A Comparison of Auction and Flat Pricing for Differentiated Service Networks

Weilai Yang , *Henry Owen* and *Douglas M. Blough*
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250 USA

Abstract

In a network with quality of service (QoS) support, pricing is an effective means of dealing with congestion control and revenue generation. In the Internet, the needs of the customers and their applications are constantly evolving. An auction based algorithm is the best choice for this environment because it needs minimal a priori information. In this paper, we propose an auction based pricing algorithm which lets customers choose the price as well as the service required, and in which the service provider decides on the admission price threshold and the service level of the differentiated service provided. We then investigate the system's adaptive behavior by simulating it in various environments and situations.

1 Introduction

Over the years, quality of service (QoS) has attracted the attention of researchers. The DiffServ framework has been proposed to provide multiple QoS classes over IP networks. Within the framework, there are two types of Per-Hop-Behaviors (PHB): Expedited Forwarding (EF) and Assured Forwarding (AF). The DiffServ domain is a class-based network. Since the network supports multiple class services, a differentiated pricing strategy is necessary in place of the flat-rate pricing model.

Other than fix pricing, capacity-based pricing assumes that there is perfect a priori knowledge of demand and does an off-line calculation of optimal prices. We chose auction as our pricing method. Auction is a pricing approach with minimal a priori information needed [3]. It consists of clients submitting bids which specify the desired amount of resources and the price they are willing to pay, and the auctioneer allocating shares of the resources to the clients based on their bids.

We consider a scenario in which the auctioneer takes the customers' bids and returns the admission rates (which are also defined as thresholds) for both price and service offered for each class in a way that aims at maximizing the service provider's revenue. The customers bid for the base price, the price sensitivity and the required resource assignment. Auction based pricing schemes have typically been developed in a simplified model — the single-link network [3] [6]. The algorithm we present in this paper does away with this limitation.

Our contributions can be summarized as follows:
1) We model the transaction between the customers and the service provider. We analyze the service provider revenue optimization problem based on a bidding mechanism. The novelty is that customers have the freedom to propose their required service and the price. The service provider aims to maximize their profit by deciding which customers to service.
2) We show that auction models can work better than flat rate pricing both in a congestion or in a non congestion situation. The reasons are: a) In the auction model, when the network gets congested, the service provider can choose which flows to pass and which not, in favor of maximizing his revenue. b) When the network is over-provisioned, in the auction model, the service provider can reject lower class traffic if leaving them in gives the service provider lower revenue. Flat rating pricing does not have this facility.
3) The thresholds generated by the auctions also gives a guideline for future reference as to the condition for admitting new clients.

2 Problem Formulation

A price bid has two parts. The base price, and the price sensitivity coefficient. Besides the price, the clients need to specify the desired minimum bandwidth requirement. The base price is the amount the customer is willing to pay for the minimum required bandwidth. The sensitivity coefficient represents a customer's willingness to pay for extra bandwidth beyond his proposed minimum bandwidth requirement. For example, a customer wants to submit a video conference application. There is a minimum amount of resource required for this application to sustain communication. This corresponds to the minimum bandwidth requirement, along with which the client chooses a base price. If extra bandwidth is available, the customer has use for it because he can get better quality, thus in addition he specifies his valuation for extra bandwidth.

In order to keep the "extra assigned bandwidth" from eating up all resources, the growth of extra revenue should diminish as extra bandwidth grows. We choose a logarithmic increase. Therefore, a general revenue function is adopted as [5]:
$U_{ij} = U_{0ij} + W_{ij} \log \frac{X_{ij}}{L_{ij}}$
The auctions operate on sealed bids periodically. U_{ij} stands for the revenue from client i, who belongs to class j. U_{0ij} is client i's bid on U_0 — the base price. W_{ij} is client i's bid on the sensitivity coefficient. L_{ij} is his minimum required bandwidth. X_{ij} is the bandwidth allocated to customer i.

