all new arrivals are rejected when the price reaches p^j_{max}. If p^j_c reaches p^j_{max} frequently, it indicates that more resources are needed for the corresponding service, or usage price for a class needs to be adjusted to reflect the new demand statistics. For a period n, the total congestion charge is given by

$$c^{ij}_c(n) = p^j_c(n)v^{ij}(n). \tag{16}$$

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

$$c^{ij}_s = \sum_{n=1}^{N}[p^j_h(r^{ij}(n)\tau^j - v^{ij}(n)) + (p^j_u + p^j_c(n))v^{ij}(n)], \tag{17}$$

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 or admission control, for example, best effort service.

Even though we have introduced a pricing model that consists of different pricing components, the end users only needs to know the total price of a service class. At service request time, a user can assume it will use all the resource requested, and hence the holding charge will be zero. Therefore, network only needs to announce the price of a service class as the total of usage price and congestion price. As both usage price and congestion price are volume dependent, the user can use the equation presented in Section II to calaculate its service requests accordingly.

B. System Stability and Network Dynamics

Application adaptation as well as applications entering and leaving the network lead to resource re-allocation and possibly adjustment of service prices. The re-negotiation of network services will generally be driven by price or user requirement changes. In our proposed pricing strategy, 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. We show the stability of our pricing algorithm in the appendix.

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 our proof of price stability, the user resource requests are assumed to be known instantaneously. For a network with transmission delay, this assumption may not be true. However, we study the pricing models in the environment where the network adjusts pricing periodically in the time scale of minute or longer. Since the time period between price adjustments is relatively long, the network transmission delay has negligible impact on the system performance and stability, which is confirmed by our simulations.

On the other hand, 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 allocating resources in blocks, reducing the frequency of resource re-allocation and hence price adjustment. The resources negotiated will be incremented or decremented with some minimum granularity. When the sum of user requests approaches the resources reserved for the aggregate, an additional block of resources can be reserved. Similarly, the resources reserved are decremented in blocks as the requested bandwidth decreases. The larger the block, the less frequently the resources need to be re-negotiated, but a higher holding cost may be incurred for resources under-utilized. In our work, we also used a price adjustment threshold parameter θ^j for a service class j to limit the frequency with which the price is updated. The congestion price is updated if the calculated price increment exceeds $\theta^j p^j_c(n-1)$.

IV. 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) [5][**?**], using a centralized (RNAP-C) network management architecture. We first briefly review the RNAP framework, and then describe the pricing and charge formulation process used.

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. The users know the fixed service prices for different service classes. In addition, in an RNAP session, the NRN periodically provides the HRN updated congestion prices for a set of services through a *Quotation* 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, to reduce the overhead due to per-flow RNAP message processing and storage, we consider a

oscillation or instabilities. Also, if minimum and maximum limits are set on the congestion price (say, zero and $p_{\max}$ respectively), setting σ too high can force p_c into one of the limit states. Assume ϵ is the largest error that occurs in closed-loop operation; to avoid forcing p_c into a limit state, σ should be set no higher than $\frac{p_{\max}}{\epsilon}$.

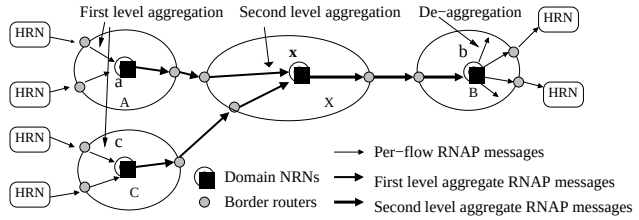

Fig. 1. Example RNAP-C message aggregation.

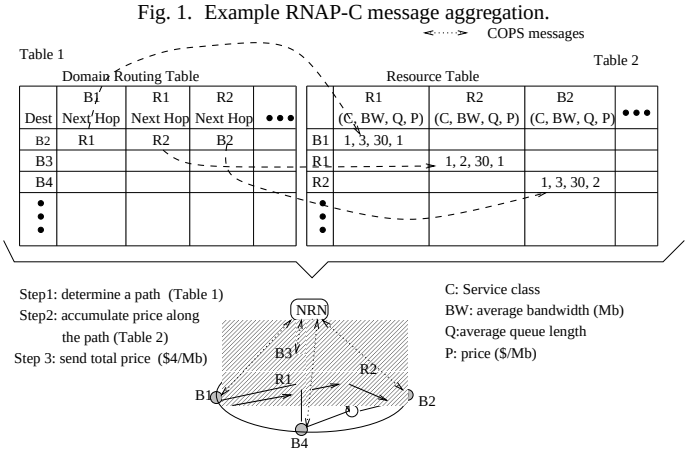

Fig. 2. Price formulation in RNAP-C

sink-tree based aggregation scheme as shown in Fig. 1. The aggregation and de-aggregation entities are NRNs. At an aggregating NRN 'a' or 'c', the aggregate *Reserve* message will be formed and sent domain by domain towards the destination domain NRN 'b'. Multiple levels of aggregation can occur, so that aggregate messages are aggregated in turn, resulting in a progressively thicker aggregate "pipe" towards the root of the sink-tree. The address of the destination domain NRN is located through DNS SRV [**?**]. In addition, the access domain NRN encapsulates the per flow *Reserve* messages with UDP packet headers and tunnels them directly to the destination domain NRN 'b' to reserve resources in the destination domain.

The destination domain NRN sends a *Commit* messages "hop by hop" (each hop is one domain) upstream towards the source in response to an aggregate *Reserve* message. The intermediate domain NRNs will deaggregate the message progressively as an aggregate response message passes by. They will map the aggregate-level pricing and charging (returned by the aggregate session *Quotation* and *Commit* messages) to prices and charges for the corresponding deaggregated messages based on the local policy. All the per-flow response messages are tunneled from destination domain directly to the access domain of the source. There is a similar message flow for RNAP *Quotation* messages in the upstream direction. With aggregation, the RNAP messages are processed at much larger granularity in the core networks.

