An Auction Pricing Strategy for Differentiated Service Networks

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

Abstract

We use pricing as an effective strategy to allocate network resources in an efficient way so as to maximize a service provider's revenue. Among all static and dynamic pricing strategies, an auction approach is a widely proposed decentralized mechanism. We propose a scenario where all clients can bid for their required bandwidth as well as the price they are willing to pay. The service provider will decide on the admission price and differentiated service provided for each class. These thresholds also provide a future reference for admitting new flows later on.

1 Introduction

The Service Provider (SP) controls network resource allocation to multi-users to provide a certain level of Quality of Service (QoS). We study the revenue maximization problem of a price-based resource allocation scheme for Differential Service (DiffServ) data networks. Since the network supports multiple class services, it needs a differentiated pricing strategy instead of the flat-rate pricing model used by current Internet services. How to charge customers in the most efficient way becomes a big issue.

Pricing has recently attracted significant attention to achieve economic efficiency in the Internet. A number of pricing schemes have been proposed [4] [7] [10]. Despite the various strategies presented, the basic idea is that the appropriate pricing policy will provide incentives for users to behave in ways that improve overall utilization and performance. An auction is a mechanism that consists of clients submitting bids, including the desired amount of resources and the price they are willing to pay, and the acutioneer who is responsible for allocating shares of the resources based on clients' bids. It is a decentralized mechanism for efficiently and fairly sharing resources inside a network [6].

The DiffServ domain is a class-based network. DiffServ defines router behaviors expected by packets belonging to each of these classes. Therefore, we need a uniformed pricing model for every single class. Paper [7] introduced the idea of the "smart market" as an efficient pricing mechanism. It allows users to bid on each packet. The packet gets transmitted if the corresponding bid exceeds the current marginal cost of transportation. An advantage of this scheme is that the user usually only pay the lower market-clearing price. The bidding process happens on a hop-by-hop basis. However, generally customers don't care about anything going on inside the network, but they do care about end-to-end behavior. This causes all the hops to have to be done with the bidding at once, which brings an overwhelming complexity to the bidding mechanism. Another main disadvantage is that in the real world, the computational burden of updating prices dynamically can be quite high.

In this paper, we consider a scenario that takes customers' bids and gives thresholds for both price and service offered for each class to maximize the SP's revenue. Flows coming and joining in later will be subjected to those thresholds during admission control. The whole pricing threshold update will occur within a certain time interval. From the historical data of a SP's charge to clients and corresponding service offered, customers know the price range generally. They then can propose their required minimum bandwidth as well as the price they are willing to pay. The SP calculates a minimum bandwidth they would provide to each class based on all the bids. Given our newly proposed DiffServ pricing strategy, our goal is to maximize the SP's profit. Our contributions can be summarized as follows:

 0-7803-7974-8/03/$17.00

- We model the transaction between the customer and the SP. We analyze the SP revenue optimization problem based on a bidding mechanism. The novelty is that customers have the freedom to choose the service and price. Because of that, the SP needs to find a way to maximize profit. The final solution reflects not only the SP's benefit but also the customer's real willingness to pay for the service.
- The actual charged price will be the lowest among those admitted flows. It assures customers that they will never pay higher than their bids and ususally it will be lower.
- The optimal solution also gives a guideline for future reference as to how to admit new flows.
- We transform flow-based input information into a class-based DiffServ domain. This makes it possible to actually realize the auction idea in DiffServ.

2 Related Work

Over the last several years, different methods of creating economic models for resource management have been proposed by a number of researchers. Some of them are based on dividing traffic into multiple priority classes, but using fixed prices for each service class [4]. By adding congestion-dependent components into the price gives a different dynamic pricing strategy. It takes network activities into account, improves network efficiency, and offers a more competitive price. Congestion-dependent pricing charges are determined on a per-call basis made at the time the call is admitted. Wang [10] proposes a strategy where the price depends on the service class' average demand. Specifically, the price is negotiable through a negotiation protocol. However, it requires resource reservation in the network, which can raise a few key issues such as inefficient use of the network, increased network cost, and most importantly impractical use in real time.