Customers bid for the base price, sensitivity coefficient and minimum required bandwidth. The objective is to maximize the service provider's revenue, subject to the restrictions imposed by the system's available resources. We introduce a superscript l to represent the value on the link l. A customer's bid is valid for all the links that his routing path crosses. The bandwidth capacity of link l, is denoted by C_l (l=1,2,..., L). Let l_{ij} represent the set of links that flow i of class j crosses in the network. A flow's bandwidth assignment also needs to be consistent on the routing path, thus we enforce $X_{ij}^{l_1} = X_{ij}^{l_2}$, where $l_1, l_2 \in l_{ij}$. The problem formulation is as follows.
Decision variables:

$$Z_{ij} = \begin{cases} 1; & \text{if client i is admitted to class j} \\ 0; & \text{otherwise} \end{cases}$$

U_{0ij} : final base price for client i in class j;
U_{0ij}^l : base price for client i in class j on link l;
U_{0j}^l : base price for class j on link l;
X_{ij} : final Bandwidth obtained by client i in class j;
X_{ij}^l : Bandwidth obtained client i in class j on link l;
L_{ij} : final minimum bandwidth for client i in class j;
L_{ij}^l : minimum bandwidth for client i in class j on link l;
L_j^l : minimum bandwidth for class j on link l;
W_{ij} : final price sensitivity for client i in class j;
W_{ij}^l : price sensitivity for client i in class j on link l;
W_j^l : price sensitivity for class j on link l;
Objective function:

$$max \sum_{j=1}^{2} \sum_{i} (U_{0ij} + W_{ij} \log \frac{X_{ij}}{L_{ij}}) * Z_{ij} \qquad (1)$$

Subject to:

$$\begin{cases} \sum_{j=1}^{2} \sum_i X_{ij}^l \le C_l; \\ X_{ij}^{l_1} = X_{ij}^{l_2}, l_1, l_2 \in l_{ij}; \\ X_{ij} \ge L_{ij} - (1 - Z_{ij}) * M; \\ U_{0j}^l \le U_{0ij}^l + (1 - Z_{ij}) * M; U_{0j}^l \ge 0; \\ W_j^l \le W_{ij}^l + (1 - Z_{ij}) * M; W_j^l \ge 0; \\ L_j^l \ge L_{ij}^l + (1 - Z_{ij}) * M; X_{ij} \ge X_j - (1 - Z_{ij}) * M; \\ X_{ij} \ge 0 + Z_{ij} * M; \\ \forall i, U_{0ij} \ge U_{0j}^l + (1 - Z_{ij}) * M; W_{ij} \ge W_j^l + (1 - Z_{ij}) * M; \\ L_{ij} \le L_{ij}^l + (1 - Z_{ij}) * M; l \in l_{ij} \end{cases}$$

Parameters:
C^l : link l's capacity
M : a very large positive number

Each customer proposes his desired values of U_{0ij}, W_{ij} and L_{ij}. We have to decide which flows to admit for each class in order to maximize the service provider's revenue. On each link and in the same class, the accepted customers will pay the same price — namely the lowest bid among those accepted. Another complication arises because a client might be across multiple links and two links might have different thresholds. We resolve this by setting the client's price to the highest among those thresholds. Since the threshold for each auction equates the lowest bid among the winners, this scheme still ensures that the customer pays at most his own bid. The solution to the optimization problem formulated above provides an optimal bandwidth assignment to each individual customer as well as maximizing the service provider's revenue.

3 Pricing Strategies

The problem formulated as in Section 2 is a non-convex integer and nonlinear problem. To the best of our knowledge, it does not fall into any existing optimization problem category that has an absolute global optimal solution. Due to the complexity of the problem, we are resorting to heuristic algorithms which can generate better results in terms of accuracy, time consumed and calculation delay as opposed to strict optimization methods.

3.1 Intra-link and Inter-link Solution

We approximate the optimal solution by splitting the problem in two steps. First we find an assignment for each individual link. Then we integrate the assignments into one multi-link assignment for the big picture. We first present the intra-link solution.