The NRN maintains local state information for a domain for charging and other purposes. It makes the admission decision and decides the price for a service, based on the service specifications alone, or by also taking into account routing and configuration policies and network load. In the latter case, the NRN sits at a router that belongs to a link-state routing domain (for example an OSPF area) and has an identical link state database as other routers in the domain. This allows it to calculate all the routing tables of all other routers in the domain using Dijkstra's algorithm.

The NRN maintains a domain routing table which finds any flow route that either ends in its own domain, or uses its domain as a transit domain (Fig. 2). The domain routing table will be updated whenever the link state database is changed. A NRN also maintains a resource table, which allows it to keep track of the availability and dynamic usage of the resources (bandwidth, buffer space). In general, the resource table stores resource information for each service provided at a router. The resource table allows the NRN to compute a local price for each router (for instance, using the usage-based pricing strategy described in Section III). For a particular service request, the NRN first looks up the path on which resources are requested using the domain routing table, and then uses the per-router prices to compute the accumulated price along this path. The resource table also facilitates monitoring and provisioning of resources at the routers. To enable the NRN to collect resource information, routers in the domain periodically report local state information (for instance, average buffer occupancy and bandwidth utilization) to the NRN. In this paper, we extend COPS [30] for this purpose.

To compute the charge for a flow, ingress routers maintain per-flow (or aggregated flow from neighboring domain) state information about the data volume transmitted during a negotiation period. This information is periodically transmitted to the NRN, allowing the NRN to compute the charge for the period.

A network domain manages its own pricing scheme (which may be congestion sensitive or static) independent of other domains, and will have its own per unit resource costs for each class. When an user flow traverses multiple domains, RNAP messaging collates pricing and billing information from each domain and determine the total price/charge for the user.

V. Simulation Model

In this section, we describe our simulation model for the CPA and FP policies. We simulate a single DiffServ service domain, under which resources are not explicitly reserved for each flow. We simulate the service performance with or without admission control from the domain. User resource requirements are declared explicitly through RNAP, allowing admission control to be enforced if required in an experiment. The individual and total user resource demands are also obtained through measurement. Price and network statistics are signaled to users through RNAP.

We used the *network simulator* [31] environment to simulate two 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 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 congestion to be simulated at a single bottleneck node, and use topology 2 to illustrate the CPA performance under a more general network topology [33].

We modified the DiffServ module developed by Sean Murphy to support dynamic SLA negotiation, and monitor the user traffic at ingress point. A Weighted-Round-Robin scheduler is

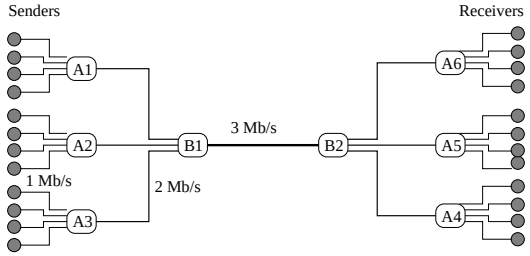

Fig. 3. Simulation network topology 1

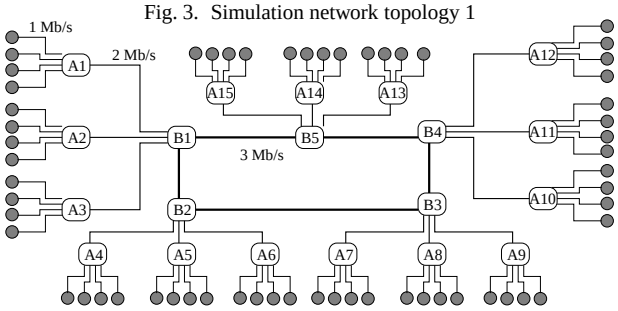

Fig. 4. Simulation network topology 2

modeled at each node, with weights distributed equally among EF, AF, and Best Effort (BE) classes. Although the DiffServ proposals mention four AF classes with three levels of drop precedence in each, we only simulated one AF class to make the simulations less resource-intensive, since this does not affect the general results in any way. Three different buffer management algorithms are used for different DiffServ classes - tail-dropping for EF, RED-with-In-Out [34] for AF, and Random Early Detection [35] for the BE traffic. The default queue length for EF, AF and BE are set respectively to 50, 100, 200 packets. Other parameters are set to the default values in the *network simulator* implementation.

A combination of exponential on-off and Pareto on-off traffic sources are used in the simulation. Unless otherwise specified, the traffic consists of 50% of each for all the service classes, and the on time and off time are both set to 0.5 seconds. The shape parameter for Pareto sources is set to 1.5. The mean packet size is set to 200 bytes. The traffic conditioners are configured with one profile for each traffic source, with peak rate and bucket size set to the on-off source peak rate and maximum amount of traffic sent during an on period respectively for both EF and AF classes.

We also characterize the system load by *burst index* and *offered load*. The burst index is defined as *OffTime/(OnTime + OffTime)* for both types of on-off sources. The offered load for a service class is defined as the ratio between the total user resource requirement for a service type, and the configured class capacity at the bottleneck. 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.

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. In topology 1, users from the sender side independently initialize unidirectional flows towards randomly selected receiver side end nodes. N_s flows will be initialized at one node. At most $12N_s$ flows (60 sessions with N_s set to 5) 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.

For ease of understanding, all prices in this section are given in terms of price per minute of a 64 kb/s transmission, currently equivalent to a telephone call. The basic price charged by the FP policy, and the basic usage price charged by CPA (p_{basic}), are both set to \$0.08/min. We set the target average load of the EF class at 40%, the AF class at 60%, and the BE class at 90%. Therefore, based on the pricing strategy proposed in Section III, the usage price for EF, AF and BE classes are set respectively as \$0.20/min, \$0.13/min, and \$0.089/min. When admission control is enforced, the holding price for the CPA policy is correspondingly set to \$0.067/min for EF class, and \$0.044/min for AF class.

Congestion pricing is applied when instantaneous usage exceeds the target load threshold of each class or when the loss or delay exceeds $1/3$ of the bounds at a node associated with the class (delay bound of 2 ms, 5 ms, and 100 ms respectively for EF, AF, and BE, and loss bounds of 10^{-6}, 10^{-4} and 10^{-2} respectively). The price adjustment procedure is also controlled by a pair of parameters, the price adjustment step σ from equation 15 and the price adjustment threshold parameter θ, defined in Section III. Unless otherwise specified, values of $\sigma = 0.06$ and $\theta = 0.05$ are used.