As Mackie-Mason and Varian mentioned, the price for sending packets from a particular flow should be positive [7]. From a service provider's stand point, the price should be differentiated when different kinds of services are offered. When the SP faces a potential customer, he would be able to compare his own benefit by adding a new customer to the marginal cost he imposes on other users.

Courcoubetis *et al.* gave a pricing model for DiffServ by assigning the same amount of bandwidth to all classes [5]. Equivalent bandwidth allocation for DiffServ is not a reasonable strategy though it keeps the format simple. Pricing based on equivalent bandwidth is not fair for customers getting different treatments. The scheme proposed in [4] used price to reflect the resource demand and supply situation. Pricing is worked out under a well-defined statistical model of source traffic. However, they do not take into account of traffic dynamic changes. This limitation does not fit well with current network status.

The auction algorithm is an effective model for solving classical assignment problems [3]. Bertsekas also pointed out that an auction outperforms substantially its main competitors for important types of problems, both in theory and in practice, and is also naturally well suited for parallel computation [3].

3 Problem Formulation

The network model that we use makes the same assumption as reference [9]. This assumption is that the network can be abstracted into a single bottleneck capacity, thus the analysis is simplified to a single link. An absolute amount of bandwidth can also be used to represent the capacity [9]. User experiments reported in the literature [1] suggest that utility functions typically follow a model of diminishing returns to scale, that is, the marginal profit as a function of bandwidth diminishes with increasing bandwidth. Hence, [10] develops a general revenue function as a function of bandwidth:
$U_{kj} = U_{0j} + W_j \log \frac{X_{kj}}{L_{kj}}$
where U_{kj} stands for the revenue from client k, which belongs to class j. W_j is the sensitivity of the price[1]to bandwidth for class j. U_{0j} is the base price for class j.

The base price for each class has been fixed by an internet service provider already. Customers bid with a sensitivity and a minimum bandwidth. The objective will be to maximize the SP's revenue, subject to the system's available resources. The mathematical formulation is as follows.
Decision variables:

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

X_{ij} : bandwidth obtained by client i for class j;
L_{mj} : minimum bandwidth for class j;
W_j : price sensitivity for class j;

[1]The sensitivity of the price is the amount that customers are willing to pay if allocated bandwidth is more than the minimum they require.

X_j : bandwidth assigned to each individual client in class j;
Objective function:

$$max \sum_{j=1}^{3} \sum_{i} (U_{0j} + W_j \log \frac{X_{ij}}{L_{mj}}) * Z_{ij} \qquad (1)$$

Subject to:

$$\begin{cases} \sum_{j=1}^{3} \sum_i X_{ij} <= Q \\ X_{ij} \geq L_{mj} - (1 - Z_{ij}) * M \\ W_j \leq W_{ij} + (1 - Z_{ij}) * M \\ X_{ij} \geq V_i - (1 - Z_{ij}) * M \\ X_{ij} \geq X_j - (1 - Z_{ij}) * M \\ X_{ij} \leq 0 + Z_{ij} * M \\ X_{ij} \geq 0; L_{mj} \geq 0; W_j \geq 0 \\ X_{ij} \leq X_j \end{cases}$$

Parameters:
U_{0j} : base price for class j
Q : total bandwidth
V_i : minimum bandwidth required by client i
M : a very large positive number

The scenario is that all the customers propose their values of W_{ij} and L_{ij}. We have to decide which flows should be admitted for each class with the objective of maximizing the SP's revenue. For the flows admitted to class j, we adopt the minimum W_{ij} as our W_j and the maximum L_{ij} as our L_{mj}.

4 Optimal Solution

First, we introduce an optimal solution within one class, given a certain amount of bandwidth. Then, we will expand that into multiple classes.

