Price Discovery at Network Edges

Gurjeet S. Arora[†], Murat Yuksel[‡], Shivkumar Kalyanaraman[*], Thiagarajan Ravichandran[**], Aparna Gupta[***]
[‡]CS Department, [*]ECSE Department, [**]School of Management, [***]DSES Department
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY, 12180, USA
garora@emc.com, {yuksem@cs., shivkuma@ecse., ravit@, guptaa@}rpi.edu

Keywords—**Network Pricing, Congestion Pricing, QoS, Optimization**

Abstract— **Congestion-sensitive pricing for providing better than best effort service has received significant attention in the last decade. In this paper we identify a robust parameter for capturing congestion conditions in an edge-to-edge framework and propose a family of adaptive pricing schemes for premium network services. The parameter is the ratio of two values: *Edge queue* and *estimated edge-to-edge capacity*. By coordination between edge routers, both of the values are available at the ingress point in an edge-to-edge framework. Thus, the pricing schemes deployable. Based on the identified parameter, we propose a new adaptive pricing framework, Price Discovery. Based on the Price Discovery framework and the identified pricing parameter, we develop and analyze four pricing schemes. We compare the pricing schemes, and select the best one in performance. We identify stability conditions for the best scheme. This is followed by evaluation of the best pricing scheme with extensive simulations of various scenarios.**

I. INTRODUCTION

The players that engage in transactions involving bandwidth can be grouped in the following three categories: Capacity providers, re-sellers (Internet service providers and other bandwidth service providers) and end-consumers. The transactions between the capacity providers and re-sellers are best handled in an exchange kind of environment This is because the environment offers the capacity providers and re-sellers a variety of options by providing access to a large number of capacity providers and re-sellers. Another advantage of exchanges is transparent pricing, which encourages healthy competition. This kind of pricing where typical transactions (*between capacity providers and re-sellers*) are for a large amount of bandwidth is, what we call, *bandwidth pricing*.

Re-sellers create value by catering to the variety of needs of end-users (their diverse applications which vary in terms of their delay and jitter sensitivities). Bandwidth, among its other uses, is used for Internet services as well. The domain of pricing bandwidth for providing Internet services to end users (*transaction between re-sellers and end-users*)is known as *Internet pricing*. Several proposals have been made for Internet pricing. We can classify those proposals into two major groups: congestion-sensitive pricing proposals (e.g. [1], [2], [3], [4], [5], [6], [7], [8]), static pricing proposals (e.g. [9], [10], [11]). In congestion-sensitive pricing, as the name suggests, price per unit traffic volume varies by time depending upon the actual network congestion. In static pricing, price per unit service (as traffic volume or service duration) is fixed.

Congestion-sensitive pricing is becoming popular as a method for network resource allocation and congestion control. The main motivation behind this is that network performance cannot be solely derived from congestion control protocols [12]. During periods of resource contention or congestion epochs, there is a need to distinguish one packet from the other based on their importance as indicated by a utility function (utility functions are typically a combination of user preferences and application requirements). Thus during congestion, increasing prices of network services makes an allocation closer to Pareto efficient [13] (i.e. allocates resources to users such that the overall user surplus is close to the highest possible) and gives users right incentives to adjust their demands to alleviate congestion [2].

We can also classify the pricing proposals based upon their granularity. Some of the congestion-sensitive pricing proposals have used *per-packet charging*, while others have used *per-contract charging* which provides price information to the user prior to charging. MacKie Mason and Varian's Smart Market [3] proposes to use per-packet charges based upon the total marginal congestion cost the packet imposes on other users. For price determination, Gibbens and Kelly's Packet Marking scheme (also known as Proportional Fair Pricing) [2] also uses per-packet charging by using the number of packets marked at a congested network router.

Wang Schulzrinne's Resource Negotiation and Pricing (RNAP) [7] is an example of congestion-sensitive pricing scheme with per-contract charging. RNAP combines admission control with congestion pricing by using the service level agreement flexibility of diff-serv [14] architecture. RNAP leave the job of price determination to local network routers and uses probing techniques to determine edge-to-edge prices. Also, Dynamic Capacity Contracting (DCC) framework [8] uses per-contract charges by making price determination on an edge-to-edge basis at the edge routers rather than leaving it to local network routers (a major difference from RNAP).

In this paper, we propose a new pricing framework, *Price Discovery*, to solve the problem of allocating premium (better than best effort) Internet services without making any assumptions about user behavior. Price discovery is deployable in a general contracting framework in which end users and ISPs enter into service level agreements at the beginning of a contracting period for service. The ISPs vary price from one period to another in order to meet the service level agreements. Price for premium services does not change during a period in contracting framework.

Price Discovery attempts to employ congestion-sensitive pricing at network edges and uses an adaptive event-based algorithm to increase or decrease price. It operates in cycles of queue drain and queue build while discovering the price in an adaptive manner. In order to determine the price in period i, it uses two parameters: queue length at the edge (edge queue) q_{i-1} and estimated capacity C_i. With a simple coordination between ingress and egress edge routers, both q_{i-1} and C_i are available in an edge-to-edge framework. By using edge-to-edge congestion detection techniques (e.g. [15]), C_i can be varied to alleviate congestion, it typically decreases during congestion epochs (so that a part of available capacity can be used for draining the queues that build up during congestion) and increases (to increase utilization of the edge-to-edge capacity) when there is no congestion. So, Price Discovery takes advantage of the available information in an edge-to-edge framework and employs congestion-sensitive pricing to control the queue lengths at the network edges. Price Discovery is a general framework that can be applied to any scenario where q_{i-1} and C_i or their logical equivalents are available for price determination. DCC, for instance, provides both the parameters.

This work is sponsored by NSF under contract number ANI9819112.
[†] G. S. Arora is now with EMC Corporation.

Price Discovery is close to Shenker et al.'s Edge Pricing [11] proposal in terms of operating points, i.e. network edges. However, the major difference is that Edge Pricing proposes static usage-based pricing, while Price Discovery supports the idea of congestion-sensitive pricing using an adaptive algorithm.