The users are assumed to have the general form of the utility function shown in Section II. At the beginning of each experiment, the user population is divided into users of the EF, AF and BE classes, although in some experiments they are allowed to adapt to price changes by switching to a different class.

For EF users, the elasticity factor factor w (which is also the user's willingness to pay), is uniformly distributed between \$0.13/min and \$0.40/min for a 64 kb/s bandwidth. For AF and BE users, it is uniformly distributed between \$0.09/min and \$0.26/min, and \$0.06/min and \$0.18/min respectively. The minimum delay and loss requirements for each type of users are set to be the same as the expected performance bound of the corresponding service class. The opportunity cost parameter U_0 is set to the amount a user is willing to pay for its minimum bandwidth requirement, and is hence given by $U_0 = p_{high} \cdot x_{min}$, where p_{high} is the maximum price the user will pay before terminating his connection altogether. Users re-negotiate their resource requirements with a period of 30 seconds in all the experiments. The total simulation time for each experiment is 20,000 seconds.

We use a number of engineering and economic metrics to evaluate our experiments. The engineering metrics include the average traffic arrival rate at the bottleneck, the average packet delay, the average packet loss rate, and the user request blocking probability. The averages are computed as exponentially weighted moving averages. The economic performance metrics include the average user benefit (the perceived value obtained by users based on their utility functions), the end-to-end price for each service class.

VI. Results and Discussion

In this section, we simulate the FP policy and CPA policy under identical traffic conditions, and compare the relative performance.

For ease of presentation, a single traffic parameter for the AF class was varied in each experiment, and its effect on CPA and FP policy performance was studied. We conducted four groups of experiments. In the first and second groups, we vary the load burstiness and average load respectively of the AF class, and evaluate the improvements given by CPA over FP. In the third experiment, incentive driven traffic migration between classes is shown to improve the overall system performance. In the last experiment, we show that admission control to a service class is critical in maintaining expected performance levels. Combining admission control with user service adaptation effectively reduces the request blocking rate.

A. Effect of Traffic Burstiness

We first compare the performance of FP and CPA policies as the burst index of AF class increases, at a constant average offered load of 60%.

Fig. 5 (a) shows that the average AF price increases under CPA due to the increasing congestion price as the burst index exceeds 0.4. In response, the AF traffic backs off. Fig. 5 (a) also shows that the standard deviation in the AF price increases with the burst index, indicating greater fluctuations in the price. Fig. 5 (b) shows the dynamic variation of the AF class price at three different levels of burstiness, confirming this trend.

Fig. 5 (c) and (d) show that under FP policy the average packet delay and loss of the AF class increase sharply as the burst index exceeds 0.4. As a result of the user traffic back-off under CPA the delay and loss of AF class are well controlled below the respective performance bounds of 5 ms and 10^{-4} up to a burst index of 0.8. The average user benefit for CPA (Fig. 5 f) decreases due to the reduction of bandwidth, but remains higher than that of the FP policy. There is also a smaller degradation in the performance of the BE class at high burst indices. This appears to be because the BE class operates under a relatively high load, and therefore borrows bandwidth from the AF class when the AF class is lightly loaded. It can no longer do so when the AF traffic burstiness increases.

The results in this section indicate that the CPA policy takes advantage of application adaptivity for significant gains in network performance, and perceived user benefit, relative to the fixed-price policy. The congestion-based pricing is stable and effective.

B. Effect of Traffic Load

In this simulation, we keep the load and burstiness of EF class and BE class and the burst index of the AF class at their default values, and vary the offered load of AF class. The average AF price under CPA is seen to increase with offered load (Fig. 6 (a)). The standard deviation of the price shows an increase to a certain level and then a decrease. Initially, the price deviation increases due to the more aggressive congestion control. At heavy loads, the increased multiplexing of user demand smooths the total demand, and therefore reduces fluctuations in the price. Fig. 6 (e) shows that the actual arrival rate of AF under CPA backs off as users adapt to the higher price.

Figs. 6 (c) and (d) show that the delay and loss of AF class under FP quickly increases after the offered load increases above 0.6 and approaches the provisioned capacity. As a result, the performance bounds for AF class can no longer be met. The high AF load also degrades BE performance. This is apparently because BE operates at a high load (0.9) and tends to borrow bandwidth from AF and EF when the latter classes are lightly loaded.

Figs. 6 (c), (d), and (e) show that CPA coupled with user adaptation is able to control congestion and maintain the total traffic load of a service class at the targeted level, and hence allows the service class to meet the expected performance bounds. Similar to our observation in Section VI-A, if the nominal price of the system correctly reflect long-term user demand, dynamic pricing driven service re-negotiation can effectively limits short-term fluctuations in load. Usage price of a class should be adjusted if persistent high user demand exist for a service.

C. Load Balance between Classes

As seen from the previous section, the performance of a class will suffer if the load into that class is too high. In general, a user under CPA policy will select a service class which provides it the highest benefit based on the price and performance parameters of a class as announced by the providers. The performance parameters are generally based on long-term statistics. In this section, we assume that a user can learn from network performance data received over a short period, and select the class that would provide the highest benefit based on the user utility function, network performance statistics and service price, as discussed in Section II.

In this simulation, the EF and BE classes are loaded at 30% and 80% respectively. When the load of AF class increases, the performance of AF class degrades and congestion price is invoked. In response, some applications switch from the AF class to the EF class, which provides better performance guarantee, or BE class, which allows it more bandwidth at a cheaper price. As the result of this re-selection, the load is better balanced across classes, and overall performance of the system improves (Fig. 7 (c) and (d)). Fig. 7 (a) shows that with load balancing in combination with adaptation within a single class, the congestion price needs to be invoked much less often than with adaptation within a class only, as in Fig. 6 (b). The proportion of migrating traffic is shown in Fig. 7 (b). We see even when a small portion of users select other service classes, the performance of the over-loaded class is greatly improved.

