A Comparative Evaluation of Internet Pricing Models: Smart Markets and Dynamic Capacity Contracting

Ranjeeta Singh, Murat Yuksel, Shivkumar Kalyanaraman, T. Ravichandran[1]
Rensselaer Polytechnic Institute, Troy, NY 12180
Email: ranjeeta@intel.com, yuksem@cs.rpi.edu, shivkuma@ecse.rpi.edu , ravit@rpi.edu

Abstract

Internet pricing is receiving increased attention in industry and academia. In this paper, we report the results of comparing two Internet pricing models. Using simulation techniques, we evaluate the technical and economic efficiencies of the Smart Market model proposed by MacKie-Mason & Varian (1993) and compare it with Dynamic Capacity Contracting, a pricing scheme that we have developed. Dynamic Capacity Contracting is a congestion-sensitive pricing model implementable in the differentiated services architecture of the Internet. The central idea of congestion-sensitive pricing is that, based on congestion monitoring mechanisms, a network could raise prices and vary contract terms dynamically. Our results indicate that while the smart market model achieves a higher economic efficiency, it results in poor technical performance of the network. On the other hand, the dynamic contracting model achieves a better balance of economic and technical efficiencies. We discuss the implications of our work and identify future research directions.

Key Words: Internet Pricing, Internet Economics, Differentiated Services, Congestion Control.

Introduction

The Internet traffic volume has been increasing at an exponential rate over the last few years. So far, capacity provisioning and developments in traffic management have sustained this growth in network traffic. However, network congestion has become more common resulting in a general deterioration of service levels experienced by users. Many scholars believe that it is necessary to share bandwidth in a more controlled manner taking into account factors such as application requirements, network efficiencies and economic efficiencies. Many have suggested that responsive pricing schemes can help achieve both network and economic efficiencies (MacKie-Mason, Murphy & Murphy, 1997 [11]; MacKie-Mason & Varian, 1993 [9]; Gupta, Stahl and Whinston, 1997 [6]).

Many schemes have been proposed to price the Internet and its various domains (MacKie-Mason & Varian, 1995 [10]; Clark, 1997 [4]; Gupta, Stahl & Whinston, 1997 [6]). However, there has been little experience in implementing and studying these schemes in the production Internet. We have developed a pricing model, Dynamic Capacity Contracting that is implementable in the diff-serv architecture. The objective of this paper is twofold. First, to develop mechanisms to implement both the Dynamic Capacity Contracting and the Smart Market model in the differentiated services architecture. Second, to perform a comparative evaluation of the two models on dimensions such as economic efficiency and technical efficiency.

A Review of Relevant Internet Pricing Models

MacKie-Mason and Varian (1993) [9] introduced the concept of congestion-sensitive pricing in their scheme called smart market. Under this model, the actual price for each packet is determined based on the current state of network congestion. Users are expected to bid a price for their packets and packets whose bids exceeded some cutoff amount will be admitted and the rest are dropped or buffered. The cutoff amount is determined by the condition that the marginal willingness-to-pay for an additional packet is equal to the marginal congestion costs imposed by that packet. In order to make the scheme incentive compatible users are not charged the prices they bid, but rather are charged the bid of the lowest priority packet that was admitted to the network. The goal is to find a pricing mechanism that will lead to efficient use of scarce resources. It is expected that congestion-sensitive pricing could convert delay and queuing costs into dollar costs and force the user to compare the value of her packets to the costs she is imposing on the system.

[1] All authors contributed equally to this paper and their names are listed in no particular order.

Since the granularity of price setting is at the packet level, the smart market model is expected to have a high transaction overhead. Moreover, users may not like the posteriori pricing unless it is a small part of their overall expenditure. More importantly, several technical challenges have to be ironed out to implement this scheme under the TCP/IP framework, which have not be addressed so far. Bidding higher prices does not guarantee an assured service quality; the smart market mechanism guarantees only relative priority. A packet with a high bid gains access sooner than one with a low bid, but delivery time cannot be guaranteed. Rejected packets could be bounced back to users or be routed to a slower network possibly after being stored for a period in a buffer in case the congestion falls sufficiently a short time later.

Clark [4] suggested that instead of charging for actual usage, users should be made to pay for the privilege of using the network capacity when needed. He proposed an expected capacity allocation scheme where users pay a price for a high probability of delivery for a given volume of traffic. Users specify the minimum data object size that they would like to be transferred with a high probability of delivery, together with an assumed duty cycle for these transfers. Users do not pay for the actual usage; instead if the actual usage is within their *expected capacity contract* their packets will be handled without any imposition of delay costs. If the actual usage exceeds the expected capacity during periods of congestion, users could experience a delay in the transmission. In either case, users do not pay more than their contracted capacity costs. Note that this scheme is not congestion-sensitive.

In summary, while the smart market model addresses the need for congestion-sensitive pricing, it does not have a clear deployment or service assurance model. On the other hand, the expected capacity contracting model is easily deployable in the diff-serv architecture and has very low transaction overheads. However, this scheme is an implicitly static one that does not account for network congestion. Our work is an attempt to address the deficiencies of both these models and strikes a middle ground between these two schemes in terms of granularity of price setting. In the next section, we discuss our proposed model, Dynamic Capacity Contracting.