3.1.1 Intra-link auction

The optimal solution on one link is a special case for a network with only one link. All the assumptions that we made for the network case works for single link as well.

There is an optimal solution to this problem [6]. However, the search space and time complexity is high, so it is not appropriate to adopt the optimal solution here, especially when a large amount of customers are involved. We propose a simple heuristic algorithm to approximate the optimal results.

1. Golden search [4] is employed to decide each class' bandwidth allocation. Golden search is used here to find the optimal point. If each class' bandwidth allocation is known, step 2 is to find a good customer assignment and its corresponding revenue.

2. For each class, we perform the following operations. a) Build a queue of the bidding clients in decreasing order of their U_{0i}/L_i such that S_1:{U_{01}/L_1, U_{02}/L_2,...,U_{0n}/L_n}. b) Dequeue the first client, say flow i. If we can admit it, do so. Otherwise drop the client from the auction. By "can admit" we mean c/m $\geq$ max{L_1,..,L_m}, where c is the total bandwidth assigned to this class, m the number of admitted flows (including flow i) and max{L_1,..,L_m} is the maximum L value from those flows which have been admitted. Repeat this step until the queue is empty. c) Calculate the total revenue R_1. Let the accepted clients be denoted Acc_1:{i, j,...}. U_{0i}/L_i is the ratio of quality to price. The larger U_{0i}/L_i is, the more revenue the SP generates, thus the prioritizing of the clients with larger U_{0i}/L_i. However, there might be the case that a customer requires a huge amount of bandwidth, and is willing to pay a lot more as well — he has a high U_0 bid. In this case, U_{0i}/L_i might be low, which gives the clients a low priority. If this customer gets a good amount of bandwidth, he could generate very good revenue because of his high U_0, possibly more than the sum of the "small" clients. To address this, we create another queue to detect these valuable customers. d) Create a queue in decreasing order of U_{0i} such that S_2 :{U_{01},U_{02},..., U_{0n}}. e) Apply the same procedure as applied to S_1 to queue S_2. The total revenue R_2 is calculated at the end and we denote Acc_2 the accepted clients. f) We let simply take the better of these two results: $R = \max\{R_1,R_2\}$ and Acc the accepted clients corresponding to R. g) Now we look at the W value. W affects the secondary charge that the customer pays if he is allocated more bandwidth than his minimum requirement. When two customers bid the same U_0, the one with a higher W should have a better chance of winning the auction. So it is reasonable to use W bids as a complimentary judgement on the admission, in addition to the procedure we introduced above. Again we form a customer queue, this time of all the clients yet to be admitted, in decreasing order of W — S_3:{W_i, W_j,...,W_n}. h) Dequeue the W-queue and check if swapping the client with the last accepted client in Acc yields a better revenue. If it does, perform the swap. Repeat this step until half of the clients are left on the W-queue. W becomes smaller, it has less and less impact on the original accepted flow list and it is not necessary to adjust it any more. i) At this point we have the assignments for the accepted clients for each link and class. We let $X_{ij} = X_j$ for all i, be the bandwidth allocated to the link and class divided by the number of clients admitted. Now, it is time to put the pieces together.

3.1.2 Inter-link Adjustments

We need to perform inter-link adjustments to make the flows' assignment consistent throughout the network. We require $X_{ij}^{l_1} = X_{ij}^{l_2}$, where $l_1, l_2 \in l_{ij}$. 1) For each flow, set its assigned bandwidth to that of its bottleneck link — the link in the flow whose bandwidth per client is the lowest, i.e. the bandwidth for the relevant class in the link divided by the number of admitted clients. That can be expressed by :$X_{ij} = min\{X_{ij}^l\}$, where $l \in l_{ij}$. This will satisfy the consistency requirement. After this is done for all flows, there may be free bandwidth on some links for a certain class. Since the assignment was done independently on each link, it is very likely that the bottleneck links of various flows are spread out in the network, which yields a better chance for there to be extra bandwidth. In order to utilize some of this free bandwidth, the following adjustment is performed. 2) A set L is built to represent all the links which are currently under utilized. Start from the link k which has the minimum gap between the current usage and link capacity. For each class c and link k that has extra bandwidth, denote the flows in class c running on link k $flow_k$. We generate a set of *related* links for link k. A link l is said to be *related* to link k if k and l both belong to some flow f, or is related to any link that is related to l by the previous way. If all the links related to k are under utilized, do the following.