D. Effect of Admission Control

We have seen that the performance of a class can not be expected without any admission control. In this section, we compare the performance of FP and CPA for a network with admission control for EF and AF class. The admission threshold for each class is set to 1.5 times the target load to increase the efficiency of the network.

With admission control, the performance of EF and AF classes are well controlled (Fig. 8 c and d). However, due to the burstiness of the traffic, the blocking rate under FP is high even at a very small offered load (Fig. 8 b), and increases almost linearly as the offered load increases beyond 0.6. With congestion control and service contract re-negotiation, the blocking rate of CPA is seen to be up to 30 times smaller than that under the FP policy, and actually starts to decrease after reaching a maximum at offered load 0.8. 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

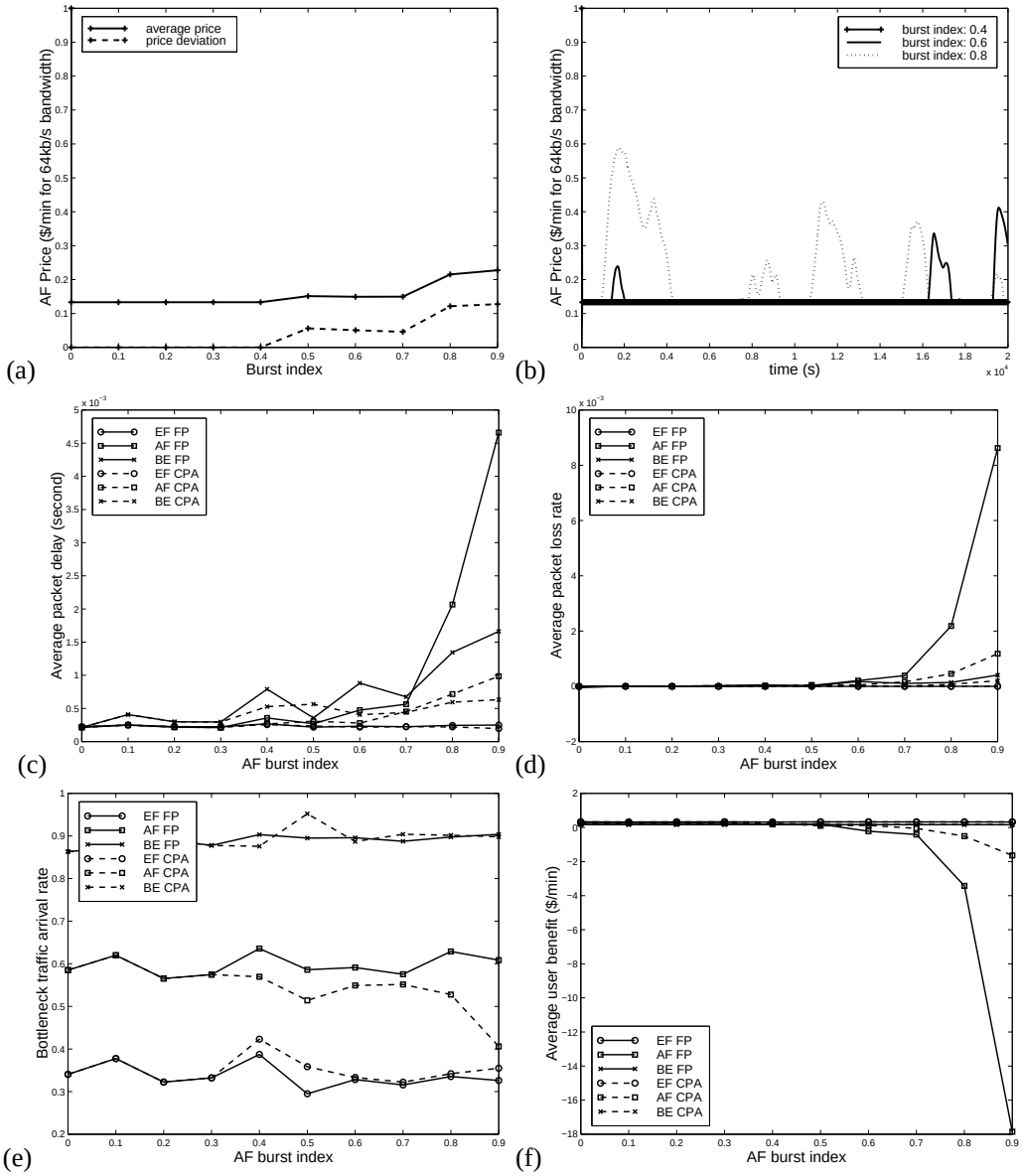

Fig. 5. System dynamics under CPA with increase in AF traffic burst index: (a) price average and standard deviation of AF class; (b) variation over time of AF class price. Performance metrics of CPA and FP policies as a function of burst index for classes: (c) average packet delay; (d) average packet loss; (e) average traffic arrival rate; (f) average user benefit.

higher loads, and the user bandwidth request decreases proportionally with the price according to the general utility function of Section II. Compared to Section VI-B, the average price under CPA (Fig. 8 a) is bounded to a smaller value at high offered loads, and has a smaller fluctuation.

The results indicate that admission control is important in maintaining the expected performance of a class. However, admission control by itself may lead to a high blocking rate due to the network dynamics. By combining admission control with user traffic adaptation, the network is more efficiently used. With admission control, the dynamics of the network price can also be better controlled, so that users have a more reliable expectation of the price.

VII. Related Work

Microeconomic principles have been applied to various network traffic management problems. The studies in [36][37][38][39][15] 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 user 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 the knowledge of the user's utility functions, which is generally not practical.