The paper is organized as follows: In Section II, we investigate basic issues and concepts in an edge-to-edge framework based on diff-serv architecture. In Section III, we take a look at the problem of Pareto efficient network resource allocation in the face of congestion, while maintaining high utilizations. In Section IV, we present the Price Discovery framework, and show that the edge queue can be used as an effective parameter for pricing network services as it represents the cumulative difference between demand and capacity from the last time the queue was empty. Next in Section V, we formulate four possible linear pricing schemes (PIPD, PIAD, AIPD, AIAD) within the Price Discovery framework and show by comparative evaluation that Proportional Increase Additive Decrease (PIAD) performs best. Then in Section VI, we evaluate stability conditions for PIAD pricing scheme. In Section VII, we develop a simple user model and use it to evaluate PIAD. We present experimental results to show how the PIAD scheme reacts to congestions of different magnitude and how the PIAD parameters can be tuned to get desired queue lengths and other operating conditions (Section VII-F). We also evaluate premium service allocation done between two users for different base demand (demand at price=0) and reservation price (Section VII-G). Summary and directions for further work are presented in Section VIII.

II. Basic Edge-to-Edge Concepts

Given the recent trend toward diff-serv architecture to provide better than best effort services, it is desirable to perform complex operations at network edges and simple forwarding operations at network interior. In such an edge-to-edge scenario, it is possible to coordinate edge routers in order to achieve various objectives [15], such as congestion control, capacity estimation, flow control.

By employing edge-to-edge congestion control, it is possible to push congestion back from the interior of a network and distribute it across network edges where the smaller congestion problems can be handled with flexible and cheaper methods [15]. This method will create an environment where traffic backlogs occur at network edges, i.e. edge queues. In such an environment, one method of managing the edge queues is to employ congestion-sensitive pricing at the edges. Price Discovery has been designed to take advantage of this scenario, by leveraging the available information effectively in an adaptive pricing scheme that operates in cycles of queue build and queue drain (queue varying in ranges permitted by the service level agreements) in the process of discovering the optimum price. However, notice that Price Discovery does not necessarily require an edge-to-edge congestion control mechanism. It only requires some form of coordination between edge routers so that premium capacities available for next period can be estimated.

Given the possible coordination among edge routers (or a tight edge-to-edge congestion control), the problem of congestion-sensitive pricing can be reduced to a problem of using adaptive pricing to control the edge queues. Figure 1 shows the big picture of the scenario along with the parameters we will use in Price Discovery.

The parameter used to measure the severity of congestion in the pricing scheme is the edge queue at the ingress. q_{i-1} is the length of edge queue at the end of period $i-1$ and at the beginning of period i. C_i is the estimated capacity in period i calculated based on either edge-to-edge congestion control algorithms or edge-to-edge coordination used. E_i is the rate at which packets leave the egress. This will be lower than C_i during congestion epochs. X_i is the total demand during period i.

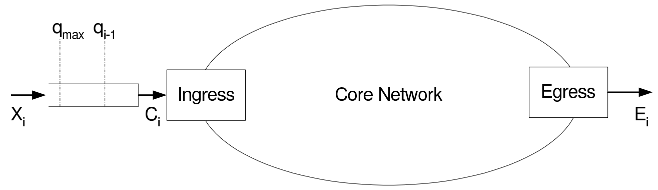


Fig. 1. Edge to edge framework at the beginning of period i.

III. Issues in Congestion Sensitive Pricing

In this section, we first formulate the utility maximization problem for a user consuming premium network service in a contracting framework. At the beginning of a contract period, each user contracts for network resources at a price that is declared by the service provider. We now formulate the objective function for an individual user who wishes to maximize his benefits from network services in contract period i.

Suppose that there are a total of M users. The utility function of user j (where $1 \leq j \leq M$) in contract period i is given by U_{ij}. Let the number of packets sent by user j in contract period i be x_{ij}. Then the single period utility maximization problem for user j in contract period i can be stated as :

$$\max_{x_{ij}} [U_{ij}(x_{ij}) - p_i x_{ij}]$$

subject to:

$$x_{ij} \geq 0 \tag{1}$$

This leads to the first order condition :

$$\frac{dU_{ij}(x_{ij})}{dx_{ij}} = p_i \tag{2}$$

The expression on the left of Equation 2 is also called the reservation price. Thus, we see that in order to maximize its individual benefits from the network, each user continues to use the network resources until the actual price equals his reservation price. In order to make the goals of individual users concur with the premium service level agreements, the price in contract period i should be set such that :

$$\sum_{j=1}^{M} x_{ij}^{\star} = C_i \tag{3}$$

where, $x_{ij}^{\star}$ is the solution of the maximization problem (stated in (1)) that satisfies Equation 2.

The solution obtained from Equation 3 is the ideal solution to the problem of allocating network resources, as it not only ensures that users with highest utilities get the premium network services (Pareto efficiency or economic efficiency), but also ensures that the network resources are fully utilized (network efficiency). However, it is hard to determine the individual user utility functions and they often change with time. The problem is that if the price advertised (determined by an alternate means) is higher than the optimum price that satisfies Equation 3, the capacity contracted by users will be less than the available capacity. This will decrease the resource utilization and *cause an inefficient allocation of resources.* If the price is lower than the one obtained from Equation 3, total user demand will exceed capacity and it will be hard to maintain premium service conditions.

To operate in this scenario when the information required for ideal solution is not available, we take the approach of having a queue at the edge (which acts as buffer for demand) and use an adaptive pricing scheme to control the length of the queue. Clearly, the longer the permissible queue, the higher the average utilization, but greater the delay each packet encounters. Thus, there is a trade-off. Some ISPs would like to offer better service even if that means lower utilizations, while others would like to maintain higher utilizations (bigger customer base) even if this implies increase in delays packets encounter.

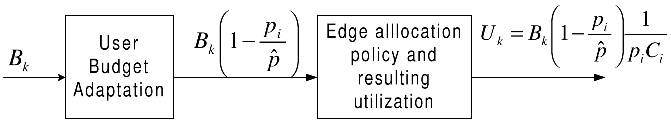


Fig. 2. Model of user adaptation and allocation for each of the k cases.

IV. A System Parameter for Price Discovery

The state of the network system at the beginning of contracting period i is completely specified by the queue length carried over from previous period, q_{i-1} and estimated capacity, C_i. q_{i-1} represents the cumulative difference in demand and capacity for premium services since the last period in which queue length was equal to zero. Thus, it is natural to expect the parameter q_{i-1} to be chosen for congestion-sensitive pricing. By coordinating edge routers, the edge-to-edge capacity C_i can be estimated. So, the two parameters q_{i-1} and C_i are known at the beginning of a contract period in an edge-to-edge framework and thus can be used in a pragmatic scheme for pricing. Their ratio is particularly interesting as it indicates the severity of congestion. Thus, the ratio of queue length to capacity is intuitively an attractive parameter for use in congestion-based Internet pricing.