On each link, we impose the inequality: $m \times x + CurrentUsage \leq Capacity$, where m is the number of flows on the link from one class. x is the additional bandwidth for each flow in the class. CurrentUsage is the sum of the flows' assigned bandwidths.
We solve these inequalities independently and then pick the smallest x. Now we need only to add x to the assigned bandwidth of each affected flow. 3) Update the links' current usage and delete link k from set L. Repeat the procedure above to the remaining links in the set until it's exhausted. After the inter-link adjustments are done, we reach the local optimal solution.

4 Simulation and Analysis

We now study the performance of our algorithm in various settings in a multiple-bottleneck-link network. The scenario is that for auctions, each client bids for base price, minimum required bandwidth and the price sensitivity coefficient, the service provider uses the algorithm to calculate the bandwidth assignment, thresholds and total revenue. We compare these results to flat rate pricing by using the same set of customers to calculate the revenue generated by flat rate pricing.

4.1 Simulation Design

As an example, assume that the Best Effort (BE) class is charged $35 per client per month. On average BE traffic needs to pay 0.00135 cents per minute. Let a mcent (or a unit) be equal to 1/1000 cents, so the charge for BE as 1.35 mcents. Based on this, we define EF traffic's price twice as much as BE's, which is 2.7 mcents per minute. Meanwhile, AF's price is 2 mcents per minute. These values are the service provider's price thresholds for each class. Any customer who bids lower than the threshold price will be rejected. The customers' valuations for EF and AF are assumed to be normally distributed.

In the flat rate scenario, a customer is admitted if and only if his valuation (which is same as the bid in the auction context) is greater or equal to the fixed price set by the service provider. The revenue is the number of customers multiplied by the price.

In the bidding system, the valuations of the customers are used as their bids. The service provider would set the threshold values using the algorithm presented. When we have the thresholds, we also calculate the total revenue generated by the auction.

The comparison between revenue generated from the fixed price and the auction are done in various ways by varying the set of customers (and also their valuations), the network topology, the network load and the assumed fixed prices.

4.2 Simulation Results

Graphs are commonly used to model the topological structure of internetworks for studying problems ranging from routing to resource allocation. We use the GT Internetwork Topology Models to generate graphs that model the topological structure of the Internetwork. The details of this tool may be found in [8].

Group 1 (Figure 1, 2 and 3): In this simulation group, we vary the fixed rate cost from 0.2 to 2.2 times the mean valuation (MV) of what the customers are willing to pay. Customers' mean valuations are 4 mcents for EF, 3 mcents for AF in this case. There are 20 nodes and 200-400 customers in the network. The network load is 70% and the customers' valuations are normally distributed with mean of 3.5 mcents and standard deviation 2 for EF, and a mean of 2.5 mcents and a standard deviation 2 mcents for AF. Figure 1 shows a comparison of the bidding system and fixed pricing. Figures 2(b)(c) show how the network behaves when the network load is 100% and 140%.

We run the same simulations on different network topologies, with 200 nodes and 400-1000 customers (Figure 2 with 70%, 100% and 140% network load) and 1000 nodes and 3500-5000 customers (Figure 3 with 70%, 100% and 140% network load).

Group 2 (Figure 4): This group of experiments examines what happens to the revenue as the network load increases. We compare three cases: 1) fixed pricing, 2) auction where customers always bid the same way, and 3) auction where customers may change their bids at each auction. The customers' valuations are normally distributed with mean of 4.5 mcents and a standard deviation of 2 mcents for EF, and mean of 3.5 and a standard deviation of 2 for AF. The fixed price rate is 4.5 mcents for EF and 3.5 for AF and is constant for this group of experiments. We observe the revenue as the network load increases, shown in Figure 4. The X-axis in the figure represents the network load.