Theoretical frameworks of congestion pricing have been discussed thoroughly by several authors [15][40][41][42]. Kelly et al [15] and Low et al [40] 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 [41] to convey congestion information back to the end systems, and the resulting system converges to an optimal system state as long as all utility curves are strictly concave. Instead of only marking the packets, the authors in [42] proposed assigning each packet a price to reflect the con-

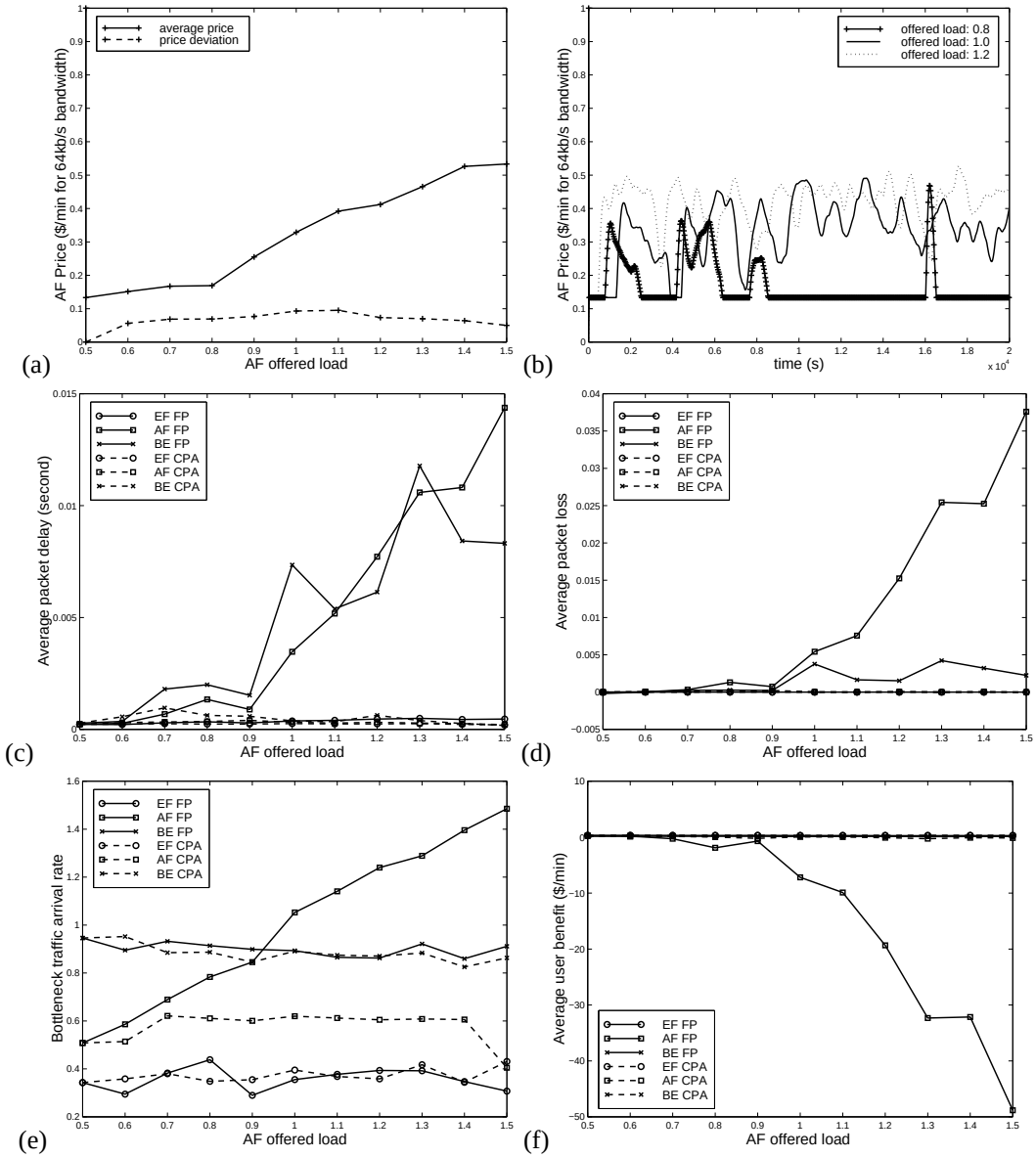

Fig. 6. System dynamics under CPA with increase in AF offered load: (a) average and standard deviation of AF class price; (b) variation over time of AF class price. Performance metrics of CPA and FP policies as a function of AF offered load: (c) average packet delay; (d) average packet loss; (e) average bottleneck traffic arrival rate; (f) average user benefit.

gestion of the network. These schemes assume network services are best-effort, and rely on a pure market mechanism to maximize social benefit. It is also not clear whether all the theoretical results hold in the presence of transmission delay at the scale of a large network.

Several auction-based mechanisms have been studied to elicit truthfully reported user utility functions and encourage the efficient utilization of scarce network resources. In the "smart market" model [43], each packet header contains a bid field, and the packet is admitted if the bid exceeds the current cutoff amount, determined by the marginal congestion costs. The mechanism only provides a priority relative to other users, and is not an absolute promise of service. Issues that need to be addressed include accounting complexity, service interruptions during traffic peaks, and user response to fluctuations in price. The model in [44] supports multiple levels of QoS guarantees. The implementation scheme is again called "smart market", also called "generalized Vickrey auction (GVA)". GVA extends the idea of [43] to allow agents to have preferences over more than one item, and more than one unit of the item. The optimal solution requires substantial computation, which increases polynomially with the number of users, and the number of optimizations increases linearly with the number of users. The progressive second price auction (PSP) scheme proposed by Semret et al [45] extends the traditional single non-divisible object auction to the allocations of arbitrary shares of the total available resource. Yuksel and Kalyanaraman [46] investigated the implementation issues of the smart market model and proposed a strategy for implementing smart market pricing in DiffServ framework.