To further investigate the behavior of the system and to confirm the validity of this intuitive parameter, we set-up a single period optimization problem and solved it using optimizer functions built in Matlab. We then investigated the relationship between the optimal prices obtained for a period i and the state at the end of period $i-1$ (which is known when price for period i is to be determined). In particular, we investigated the relationship of the ratio of edge queue length and capacity with optimal prices.

This optimization and hypothesis experiment can logically be viewed as the following two steps :

- In step 1, we assume that we know every bit of information required to find the optimum price (in the typical scenario formulated).
- In step 2, we examine the relationship between the optimum prices of step 1 and our intuitive parameter for capturing congestion conditions, q_{i-1}/C_i. The parameter is known at the time of pricing in an edge-to-edge framework.

The idea is to find optimum price assuming we knew everything, and then see how good we could do knowing what we actually know.

The objective in the formulated problem is to maximize the utilization of network resources in a period i, given the distribution of the total budget of all users [1], distribution of available network resources available for contracting in period i, maximum permissible queue length (expected to be determined by service level agreement specifications) and queue length at the end of period $i-1$.

In the optimization model, we take k=1000 sample size for each random component (i.e. observations) in a single period problem. Further, to check the sensitivity of the regression results obtained to the assumed distributions, we do the hypothesis testing for two different distributions of user budgets (one with large varying loads). The process of budget adaptation and eventual utilization is explained in Figure 2.

User adaptation to varying demand is captured by changing the budget according to the following simple linear rule:

$$B_{k,spent} = \max\left\{0, B_k\left(1-\frac{p_i}{\hat{p}}\right)\right\} \tag{4}$$

[1] For tractability reasons, all users are assumed to be in the same class in the sense that they have the same reservation price. The total budget of all users was used in the formulation as it can be estimated with better confidence as compared to individual user budgets.

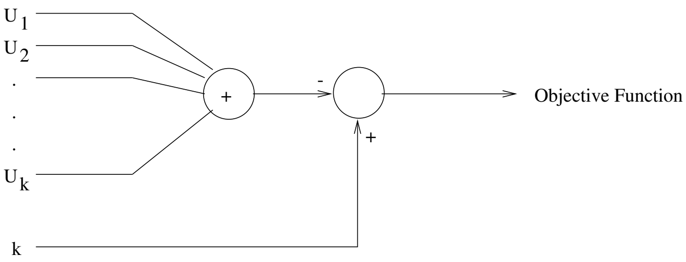


Fig. 3. Combining the k cases to make an objective function to be minimized.

where $B_{k,spent}$ is the part of budget actually spent, $\hat{p}$ is the reservation price for the individual user, p_i is the price in period i and B_k is user's budget for the observation k (period i).

We define objective function as simply the sum of fraction of network resources not utilized in each of the k samples for period i. The constraint simply states that the service level agreements require the queue length not to exceed q_{max}. As the `fmincon` (optimizer in Matlab for multi-dimensional constrained optimization) optimizer takes a function to find its minima, the individual utilizations are combined to create an objective function as shown in Figure 3. Given price p_i, the customer contracts for $B_{k,spent}/p_i$ amount of capacity. So, the single period optimization problem solved using `fmincon` can be formulated as to minimize aggregate idle capacity over a number of periods:

$$\min_{p_i} \sum_{k=1}^{1000}\left[1-\frac{B_{k,spent}}{p_i C_k}\right]$$

subject to:

$$q_i = q_{i-1} + \frac{B_{k,spent}}{p_i} - C_k \le q_{max} \qquad \forall k \in [1,1000] \tag{5}$$

where k is the number of observations, C_k is the estimated capacity at observation k, q_{max} is the maximum permissible queue length, and B_k is the user's budget in observation k. For C_k, we use Normal distribution with parameters $\mu = 98$ and $\sigma = 2$ truncated in the range $[96,100]$, i.e. $C_k \sim N(98,2)$ truncated in the range $[96,100]$. For B_k, we use Uniform distribution. In order to examine the sensitivity to the total budget distribution, we examine two cases: $B_k \sim U(20,50)$ and $B_k \sim U(30,150)$. Sample size for the budget and capacity is taken to be $k = 1000$. We set q_{max} to 50.

The primary motivation of the above numerical optimization exercise is to observe the behavior of optimal price if all the needed information (like budget and user adaptation model) was available. Thus, any reasonable model for user adaptation is good for the task at hand. After optimization, we try to see how well we have done at predicting the optimal price from the information that we know at the beginning of a contracting period (like q_{i-1} and C_i) without using the assumed parameters in problem formulation.

A. Regression Analysis for Budget

In this section, we present regression analysis when the Budget distribution is U[20,50]. We present the results of optimization for different values of q_{i-1} in Table I. In table, $C_{i,mean}$ is the average of k (=1000, in this case) random instances of capacity (drawn from the chosen distribution) for period i.

Regression results of best fit linear model between $q_{i-1}/C_{i,mean}$ (independent variable) and $p_i^\star$ are presented in Table II and III. The regression equation is:

$$p_i^\star = 0.281 + 0.223\frac{q_{i-1}}{C_{i,mean}} \tag{6}$$

TABLE III

ANALYSIS OF VARIANCE: $p_i^\star$ VERSUS $\frac{q_{i-1}}{C_{i,mean}}$

Source	Degrees of Freedom	Sum of Squares	Mean Sum of Squares	F	P
Regression	1	0.0143	0.014296	905.92	0.00
Residual Error	9	0.0001	0.000016		
Total	10	0.0144			

TABLE I

OPTIMIZATION RESULTS : $p_i^\star$ FOR VARIOUS q_{i-1}.

q_{i-1}	$q_{i,mean}$	$p_i^\star$	$q_{i-1}/C_{i,mean}$
0	13.11	0.286	0.00
5	14.28	0.296	0.51
10	14.15	0.304	0.10
15	15.02	0.314	0.15
20	16.73	0.324	0.20
25	16.97	0.332	0.26
30	17.72	0.346	0.31
35	17.52	0.360	0.36
40	18.50	0.371	0.41
45	19.92	0.388	0.46
50	20.16	0.401	0.51

TABLE II

REGRESSION ANALYSIS : $p_i^\star$ VERSUS $\frac{q_{i-1}}{C_{i,mean}}$

Predictor	Coefficient	SE Coefficient	T	P
Constant	0.281	0.0022	125.54	0
$\frac{q_{i-1}}{C_{i,mean}}$	0.223	0.0074	30.10	0

Similarly, we performed the same regression analysis when budget distribution is U[30,150] and got the following regression equation:

$$p_i^\star = 0.822 + 0.634\frac{q_{i-1}}{C_{i,mean}} \tag{7}$$

So we observe that there is a strong association between $\frac{q_{i-1}}{C_{i,mean}}$ and $p_i^\star$, and the association does not weaken considerably with large variations in budget. This implies that we could use the predictor variable $\frac{q_{i-1}}{C_{i,mean}}$ to determine p_i in adaptive pricing. Further, there is no need for exploring more complicated models as the ratio of the regression sum of squares to the total sum of squares is pretty high for the linear model (99%). Notice that, the predictor variable $\frac{q_{i-1}}{C_{i,mean}}$ does not require any knowledge of total user budgets which were assumed to be known during the simulation. Further, the predictor variable for a contract period can be estimated at the beginning of the period, as queue length at the end of previous period is known and the estimated capacity available in a period is known at the beginning of the period.