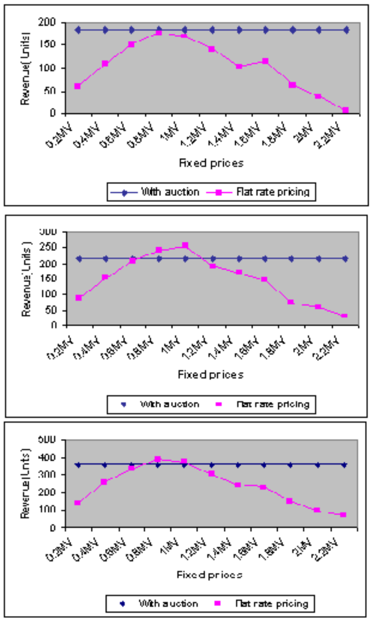

Figure 1: Revenue Comparison between Flat Rate Pricing and Auctions (with fixed price as a variable and 20 nodes)

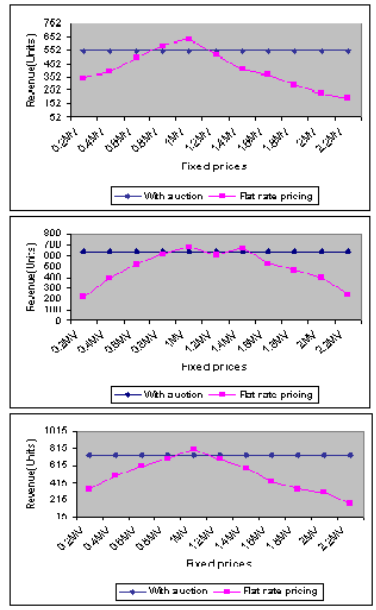

Figure 2: Revenue Comparison between Flat Rate Pricing and Auctions (with fixed price as a variable and 200 nodes)

4.3 Analysis of Results

The first group of simulations shows that in three different load conditions, auction beats fixed pricing on average. As shown in Figure 1(b), as the fixed price rate is increasing, the revenue increases and then drops again. At the peak of this curve, flat rate can beat auction in revenue generated. Therefore, if the service provider knows the exact network load and customers' valuations, there exists an optimal price that can generate the most revenue. However, in reality because those values cannot be obtained beforehand and time to generate a solution is limited, it is very hard to find the optimal solution in real time. Therefore, auctions are more adaptive to network conditions and generate more revenue on average. Figure 1(a), 2(b) and 3(b) show that as the flat price increases, the revenue generated by fixed price fluctuates. This is because when the fixed price increases, the number of admitted customers decreases. This is the reason that the revenue function is not linear.

The most important characteristic of the auction is that it is dynamic. So it is important to examine how a fixed price approach behaves compared to an auction when the network condition is changing dynamically. Such is the purpose of the group 2 experiments. In these simulations, the network topology and the flat rate prices for EF and AF are fixed, and we vary the network load. It is evident that when the network is not congested, flat rate pricing and auction behave similarly. When the network is overloaded, auction performs better. Intuitively, when customers lose an auction, they may have the intention of increasing their bids in order to win in the future. To try to model this behavior, we assume that at each auction, a certain percentage of the losers intend to increase their bids in the following auction. Compared to flat rate pricing, either that losers increase their bids or not give the service provider more room to make the selection in favor of more valuable customers while discarding the low bids. When there is congestion in the network, customers compete with each other for resources, which causes the price to go up an extra amount. This extra amount is what sets the auction apart from flat pricing.

From the results of the experiments that we have conducted, in most cases auctions outperform the flat rate pricing in either a congested or non-congested network. Moreover, the auction algorithm works adaptively to maximizing the service provider's revenue. Meanwhile, customers also benefit from this pricing model because they are given the freedom to express their valuation of service.