In [47][48][49][50], the resources are priced to reflect demand and supply. However, the methods in [47][48][50] rely on on well-defined sources models and cannot adapt well to changing traffic demands. Similar to our work, the scheme in [49] also takes into account network dynamics (session join or leave) and

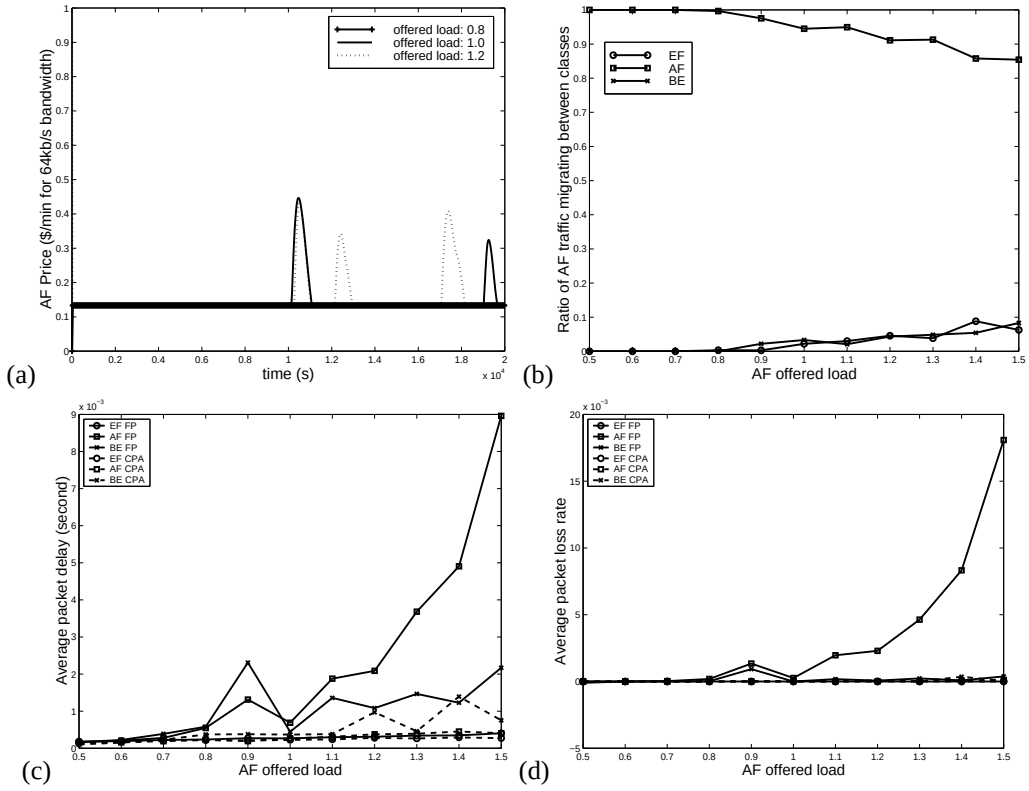

Fig. 7. Performance metrics of CPA and FP policies with traffic migration between classes: (a) variation over time of AF class price; (b) ratio of AF class traffic migrating through class re-selection; (c) average packet delay of all classes; (d) average packet loss of all classes.

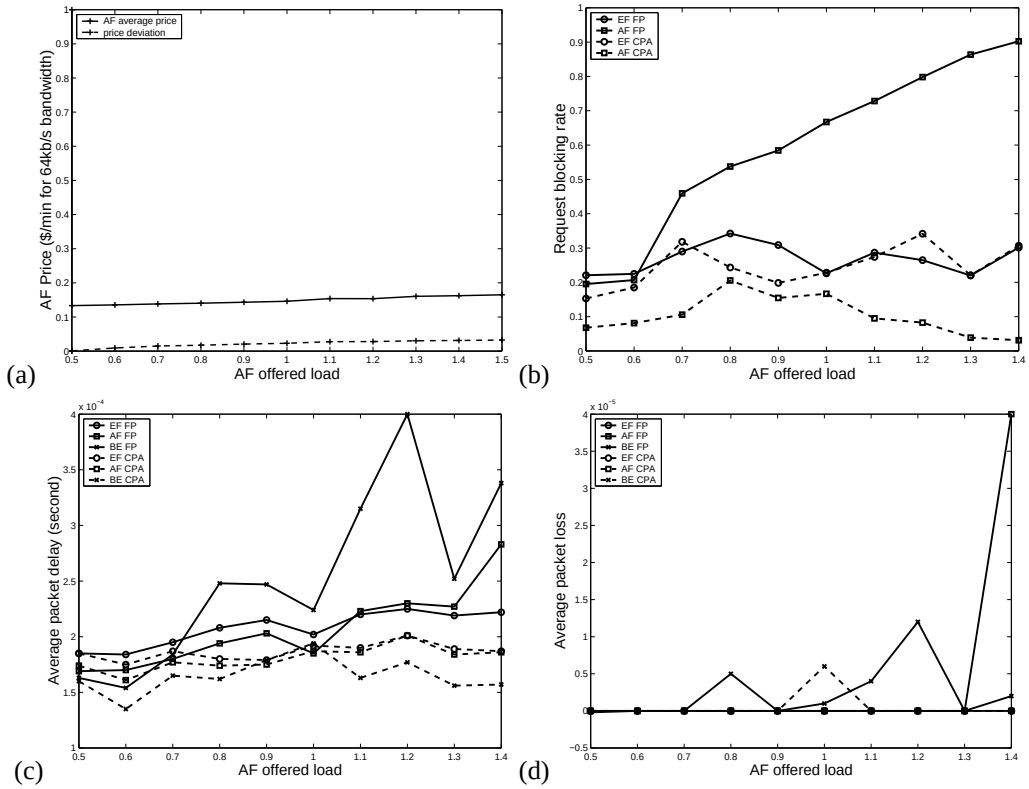

Fig. 8. System dynamics under CPA with admission control as AF offered load increases: (a) average and standard deviation of AF class price. Performance metrics of CPA and FP policies with admission control as a function of AF offered load: (b) user requests blocking rate; (c) average packet delay; (d) average packet loss.

source traffic characteristics, and 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.

The study in [51] demonstrates through experiments that compared to traditional flat pricing, service-class sensitive pricing results in higher network performance. Paris Metro Pricing (PMP) [52] scheme partitions the network into logically separated channels with different prices. It is expected that the higher-priced classes will have less load and will provide better service. The behavior of PMP under equilibrium conditions is considered and compared with a uniclass pricing system in [53][54]. Marbach [55] analyzed the equilibrium of such a system using non-cooperative game theory. Altmann et al [56] considered a similar framework based on queuing theory and experiments. The PMP-related work considered the impact of differential pricing on the relative performance of the system as a result of user self selection process. There is however no guarantee on the service quality delivered at each priority level.