4.1 Solution in One Class

Within one class, *i.e.*j, we omit j index and simplify the formulation. The objective function becomes max $\sum_i (U_0 + W \log \frac{X_i}{L_i})$. Suppose N is a set of all customers who bid, M is the accepted customer set and Q_j is the assigned amount of bandwidth to this class j. The bid values from customer i are L_i and W_i. According to our policy of choosing L and W values, L should be the maximum value from the set M, while W the minimum value. This rule also ensures that {L∈L_x,x∈N} and {W∈W_x, x∈N}. Therefore, $L =$ max {L_i, i∈M}, and W =min{W_i, i∈M}. We construct another set S with all the combinations of L_x, W_x values. S={(L_x,W_x), x∈N}. For each combination (L_x,W_x), we find all the flows whose L value is less than or equal to L_x and whose W value is greater than or equal to W_x. Record the number of flows as k. k is the number of flows that can be possibly admitted if L= L_x and W=W_x. As defined earlier, each flow is sharing Q/k amount of bandwidth. If Q/k < L, that means there are too many flows and there is not enough bandwidth to support all of them. So we have to reduce the number k until the inequality Q/k >L is valid. When we have number k for each combination, the corresponding profit $U_x = \sum_x (U_0 + W \log \frac{X_x}{L_x})$ can also be generated. So each combination has 2 values associated: k_x and U_x. Now, we need to filter out some unqualified combinations by using the value m_j, the number of flows that can be accepted in class j. Beginning with m=1, find all the combinations whose $k_x \geq m$ and choose the one with the highest U_x, recorded as U_{x_1}. Increment m until m is equal to the total number of flows n, and calculate a U_x for each m. From the set { U_{x_1}, U_{x_2},..., U_{x_n}}, the U_{x_i} with the highest profit is selected. We then can get the corresponding m and (L,W). Therefore, we have obtained the best values of L_i, W_i, and m_i for the class. The corresponding value U is the SP's optimal profit for class j. L and W are used as the bid thresholds for the flows in class j.

To elaborate on this procedure, we give a simple example as follows. Suppose in this single class, we have the following bids:
C1:(L_1=2M,W_1=10); C2:(L_2=3M,W_2=11); C3:(L_3=2.5M,W_3=9); C4:(L_4=10M,W_4=12); C5:(L_5=8M,W_5=6).
The available bandwidth is given at 12M and the base price is given at 20. From these bids, we can build a matrix, including all the combinations of L_x and W_x.

$$\begin{bmatrix} (L_1, W_1) & (L_1, W_2) & (L_1, W_3) & (L_1, W_4) & (L_1, W_5) \\ (L_2, W_1) & (L_2, W_2) & & & \\ \ldots & & & & \\ (L_5, W_1) & (L_5, W_2) & .. & .. & .. \end{bmatrix}$$

Starting from (L_1, W_1), we look for clients whose $L_i \leq L_1$, $W_i \geq W_1$. Only client 1 itself satisfies the criteria. So, k=1. Test by dividing Q_j/k = 12M and it's greater than L_1=2M; therefore, it's a valid k value for $(L_1 = 2, W_1 = 10)$ combination. Provided L=2 and W=10, the revenue is $U = 20 + 10log12/2$=27.78. Proceed in this way to get all the ks and Us. Then, the next loop involves with the number of flows getting admitted. When m=1, all combinations with k≥ 1 will be considered and the highest one with U value is U_1. Next, m increases to 2, and a corresponding U_2 is chosen. Then, we have {U_1, U_2, U_3, U_4, U_5}. Among them, the highest U is our final revenue. The m value associated with that U is the number of clients SP should admit. The combination (L, W) that generated the U value is SP's threshold.

Until now, we have generated the L and W values to maximize SP's revenue, based on current clients' bids. Next, we introduce some properties to show how the SP should admit new flows to maintain its maximum profit.

Property 1: If W_j and L_j are kept the same, as long as the inequality $Q_j/m_j > L_j$ is valid, it's always true that the more flows added in, the higher value U_j is.