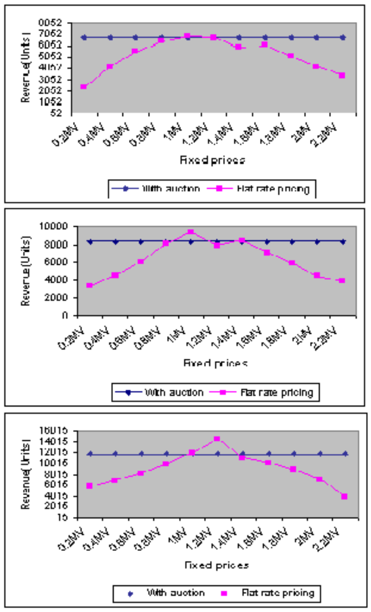

Figure 3: Revenue Comparison between Flat Rate Pricing and Auctions (with fixed price as a variable and 1000 nodes)

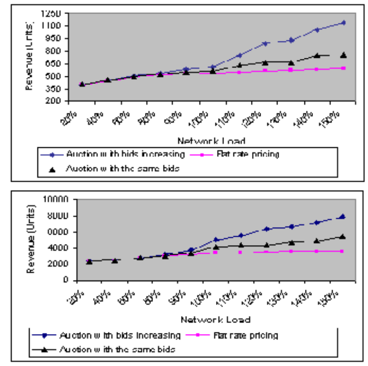

Figure 4: Revenue Comparison between Flat Rate Pricing and Auction as a Function of Offered Load

5 Conclusion

This paper was motivated by the problem of providing differentiated QoS to clients to maximize a service provider's profit through pricing. In our model, clients are allowed to bid on the price that they are willing to pay and the service that they require. The service provider decides the resources to allocate to each individual as well as thresholds for each DiffServ class based on the clients' bids, in a way that aims to maximize his revenue. Most of the work that has been done in the past considers one single bottleneck link. We have considered the problem in a larger network scope, which can contain hundreds to thousands of nodes and many bottleneck links. We presented our heuristic algorithm and the results of various simulations. The results showed that fixed pricing beats auction in revenue if the optimum price, representing the exact supply and demand balance can be selected. However, since the network changes rapidly, the best fixed price at one time could be the highly sub-optimal for the next moment in time. Auctions adjust the pricing adaptively as the network changes, which makes them a better strategy in the long run. The results of our experiments show that overall the bidding system brings the service provider more revenue than does fixed pricing.

References

[1] J. F. MacKie-Mason and H. Varian, Pricing the Internet. In B. Kahin and J. Keller, editors, *Public Access to the Internet*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[2] P. Reichl, G. Frankhauser, B. Stiller, Auction Models for Multi-Provider Internet Connections *Proceedings of IEEE Infocomm*, July 2002.

[3] N. Semret, R. Liao, A. T. Cambell and A. A. Lazar, "Pricing, provisioning, and peering: dynamic markets for differentiated internet services and implications for network interconnections," *IEEE Journal of Selected Areas in Communications*, vol. 18, No. 12, December 2000.

[4] J.N. Sidal, *Optimal Engineering Design: Principles and Applications*, Marcel Dekker, New York, NY 1982.

[5] X. Wang and H. Schulzrinne, Pricing Network Resources for Adaptive Applications in a Differentiated Services Network, *Proceedings of IEEE Infocom*, 2001.

[6] W. Yang, H. Owen and D. Blough, An Auction Pricing Strategy for Differentiated Service Networks, *Proceedings of IEEE Globecom*, 2003.

[7] Yuksel, M., Kalyanaraman, S. Effect of pricing intervals on the congestion sensitivity of network service prices. *Proceedings of IEEE Infocom*, 2001

[8] Ellen W. Zegura, Ken Calvert and S. Bhattacharjee. *How to Model an Internetwork* Proceedings of IEEE Infocom 1996, San Francisco, CA.