A set of other game-theoretic algorithms have been proposed for multi-class QoS provision. In [57], packets are marked according to customer's QoS requirements. Without associating any price with a service class, the costs incurred to customers are purely performance related. The authors in [58] studied the dimensioning of network capacity for different service classes, while Mandjes [59] and Marbach [60] studied the static pricing scheme based on the priority classes. Marbach [60] extended the number of classes from two [59] to a finite number and showed how pricing decisions would affect the link performance and revenue. However, as shown in Marbach [60], the strict priority scheduling can lead to a two-level service which does not allow multi-class QoS provisioning. The work of Shu and Varaiya [61] generalized the idea in [43] to support auctions for different service levels. The in-profile traffic is charged flat fee without specifiing how the price can be determined, while out-profile traffic is charged congestion price based on an auction-based admission control algorithm and treated differently for different user bids.

Kumaran et al. [62] described the utility maximization by users and revenue optimization by service providers based on the quantitative admission control model proposed in [63][64]. The equivalent bandwidth estimation in [63][64] is under very specific assumptions about the traffic, and the analysis is also constrained to two classes, with users in the same class assumed to have the same bandwidth requirements. This prevents the application of the results to a general network service environment. O'Donnell et al [65] borrowed the framework in [42] and calculated a price for each packet based on bandwidth consumption, service level, and buffer occupancy, which may result in high implementation cost.

Jordan [66] proposed adjusting bandwidth and buffer allocations between classes in a network with reservation-based QoS to guarantee the target delay and loss. In our work, we assumed the resource provisioning for classes does not need to be adjusted unless longer-term network performance can not be ensured. In a short-term, our pricing schemes motivate user request adaptations to gain the target service performance. Our proposed model could hence jointly work with that of [66], in different time scales. Pricing for DiffServ has also been studied in [18] through equivalent bandwidth. As has been pointed out earlier, equivalent bandwidth may be too conservative for resource provisioning in a DiffServ environment, and hence pricing based on equivalent bandwidth may not be fair to the users. Also, it is not trivial for users to adapt their requirements dynamically to meet their equivalent bandwidth constraints.

In general, the literature work on network pricing is restricted to theoretical issues, and the results are not easily applicable to real networks. We have designed a pricing algorithm that takes into account the current Internet service infrastructure, and considered different service classes, the long-term user traffic demand, and short-term network dynamics. The algorithm allows the network to optimize profit, and also allows a user to select service flexibly based on its preferences and to maximize its total net benefit. Also, the literature generally does not enter into details about the negotiation process and the network architecture, and mechanisms for collecting and communicating prices. We address these issues by integrating our pricing algorithm in a resource negotiation framework with pricing and billing mechanisms.

By putting our pricing algorithm in an infrastructure consisting of both static and dynamic pricing components, our model allows a network provider to play different performance tradeoffs and support different business models. The literature work on network economics normally focus on studying the performance due to pure market mechanisms. However, we consider economic approach a means to motivate more efficient network resource usage and also emphasize the engineering approach as a means to enforce network service assurance. That is, our model gives users the options for network services based on their individual valuation of the transmissions, and also provides engineering control to restrict the load of a class to its target level when an economic approach is not enough for providing expected service (for example, not enough users would adapt their service requirements when congestion happens) or not supported by a network. Our pricing model enables different service levels through price differentiation and admission control, and also uses congestion price as a signal to control network traffic dynamics. We have studied the performance of a system with both price differentiation and service adaptation, and admission control and queue management, to regulate the traffic load of a service class. Our results indicate that the performance of the system supported by both economic and engineering mechanisms is superior to that managed by using only one of the means.

VIII. Summary

In this work, we have developed a reasonably complete DiffServ pricing model. We have proposed a price structure for different service classes in DiffServ based on their relative performance, long-term demand, and short-term fluctuations in demand. We have integrated this pricing model into a dynamic service negotiation environment in which service prices increase in response to congestion, and users adapt to price increases by adapting their sending rate and/or choice of service. We have also modeled the demand behavior of adaptive users based on a physically reasonable user utility function.

Our simulation results show that different service classes provide different levels of service only when they operate at different target utilization. In the absence of explicit admission control, a service class loaded beyond its target utilization (under ei-

ther sustained or bursty loads) no longer meets its expected performance levels. Under these conditions, a congestion-sensitive pricing policy (CPA) coupled with user rate adaptation is able to control congestion and allow a service class to meet its performance assurances under large or bursty offered loads. Users see a reasonably stable service price and are able to maintain a very stable expenditure. Allowing users to migrate between service classes in response to price increase and network performance further stabilizes the individual service prices while maintaining the system performance.

When admission control is enforced for each class, performance bounds can be met with a fixed service price. However, in this case, the CPA policy provides a greatly reduced connection blocking rate at high loads by driving down individual bandwidth requests, resulting in a higher overall user satisfaction. Compared to the CPA policy without admission control, the service price is further stabilized in this case.

In this paper, we assume that users do not have the option of choosing a different path or provider, reflecting current network reality. However, pricing in the presence of competition or alternative paths remains an interesting open issue.

APPENDIX

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 $X(p)$. For a network service class, the targeted resource supply is fixed and is denoted as ρ^*. Suppose the rate of change of price moves directly with excess demand, $E(p) = X(p) - \rho^*$ as follows:

$$p' = \frac{dp}{dt} = f(X(p) - \rho^*) = f(E(p)), \tag{18}$$

where $f' \geq 0$, and $f(0) = 0$. The price change drives the demand and supply towards equilibrium. If the *tâtonnement* process is successful, the mechanism in equation 18 will generate a path of prices which will approach p^e as t increases:

$$\lim_{t \to \infty} p(t) = p^e \tag{19}$$

If equation 18 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 19 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:

$$\frac{dp}{dt} = f(E(p^e)) + f'(E(p))E'(p^e)(p - p^e) + \cdots \tag{20}$$

The higher order terms are negligible in comparison with the first-order term in equation 20, as long as only *local* stability is considered. Since $E(p^e) = 0$ by the definition of price, the equation 20 can be written as:

$$\frac{dp}{dt} = f'(E(p))E'(p - p^e) \tag{21}$$

The solution of this equation is:

$$p(t) = p^e + (p^0 - p^e)e^{(f'(E(p))E')t}, \tag{22}$$

where p^0 is any initial price.