V. SOME ADAPTIVE PRICING SCHEMES WITHIN PRICE DISCOVERY FRAMEWORK

In this section, we use the parameter identified in the previous section, to define four adaptive pricing schemes and evaluate them. In order to evaluate their performance, we use the response of the schemes to a sudden increase in load. It is a requirement from the pricing scheme that it be able to control the queue length in desirable limits (determined by service level agreements) while maintaining high utilization.

Typically an ISP providing premium services would like to control the edge queue in a range $[q_l, q_h]$. The precise selection of this range would be determined by the trade-off that the ISP and the users are ready to make between delays and utilization (service level agreement specifications). For instance, higher levels of network utilization and lower prices are possible if higher edge queues are maintained. This comes at the cost of higher delays because of the larger edge queue.

The main idea is to increase the price if the network is getting congested, and to decrease the price when the congestion alleviates. We now define four possible pricing schemes some of which use the parameter identified in the previous section.

1. *Proportional Increase and Proportional Decrease (PIPD):* In this scheme, the price is increased or decreased proportional to the difference between current queue length and the lower or higher queue threshold. Let β_1 and β_2 be two positive constants, then we can write function for calculating price for contract period $i+1$ as follows:

$$p_{i+1} = \begin{cases} p_i + \beta_1\frac{(q_i - q_h)}{C_{i+1}}, & q_i > q_h \\ p_i - \beta_2\frac{(q_l - q_i)}{C_{i+1}}, & q_i < q_l \\ p_i, & otherwise \end{cases}$$

2. *Proportional Increase and Additive Decrease (PIAD):* In this scheme, the price is increased proportional to the difference between current queue length and the higher queue threshold, and decreased by a constant when the queue length falls below lower threshold. Let α_1 and α_2 be two positive constants, then we can write function for calculating price for contract period $i+1$ as follows:

$$p_{i+1} = \begin{cases} p_i + \alpha_1\frac{(q_i - q_h)}{C_{i+1}}, & q_i > q_h \\ p_i - \alpha_2, & q_i < q_l \\ p_i, & otherwise \end{cases}$$

3. *Additive Increase and Additive Decrease (AIAD):* In this scheme, the price is increased or decreased by a constant (γ_1 and γ_2 respectively) when the queue length exceeds higher threshold or goes below lower threshold. Let γ_1 and γ_2 be two positive constants, then we can write function for calculating price for contract period $i+1$ as follows:

$$p_{i+1} = \begin{cases} p_i + \gamma_1, & q_i > q_h \\ p_i - \gamma_2, & q_i < q_l \\ p_i, & otherwise \end{cases}$$

4. *Additive Increase and Proportional Decrease (AIPD):* In this scheme, the price is increased by a constant when the edge queue length goes higher than higher queue threshold, and decreased proportional to the difference between current queue length and the lower queue threshold. Let ζ_1 and ζ_2 be two positive constants, then we can write function for calculating price for contract period $i+1$ as follows:

$$p_{i+1} = \begin{cases} p_i + \zeta_1, & q_i > q_h \\ p_i - \zeta_2\frac{(q_l - q_i)}{C_{i+1}}, & q_i < q_l \\ p_i, & otherwise \end{cases}$$

The schemes PIPD, PIAD, and AIPD use the parameter (i.e. q_i/C_{i+1}) we identified in the previous section. We will now evaluate the above four pricing schemes and show that PIAD performs best. Next, we outline the user model and pricing scenario used for comparison of schemes.

TABLE IV

PARAMETERS USED FOR COMPARING THE FOUR SCHEMES

Scheme	Parameters	$\overline{q}$	$\overline{u}$	$\overline{p}$
PIPD	$\beta_1 = 3; \beta_2 = 3$	19.77	98.92%	0.612
PIAD	$\alpha_1 = 3; \alpha_2 = 0.3$	20.65	99.56%	0.602
AIAD	$\gamma_1 = 0.15; \gamma_2 = 0.1$	19.36	99.01%	0.609
AIPD	$\zeta_1 = 0.1; \zeta_2 = 1$	20.61	99.12%	0.604

TABLE V

COMPARISON OF THE PERFORMANCE OF FOUR SCHEMES

Scheme (Parameters)	$\overline{q}$	$\overline{u}$	$\overline{p}$	q_{max}
PIPD($\beta_1 = 3; \beta_2 = 3$)	19.45	91.39%	0.99	159
PIAD($\alpha_1 = 3; \alpha_2 = 0.3$)	19.57	88.97%	1.03	158
AIAD($\gamma_1 = 0.15; \gamma_2 = 0.1$)	34.72	94.68%	0.86	456
AIPD($\zeta_1 = 0.1; \zeta_2 = 1$)	47.79	96.82%	0.84	506

Let, $\hat{p}$ be the reservation price. We know that at the reservation price the user demand for network service is zero. Using this fact, we use the following simple but reasonable user adaptation function for evaluation of pricing schemes:

$$X_{i,p=p_i} = X_{p=0}\frac{\hat{p} - p_j}{\hat{p}} \tag{8}$$

In the above equation, $X_{p=0}$ is the demand when price is equal to zero and $X_{i,p=p_i}$ is the demand when price is p_i.

For performance evaluation of pricing schemes, we use the following system parameter values: $(q_h = 25, q_l = 15, \hat{p} = 2, X_{p=0} = 140)$ for all schemes. Other algorithm specific parameters are specified in Table IV. Notice that the parameters are selected such that utilization is almost the same for each algorithm as the starting point for later comparisons.