Proof:

The revenue function is:

$U_j = m_j * U_{0j} + m_j * W_j \log \frac{Q_j}{m_j L_j}$

Its derivative is:

$$\frac{\partial U_j}{\partial m_j} = U_{0j} + W_j \log \frac{Q_j}{m_j L_j} - W_j$$

Since $Q_j/m_j > L_j$ is valid, as long as U_{0j} is greater than W_j, $\frac{\partial U_j}{\partial m_j}$ is always greater than 0. That guarantees that U_j is a strictly increasing function.

Using property 1, the SP can increase the revenue by admitting more flows with fixed W,L values. So, after the bidding thresholds have been decided, property 1 tells the SP how to admit new flows to maximize the profit.

4.2 Solution in Multi-Classes

When we put the scenario in three classes, the objective function and corresponding constraints formulated as a Lagrangian are

$$\begin{cases} max[\ m_1 * (U_{01} + W_1 \log \frac{Q_1}{m_1 L_1}) + \\ m_2 * (U_{02} + W_2 \log \frac{Q_2}{m_2 L_2}) + \\ m_3 * (U_{03} + W_3 \log \frac{Q_3}{m_3 L_3})]; \\ Q_1 + Q_2 + Q_3 = Q; \end{cases} \quad (2)$$

Therefore, we have the following solution

$$\Rightarrow \begin{cases} Q_1 = (m_1 W_1)/(m_1 W_1 + m_2 W_2 + m_3 W_3) * Q \\ Q_2 = (m_2 W_2)/(m_1 W_1 + m_2 W_2 + m_3 W_3) * Q \\ Q_3 = (m_3 W_3)/(m_1 W_1 + m_2 W_2 + m_3 W_3) * Q \end{cases}$$

We notice that the W value does not affect others like Q, m, and L. While Q, m and L are closely related. When value m and L are fixed, it's always the best to choose the highest W to get a high U. In the last section, we introduced how to work with all (L_x, W_x) combinations in one class to get the value k. Now we use that information and work in another way. First, start from m=1 and L_i (i=1) and check all the combinations (W_x) with L_1. From these, the effective ones are those with k≥m; then, choose the one with the highest W. Now, we have m, L, and W for class j. The same procedure applies to other classes and we have m_1, L_1, W_1, m_2, L_2, W_2, m_3, L_3, W_3. Now Q_1, Q_2, Q_3 can be solved according to equation 2. Lastly, we check the feasibility of each solution by calculating Q_1/m_1,Q_2/m_2, Q_3/m_3. Only if all of them are greater than L_1,L_2,L_3 respectively, then we consider this a feasible solution and record the corresponding U value. Otherwise, it is abandoned. Following the same steps by changing the value of m and L_i, we can get all the possible feasible solutions. Finally, among all the feasible solutions, the highest U is the optimal solution.

5 Simulation and Analysis

We randomly generate a set of clients who participate in the auction. The bids include the required service class, desired minimum bandwidth, and the price they are willing to pay. We use their bids and the network's capacity as inputs into our multiclass pricing model. The SP's revenue is produced as outputs. Meanwhile, we manually assign a fixed network capacity to each class. Then, our single-class solution is used to solve the case individually within a class. The final revenue will be the sum of three individual ones. The motivation of doing so is to show that the multiclass solution takes into account not only the single class, but also the competition among those classes. The process is actually a two-step auction. In the first step, clients inside each class compete with each other to get the price as their offered price. In the second, each class tries to get as much bandwidth as it can. Finally, it comes to a converged balance point where it reaches the highest point of the SP's final revenue.

We use three simple techniques get the bandwdith allocation for each class, run the single class optimization algorithm inside the class and get the final total revenue by adding them together. We compare those results with the ones generated by our multi-class optimization algorithm to show that the multi-class optimization outperforms the three simple ways.