The assertion of stability requires that the exponential term in equation 22 approaches zero as $t \to \infty$. Since $f' > 0$, so the stability assertion requires

$$E' = X_p(p) < 0. \tag{23}$$

In a reasonable network system, user demand will decrease as the price increases, so $X_p(p) < 0$. As our congestion price in equation 15 follows equation 18, this proves that our proposed price will reach stability as times increases.

REFERENCES

[1] 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.

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

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

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

[5] 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.

[6] X. Wang and H. Schulzrinne, "Performance study of congestion price based adaptive service," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'00)*, (Chapel Hill, North Carolina), pp. 1–10, Jun 2000.

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

[8] 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.

[9] 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.

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

[11] X. Wang and H. Schulzrinne, "An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2514–2529, Dec 2000.

[12] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE JSAC*, vol. 9, pp. 968–981, Sept. 1991.

[13] 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.

[14] 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.

[15] 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.

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

[17] J. Janssen, D. D. Vleeschauwer, and G. H. Petit, "Delay and distortion bounds for packetized voice calls of traditional PSTN quality," in *Proceedings of the 1st IP Telephony Workshop (IPTel 2000)*, (Berlin, Germany), pp. 105–110, Apr 2000. GMD Report 95.

[18] C. Courcoubetis and V. Siris, "Managing and pricing service level agreements for differentiated services," in *Proc. of 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99)*, (London, UK), Jun 1999. GMD Report 95.

[19] 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.

[20] R. Edell and P. Varaiya, "Providing Internet access: What we learn from INDEX," *IEEE Network Magazine*, vol. 13, Sept. 1999.

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

[22] J. Altmann and K. Chu, "A proposal for a flexible service plan that is attrative to users and Internet service providers," in *Proc. of Infocom*, (Anchorage, Alaska), Apr 2001.

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

[24] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Austin, Texas), pp. 1–12, ACM, Sept. 1989. also in Computer Communications Review, 19 (4), Sept. 1989.

[25] A. K. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, Jun 1993.

[26] A. K. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr 1994.

[27] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, vol. 3, Aug 1995.

[28] F. P. Kelly, S. Zachary, and I. Zeidins, "Notes on effective bandwidths," *Stochastic Networks: Theory and Applications*, pp. 141–168, 1996.

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

[30] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A. Sastry, "The COPS (common open policy service) protocol," Request for Comments 2748, Internet Engineering Task Force, Jan 2000.

[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] X. Wang and H. Schulzrinne, "Performance study of congestion price based adaptive service," Technical Report CUCS-010-00, Columbia University, New York, Apr 2000.

[34] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Trans. Networking*, vol. 6, pp. 362–373, Aug 1998.

[35] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug 1993.

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

[37] 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.

[38] 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.

[39] S. Low and P. Varaiya, "An algorithm for optimal service provisioning using resource pricing," in *Proc. of Infocom*, (Toronto, Canada), Jun 1994.

[40] 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.

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

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

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

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

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

[46] M. Yuksel and S. Kalyanaraman, "S strategy for implementing smart market pricing scheme on Diff-Serv," in *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, (Taipei, Taiwan), IEEE, Nov 2002.

[47] 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.

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

[49] 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.

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

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

[52] A. Odlyzko, "Paris metro pricing: The minimalist differentiated services solution," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, (Basking Ridge, New Jersey), Jun 1999.

[53] R. Gibbens, R. Mason, and R. Steinberg, "Internet service classes under competition," *IEEE JSAC*, vol. 18, pp. 2490–2498, Dec 2000.

[54] R. Jain, T. Mullen, and R. Hausman, "Analysis of paris metro pricing strategy for QoS with a single service provider," *Lecture Notes in Computer Science*, vol. 2092, pp. 44–58, Jun 2001. International Workshop on Quality of Service (IWQoS).

[55] P. Marbach, "Pricing differentiated services networks: Bursty traffic," in *Proc. of Infocom*, (Anchorage, Alaska), Apr 2001.

[56] J. Altmann, H. Oliver, H. Daanen, and A. S.-B. Suarez, "How to market-manage a QoS network," in *Proc. of Infocom*, (New York, New York), Jun 2002.

[57] S. Chen and K. I. Park, "An architecture for noncooperative QoS provision in many-switch systems," in *Proc. of Infocom*, (New York), Mar 1999.

[58] P. Fuzesi and A. Vidacs, "Game theoretic analysis of network dimensioning strategies in differentiated services networks," in *Conference Record of the International Conference on Communications (ICC)*, (New York, NY, USA), pp. 1069–1073, Apr 2002.

[59] M. Mandjes, "Pricing strategies under heterogeneous service requirements," in *Proc. of Infocom*, (San Francisco, USA), IEEE, Apr 2003.

[60] P. Marbach, "Analysis of a static pricing scheme for priority services," *IEEE/ACM Trans. Networking*, To appear.

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

[62] K. Kumaran, M. Mandjes, D. Mitra, and I. Saniee, "Resource usage and charging in a multi-service multi-QoS packet network," Dec 1999.

[63] A. Elwalid, D. Mitra, and R. H. Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous regulated traffic in an ATM node," *IEEE JSAC*, vol. 13, pp. 1115–1127, Aug 1995.

[64] A. Elwalid and D. Mitra, "Design of generalized processor sharing schedulers which statistically multiplex heterogeneous QoS," in *Proc. of Infocom*, (New York), Mar 1999.

[65] A. J. O'Donnell and H. Sethu, "A novel, practical pricing strategy for congestion control and differentiated services," in *Conference Record of the International Conference on Communications (ICC)*, (New York, NY, USA), pp. 986–990, Apr 2002.

[66] S. Jordan, "Pricing of buffer and bandwidth in a reservation-based QoS architecture," in *Conference Record of the International Conference on Communications (ICC)*, May 2003.