For comparing the pricing schemes described above, we first normalize the different pricing schemes by choosing a set of parameters (tunables), such that the average queue length (a measure of the average quality-of-service) and average utilization over a number of contracting periods is approximately equal. Table IV gives the parameter values for the four schemes and the values of the average queue length $\overline{q}$, average utilization $\overline{u}$ and average price $\overline{p}$ over 200 contracting periods. Afterwords, we subject all the schemes to a step increase in load and observe how well the schemes can control the edge queue while maintaining high levels of utilization. The load is:

$$\Delta X_{p=0} = 200[u_t(50) - u_t(100)] \tag{9}$$

where $\Delta X_{p=0}$ is the change in base demand and $u_t(x)$ is defined as:

$$u_t(x) = \begin{cases} 1, & t \geq x \\ 0, & otherwise \end{cases}$$

Average queue length, utilization, price and the maximum queue length for all the four schemes after subjecting them to the step increase in load are given in Table V.

The result of subjecting PIPD to the step load is shown in Figure 4-a. We see that the pricing scheme responds quickly by jacking up the price and controls the max queue length at 159. Moreover, the average utilization over the 200 period interval is 91.4%, which is good for a step load of 200 (more than 120% of base demand before loading).

The result of subjecting PIAD to the step load is in Figure4-b. We see that the pricing scheme responds quickly to control the demand. One difference from the PIPD case is that price variations are less. This decrease in price variation is achieved by observing that the pricing scheme can decrease the price additively rather than proportionally. By doing so, we gain lesser variations in price at the cost of some utilization (see Table V).

The result of subjecting AIPD and AIAD to the step load is in Figure 4-c and 4-d. These schemes lack the responsiveness required to control congestions like the one simulated. This is indicated by the fact that the maximum queue grows to as high as 456 and 506 respectively.

It is important that the pricing function does not allow the queues to grow far beyond the higher threshold, q_h. For this reason and to be able to control rising demands in the periods of congestion by providing the right pricing feedback, we should increase price multiplicatively proportional to the difference between current queue length and the higher queue threshold, q_h (like PIAD and PIPD). Especially in PIAD, when the queue length falls below the lower threshold, we start decreasing the price but this time we achieve less price variation at the cost of slight decrease in utilization. Additionally, additive decrease was chosen in order to avoid the possibility of system oscillating between high edge queue lengths and edge queue underflow. Thus, PIAD helps achieving a balance between faster congestion control and price stability.

Adaptive schemes are attractive in the sense that they do not make any assumptions about the user behavior. But, they tend to have stability problems and the idea behind including additive price increase is to alleviate some of these instability issues. In the next section, we describe the PIAD algorithm in greater detail and derive the condition for which the scheme ensures stability.

VI. PIAD WITH TUNABLES

In this section, we find the condition for PIAD tunables that ensure the system remains stable i.e. edge queues do not grow unboundedly.

Let us assume that the highest reservation price that a user has is $\hat{p}_{max}$. Let x_{ij} be the capacity contracted by user j in period i. By definition of $\hat{p}$, it follows that:

$$p_i = \hat{p}_{max} \Rightarrow \sum_{1}^{M} x_{ij} = 0$$

The above statement implies that the entire capacity is being used to drain the edge queue. Also, the following statement holds:

$$\forall \hat{p}_{max} > 0, \exists \alpha_1 \text{ such that } \hat{p}_{max} \leq \alpha_1[q_{max} - q_h]$$

where, q_{max} is the edge buffer size (capacity of the edge queue; its total size, not q_h). The above statement, gives us the following condition for ensuring stability:

$$\frac{\hat{p}_{max}}{q_{max} - q_h} \leq \alpha_1 \tag{10}$$

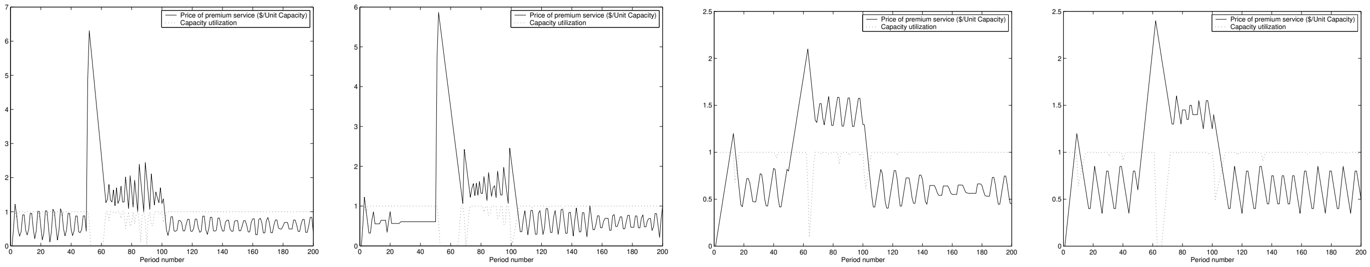
Thus, we conclude that as long as α_1 satisfies the above condition, PIAD pricing scheme will guarantee stability. An alternate manner of guaranteeing a similar effect is to provision for very long edge queue such that chances of it being fully used are rare.

In the next section we develop a user model for analysis of the PIAD adaptive pricing scheme and explore some properties of the tunables and what parameters of the system they help control.

VII. A SIMPLE USER MODEL AND EVALUATION OF PIAD

In general, the user model of adaptation to prices can be depicted as in Figure 5. In the figure, $X_{i,p}$ is the user demand for network services when price for them is p per unit. This is determined by the user adaptation model which takes the demand when price is zero, $X_{i,p=0}$ and the current price for network services p_i as inputs. The latter of course is determined by the PIAD pricing scheme defined in the previous section based on the system state variables at the beginning of period i. The state of the system at the beginning of a contracting period can be completely defined by the queue carried over from the last period q_{i-1} and capacity in period i, C_i (Figure 1). The pricing scheme can be looked at as a feedback mechanism that gives input to the user adaptation block in the figure based on the network state (C_i, q_{i-1}) at the beginning of the period.

Notice that the user model that we propose here will be used for evaluating the performance of the PIAD pricing scheme proposed in this paper and the impact that PIAD tunables have on the system performance only. The adaptive pricing scheme is not dependent on the specific form of user adaptation curves.



(a) PIPD (b) PIAD (c) AIPD (d) AIAD

Fig. 4. Response of pricing schemes to a step increase in demand.