The ways to decide on each class' bandwidth allocation are: 1) Get the ratio of each class bandwidth assignment by selecting the highest bid from the desired minimum bandwidths for that class. For instance, the highest bid from class 1 is 20M, 12M from class 2, and 0.2M from class 3. The ratio is then, 20:12:0.2. So, class 1 gets 20/(20+12+0.2) of total capacity. The same formula for class 2 and 3. 2) Use the number of customers who bid as the reference for calculating the ratio. For example, 2 customers bid for class 1, 4 for class 2, and 10 for class 3. The ratio is 2:4:10. The amount of bandwidth is divided proportionally. 3) Calculate the total bandwidth bidded for each class if all the customers get admitted. The bandwidth allocation for classes would be proportional to the total bandwidth. For example, 2 customers in class 1 bidding for 2M and 3M; 3 customers in class 2 bidding for 1M and 1.5M; 6 customers in class 3 bidding for .09M, .1M, .15M, .2M, .21M, .08M. The ratio comes at: (2+3):(1+1.5):(.09+.1+.15+.2+.21+.08).

We set the revenue results from multi-class optimization algorithm to a value of one on the y-axis. The cases shown in Figure 1 are for six different sets of customer demands and bids. These customer demands and bids are varied to allow our algorithm to be compared to the other three simple algorithms over a variety of traffic inputs and bids. The revenue generated by our multi-class algorithm (set to units) exceeds the revenue generated by the simplier algorithms.

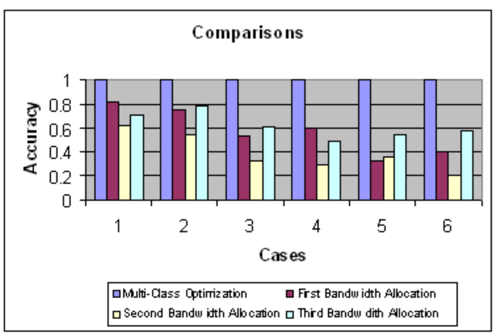

Figure 1: Comparison

6 Conclusion

This paper was motivated by the problem of providing differentiated QoS to clients to maximize service provider's profit. We presented a novel pricing strategy of maximizing service provider's revenue based on clients' bids of price as well as desired service. This scheme allows customers to express their willingness to pay along with their required service. The service provider calculates the thresholds for each service class according to network resource availability. The thresholds can also be used as a future reference for admitting new clients. Our simulation results show that the algorithm works well in assigning proper bandwidth to each class in terms of maximizing the service provider's profit.

References

[1] W.A. and M. A. Sasse, Evaluating Audio and Video Quality in Low-Cost Multimedia Conferencing Systems *Interacting with Computers*, 8:255-275, 1996.

[2] T. Basar and R. Srikant, Revenue-Maximizing Pricing and Capacity Expansion in a Many-Users Regime, *IEEE Infocom*, 2002.

[3] D. P. Bertsekas, Auction Algorithms for Network Flow Problems: A Tutorial Introduction, *Computational Optimization and Applications,* Vol. 1, pp. 7-66, 1992.

[4] R. Cocci, S. Shenker, D. Estrin, and L. Zhang, Pricing in Computer Networks: Motivation, Formulation, and Example. *IEEE/ACM Transactions on Networking*, 1(6):614-627, December 1993.

[5] C. Courcoubetis and V. Siris, Managing and Pricing Service Level Agreements for Differentiated Services, *Proc. of 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99)*. Jun 1999

[6] A. A. Lazar and N. Semret, Auctions for Network Resource Sharing CTR Technical Report CU/CTR/TR 468-97-02, Columbia University February 11, 1997.

[7] 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.

[8] J. Mackie-Mason and H. Varian, Pricing Congestible Network Resources, *IEEE JSAC*, vol. 13, no. 7,pp 1141-48, Sept 1995.

[9] N.Semret *et al.* Pricing,Provisioning and Peering:Dynamic markets for Differentiated Internet Services and Implications for Network Interconnections. *IEEE Journal on Selected Areas in Communications*,Vol.18, no. 12, Dec 2000.

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