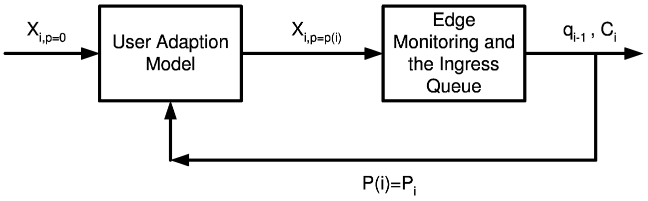


Fig. 5. General model of user adaptation to price

Let, $\hat{p}$ be the price at which the user becomes indifferent to using the service or keeping the money, also called the reservation price. We know that at the reservation price the user demand for network service is zero. Using this fact, we use the following simple, but reasonable, user adaptation function for performance evaluation of the PIAD pricing scheme:

$$X_{i,p=p_i} = X_{p=0}\frac{\hat{p}-p_j}{\hat{p}} \tag{11}$$

For performance evaluation of PIAD pricing scheme (see Figure 6-a), we use the following system parameters: $(\alpha_1 = 3, \alpha_2 = 0.1, q_h = 25, q_l = 15, \hat{p} = 2)$. Unless otherwise specified, $X_{p=0} = 140$. Price in all the figures is in units of ($/Unit Capacity).

Notice that, each set of algorithm tunables $(\alpha_1, \alpha_2, q_l, q_h)$ correspond to different positions on three trade-offs (dependent on ISP-Customer performance metrics). The three trade-offs are:

- high utilization versus smaller edge queues (better service)
- price variability versus system responsiveness
- price variability versus quality of service variation

Now, we present the results obtained from a set of controlled simulations to observe the impact that the pricing algorithm tunables have on the overall system performance.

A. *Effect of* α_1

By increasing α_1 the response to congestion $(q_i \geq q_h)$ can be made more aggressive. This also implies that on average the edge queue length will be smaller and utilization will be lower for high α_1. Comparing Figure 6-a with 6-b we see that an increase in α_1 from 3 to 4 results in faster variation in price. This is because the system has been made more responsive to queue length increases. High α_1 is a good idea if load on system can ramp up rapidly as the system is very responsive, but this comes at the cost of increased variation in price and a possible lower utilization. Thus, keeping α_1 as low as possible satisfying the service level agreements and keeping in mind the expected variability in the load is recommended (α_1 should be high enough to give the users enough incentive to restrain their demand when edge queue starts building up). α_1 of course should satisfy the stability condition of Equation 10.

B. *Effect of* α_2

By decreasing α_2, the range in which price varies between *queue drain* (demand control) and *queue build up* (demand probe) cycles can be decreased. This of course comes at the cost of compromising the capability to slash prices quickly if demand becomes very less. This is seen in Figure 6-c from period 4 to 37 when utilization drops significantly and prices are slashed at a rate slower than that in Figure 6-a. By increasing α_2, we see that the system becomes more responsive to demand changes. This also causes a increased fluctuation in the prices. (Figure 6-d)

It is important to realize that, demand variation and quality of service requirements are typical characteristics of the user preferences. The pricing scheme proposed here provides the ISP with the tunables to balance user requirements and their goals, like high utilization. These user preferences are going to differ from ISP to ISP (depending on their client base) and thus there will be need for selecting different values of the tunables. *A way of looking at this is that each value set of the tunables represents some trade-offs*. The precise trade-offs made would depend on the ISP and user requirements (the weight given to different user and ISP performance metrics).

C. *Effect of* q_l

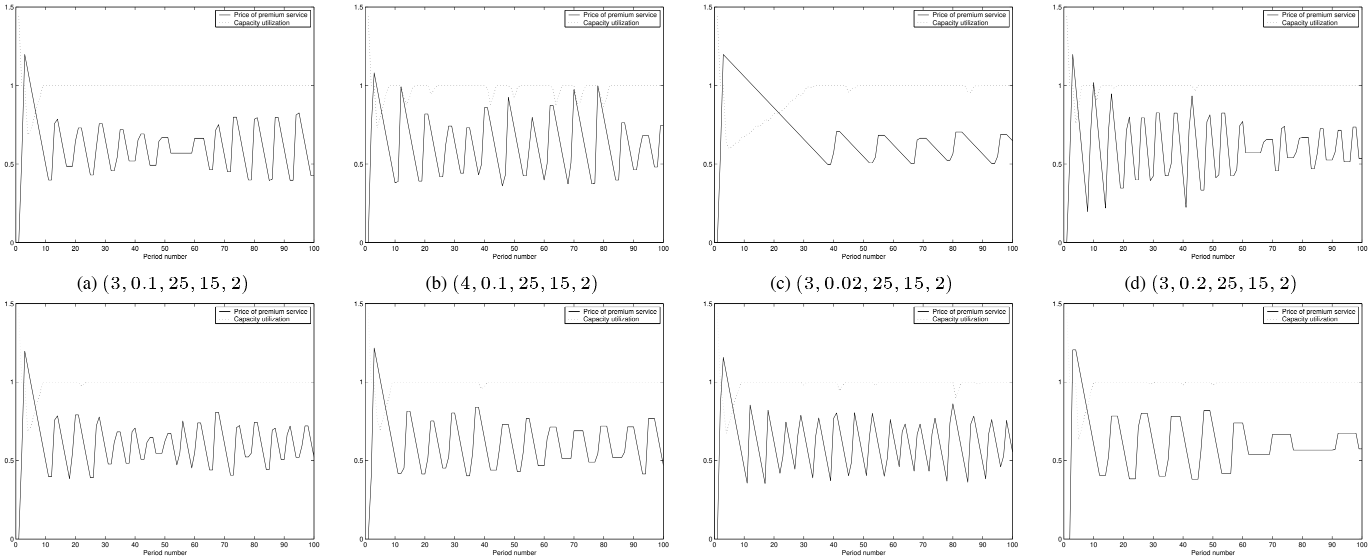
Having a higher q_l would increase the utilization of network resources in the face of demand variability. This will happen, at the cost of maintaining higher queue lengths at the edge. Thus, the trade-off here is between, very good service (low q_l) and higher utilization (higher q_l). The precise value chosen for q_l would depend on the preferences of the participants(users and ISPs). For system parameters of Figure 6-e ($q_l = 20$ in 6-e and $q_l = 15$ in 6-a), the average length of the queue on the edge is 19.22 as opposed to 17.52 for Figure 6-a.

D. *Effect of* q_h

Increasing q_h (see Figure 6-f) results in increase in the average queue length at the edge as the queue drain or demand control part of the cycle now starts later than a setting with lower q_h. There is an increase in utilization at the cost of the quality of service provided.For system parameters of Figure 6-f, the average length of the queue on the edge is 19.08 as opposed to 17.52 for Figure 6-a. Notice that this action is equivalent to resetting the definition of congestion i.e. actions taken to control congestion are now triggered at a different indicator level.

E. *Effect of Distance Between* q_l *and* q_h

In Figure 6-g, we set q_l and q_h equal to each other. As is evident this causes very frequent variations in price and the average magnitude of these variations is higher than that of Figure 6-h. q_l cannot be too low to avoid underflow and thus the resulting lower utilization levels.q_h cannot be too high as service level agreements need to be met. Thus, there are limitations to how much we can gain on this price smoothing. Moreover, this price smoothing comes from accepting large variabilities in queue length which would imply large variability in service quality. Thus, the need to choose a distance between q_l and q_h that



(a) $(3, 0.1, 25, 15, 2)$ (b) $(4, 0.1, 25, 15, 2)$ (c) $(3, 0.02, 25, 15, 2)$ (d) $(3, 0.2, 25, 15, 2)$

(e) $(3, 0.1, 25, 20, 2)$ (f) $(3, 0.1, 30, 15, 2)$ (g) $(3, 0.1, 15, 15, 2)$ (h) $(3, 0.1, 45, 15, 2)$

Fig. 6. Price and utilization for system parameters tuple $(\alpha_1, \alpha_2, q_h, q_l, \hat{p})$.

represents the needs of users and objectives of the ISP (a trade - off that meets the demands of participants the best).

F. Robustness Under Different Loads

In Figure 7-a, we subject the system to an extra user load equivalent to increment of demand at price zero, $X_{i,p=0}$ by 50 units in period numbers [50,200]. The extra load can be specified as:

$$\Delta X_{p=0} = 50u_t(50) \tag{12}$$

where $u_t(50)$ is defined as:

$$u_t(50) = \begin{cases} 1, & t \geq 50 \\ u_t(50) = 0, & otherwise \end{cases}$$

In Figure 7-a we see that increasing the demand suddenly in period 50 for premium service is responded to by increasing the price so that users have the correct incentives for regulating their demands. The average utilization in Figure 7-a is 0.980 as compared to 0.994 for Figure 6-a. The scheme reacts in a desirable manner, keeping utilization high while allocating resources efficiently.

In Figure 7-b, we increase the demand at price zero from 140 to 190 in period $i \in [50, 100]$. We see that the prices, go up during the period of higher demand and come down when the extra demand subsides. High levels of utilization are maintained throughout except for during transition because of the unanticipated change in demand involved. Being able to maintain high levels of utilization while satisfying the QoS requirements is an important characteristic of a good pricing scheme.

In Figure 7-c, the system is subjected to an increase in demand similar in form to that of Figure 7-a. The only difference is that this time the demand disturbance is higher in magnitude (double). As the system was not ready (tuned) for such a spike in demand, we see that there is a short period of very low utilization when the extra demand is introduced.

In Figure 7-d, we look at the same demand spike as in Figure 7-c. The only difference is, this time, one of the tunable parameters, α_2 has been tuned to permit for more aggressive demand probing. This results in much lower penalties in terms of lower utilization during transition. The average utilization over 200 periods for Figure 7-c is 0.941 and that for Figure 7-d is 0.965.

In summary, we see that there is a range in which the tunables work fine (this is the range for which they are set). For instance, the tunables in these simulations could handle demand spike of 50 units without a lot of deterioration of performance in the transition phase. Demand spikes significantly more than this level resulted in lower utilizations in the transient phase. Nevertheless, in the steady state, satisfactory levels of utilization were achieved irrespective of the magnitude of demand spike. Further, tunables α_1 and α_2 should be chosen keeping in mind not only the trade-offs outlined earlier, but also the variations in demand expected.

G. Users with Different Demand and Reservation Prices

In this sub-section, we will simulate different user scenarios and see how PIAD pricing scheme impacts resource allocation between user classes having different demands and utility functions.

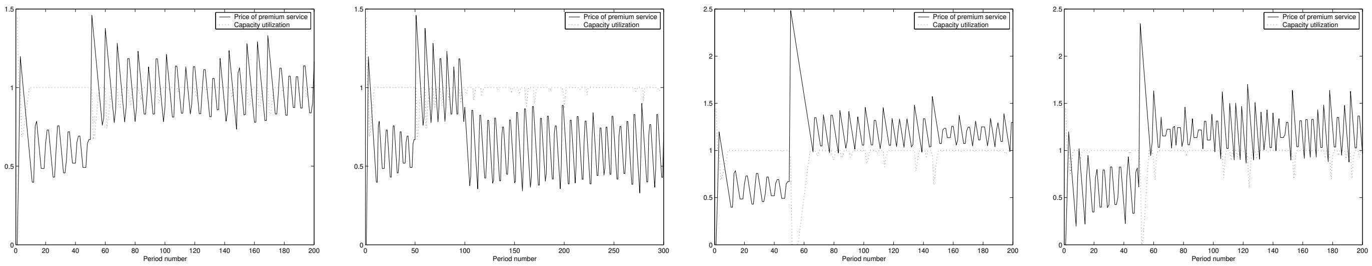
In Figure 8-a, we see that User 1, who has higher reservation price and demand is allocated a higher fraction of network resources. The average of resources allocated to User 1 is 58.07 as compared to 38.06 for User 2. The ratio of these allocations is 1.78, which is higher than the ratio of the reservation prices of the two users $\frac{2}{1.5} = 1.33$. This is because User 1 has a greater base demand (demand when price is zero) than User 2. Allocating resources according to the base demand and the reservation price, not just one of them is an important desirable characteristic of a good pricing scheme.

In Figure 8-b, we examine the effect of decreasing the reservation price of User 2 on the allocation to two users. The average allocations are found to be 62.88 and 34.32 respectively. The ratio of allocations for this case is 1.83 which is higher than that of 8-a, which is expected.

In Figure 8-c, we examine the effect of change in base demand to the scenario of Figure 8-a. The average allocations over the 100 contract periods observed is 74.31 for User 1 and 19.70 for User 2. This indicates that the base demand (demand when $p = 0$) plays a significant role in capacity allocation which is desirable.

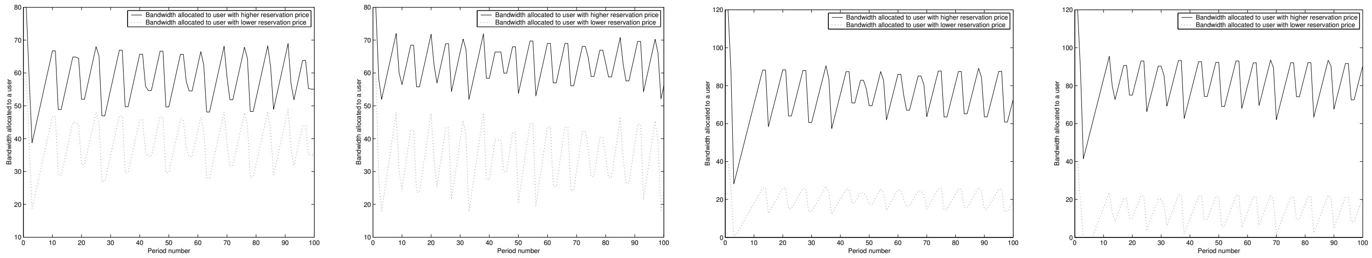
In Figure 8-d, we examine the scenario of Figure 8-c when reservation prices of the two users are changed. The average allocation to the users now becomes 80.77 for User 1 and 14.10 for User 2.

Thus, we see that the scheme can allocate resources to a wide variety of users based on their reservation price and base demand. Notice that



(a) $(0.1, 50u_t(50))$ (b) $(0.1, 50[u_t(50) - u_t(100)])$ (c) $(0.1, 100u_t(50))$ (d) $(0.2, 100u_t(50))$

Fig. 7. Price and utilization for $\alpha_1 = 3$, $X_{p=0} = 140$, $q_h = 25$, $q_l = 15$, $\hat{p} = 2$ and system parameters tuple $(\alpha_2, \Delta X_{p=0})$.



(a) $(2, 80, 1.5, 60)$ (b) $(2, 80, 1, 60)$ (c) $(2, 120, 1.5, 40)$ (d) $(2, 120, 1, 40)$

Fig. 8. Price and utilization for $\alpha_1 = 3$, $\alpha_2 = 0.1$, $q_h = 25$, $q_l = 15$ and system parameters tuple $(\hat{p}_1, X_{1,p=0}, \hat{p}_2, X_{2,p=0})$.

User 2 is able to increase its resource share by increasing its reservation price from 1 (Figure 8-d) to 1.5 (Figure 8-c).

VIII. Summary and Discussions

In this paper, we identified a parameter for congestion-sensitive pricing in a contracting framework for edge-to-edge premium services. Based on this parameter, we developed Price Discovery framework, which can be implemented in diff-serv networks. Within the Price Discovery framework, we formulated four adaptive pricing schemes (i.e. PIPD, PIAD, AIPD, AIAD) to determine prices at network edges. We then demonstrated that PIAD is the best scheme in performance.

PIAD pricing scheme is easily deployable in edge-to-edge framework as it only needs q_{i-1} (the length of queue at the edge in period $i-1$) and C_i (estimated edge-to-edge capacity for period i) to calculate the price in period i. By coordinating the edge routers, both these parameters are available in diff-serv environment. We also derived the stability condition for PIAD parameters. The PIAD pricing scheme does not assume anything about the user behavior. It is adaptive in its nature and has tunable parameters that can be set to provide a *range of better than best effort services* corresponding to suitable trade-offs. The three important trade-offs are: high utilization versus smaller edge queues; price variability versus system responsiveness; and price variability versus quality-of-service variation. We examined the impact each PIAD tunable has on these trade-offs. For a given ISP-customer setting, the PIAD tunables should be set according to the user QoS requirements (e.g. low delay) and ISP metrics (e.g. high utilization). We also examined PIAD behavior under different congestion conditions of varying severity. In general, PIAD can control congestion and ensure high utilization. During transient phases the utilization drops, but even that can be rectified by proper selection of the parameters. We also observed that resource allocations among users, made under different user utility and base demand scenarios, are well-behaved.

Future work would involve exploring algorithms for changing PIAD parameters by capturing the information about changes in base demands. An interesting parameter that can be used for changing the tunables is the *rate of change of edge queue*, since it is a good measure of change in load on the network. Also, in this paper, we focused on the problem of pricing a single type (i.e. class) of premium service by controlling the edge queue corresponding to that service type. The framework can be readily extended to manage pricing multiple types of premium services which would be priced by PIAD pricing scheme with different values of tunables. This is left for future work.

Acknowledgments

Authors would like to thank Anuj Goel and Ritesh Pradhan for their helpful comments and discussions.

References

[1] S. H. Low and D. E. Lapsley, "Optimization flow control – I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–875, 1999.

[2] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, pp. 237–252, 1998.

[3] J. K. MacKie-Mason and H. R. Varian, *Pricing the Internet*, Kahin, Brian and Keller, James, 1993.

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

[5] A. Gupta, D. O. Stahl, and A. B. Whinston, *Priority pricing of Integrated Services networks*, Eds McKnight and Bailey, MIT Press, 1997.

[6] N. Semret, R. R.-F. Liao, A. T. Campbell, and A. A. Lazar, "Pricing, provisioning and peering: Dynamic markets for differentiated Internet services and implications for network interconnections," *IEEE Journal on Selected Areas of Communications*, vol. 18, no. 12, pp. 2499–2513, 2000.

[7] X. Wang and H. Schulzrinne, "An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications," *IEEE Journal on Selected Areas of Communications*, vol. 18, no. 12, pp. 2514–2529, 2000.

[8] R. Singh, M. Yuksel, S. Kalyanaraman, and T. Ravichandran, "A comparative evaluation of Internet pricing models: Smart market and dynamic capacity contracting," in *Proceedings of Workshop on Information Technologies and Systems (WITS)*, 2000.

[9] A. M. Odlyzko, "A modest proposal for preventing Internet congestion," Tech. Rep., AT & T Research Lab, 1997.

[10] D. Clark, *Internet cost allocation and pricing*, Eds McKnight and Bailey, MIT Press, 1997.

[11] S. Shenker, D. Clark, D. Estrin, and S. Herzog, "Pricing in computer networks: Reshaping the research agenda," *Telecommunications Policy*, vol. 10, no. 3, pp. 183–201, 1996.

[12] D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in *Proceedings of SIGCOMM*, 1990.

[13] H. R. Varian, *Intermediate Microeconomics: A Modern Approach*, W. W. Norton and Company, 1999.

[14] S. Blake et. al, "An architecture for Differentiated Services," *IETF RFC 2475*, December 1998.

[15] D. Harrison, S. Kalyanaraman, and S. Ramakrishnan, "Overlay bandwidth services: Basic framework and edge-to-edge closed-loop building block," Poster in the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), 2001.

[16] S. S. Chiu and J. P. Crametz, "Taking advantage of the emerging bandwidth market," Tech. Rep., 1999.