Dynamic Capacity Contracting (DCC) Framework

DCC extends Clark’s expected capacity contracting model to incorporate short-term contracting and adds mechanisms to make it congestion-sensitive. It is similar to the smart-market scheme in the optional use of congestion-sensitive pricing and to Clark’s expected capacity scheme in the use of contracting. The key differences lie in the new mechanisms for estimating the congestion state of the network and the granularity of price setting which is per user or per session. Short-term contracts are essential to provide the degree of freedom to make DCC congestion-sensitive since long-term contracts do not give the flexibility to change the current price of a contract based upon congestion. Short-term contracts naturally expire and force re-negotiation, at which point price revisions based upon congestion measures are possible.

We can model a short-term contract (service) for a given traffic class as a function of volume (number of bytes) and the term of the contract (time units):

$$\mathit{Service(S)} = f(\mathit{Volume}\ (V)\ ,\ \mathit{Term}\ (T),\ \mathit{Price\ per\ unit\ volume}\ (P_v)) \qquad (1.1)$$

We assume that the user can send the maximum volume negotiated within the contract term at most. As in Clark's assured service model, the provider will assure that the negotiated traffic will be carried with a high expectation of delivery. In general, the user may send this traffic to any destination of its choice (i.e., a point-to-anywhere service); however for this paper, we focus on the case of point-to-point service since the measurement of congestion in the former case is non-trivial. We make one simplification to (1.1) by assuming that the term parameter (T) is fixed i.e. different users cannot choose different term values. The user now sees a simple service offering: the flexibility to contract a desired volume (V) for a fixed term (T) at a given price per unit volume (P_v), which may be congestion-sensitive (for the term).

In summary, our scheme has been designed to use pricing and dynamic capacity contracting as new dimensions in managing congestion, as well as to achieve simple economic goals. In this sense, it is well positioned as a pragmatic intermediate approach between Clark’s expected capacity model [4] and the smart-market model [9]. The key benefits of our scheme are:

- A framework for congestion-sensitive pricing (not usage-based or flat-priced)

- Does not require per-packet accounting anywhere (works at a contract term granularity)
- Provides deterministic service assurances like Clark's model [4]
- Does not require upgrades or software support anywhere in the network except at logical boundaries (or edge nodes)
- Uses price and dynamic capacity contracting as a truly new dimension in managing network congestion.

Implementation of DCC and Smart Market Models in the Diff-Serv Architecture

We implemented both the DCC and the smart market models in a simulated diff-serv architecture in order to evaluate their performance. A simple network model with a single bottleneck and edge-to-edge aggregate flows was used for the implementation.

DCC Implementation

We set up short-term contracts between a "customer" component and a "provider" component (which stands at the enterprise-ISP boundary). The customer initiates the service with a request for the table of contracts available at the provider. In response, the provider computes the entries of the table (price per unit volume, P_v), maximum available capacity (bottleneck capacity times contract term) and term of the contract and returns the table to the customer. In this initial scheme, we assume a single entry in the table, which specifies P_v for a contract term of length T. The price, P_v, is based upon the formula: $P_v = \frac{\sum B_i}{\min(\text{average rate limit, link capacity}) * Term}$, where ΣB_i is the estimated total budget of all customers for the contract term.

The *average rate limit* is calculated over the contract term and is based upon a measure of congestion in the network. This parameter, is a measure of the "congestion-sensitivity" of the pricing scheme. The contract term is sub-divided into smaller observation intervals and a decision is made whether the network is congested in each of these smaller intervals. Each observation interval when congestion is seen is called a congestion epoch. Identification of congestion epochs on an edge-to-edge basis is a non-trivial task. However, this problem has been solved by another group recently [7]. We used the tools developed in [7] to identify congestion epochs in our work.

The customer chooses a desired volume of premium data traffic to be sent in time T based upon the price per unit volume, P_v, a demand curve, and his available budget. The demand curve is assumed to be a simple hyperbolic curve between price and aggregate demand (volume). The volume contracted by a customer is calculated as (B_i/P_v) where B_i is the customer's budget. But we bound the contracted volume by a maximum volume, V_{max} (which is equal to bottleneck capacity times T) permitted by the provider to avoid access link congestion. In such cases, any unused customer budget is carried over to the next term for that customer. We also assume that the customer has equal default budget allocations per-contract term (which may differ from the allocations of other customers).

This choice of a volume is then conveyed by the customer to the provider, which sets up a leaky bucket traffic conditioner to mark up to V bytes in time T as "IN" (high priority). Observe that this contracting now defaults to the expected capacity framework proposed by Clark [4]. Specifically, this scheme provides service assurances and is *not* just a best-effort service or a service whose quality is more probabilistic and dynamic as in the smart market model [9].

What happens when a customer sends more traffic than contracted as IN-profile traffic? Clark's model [4] suggests that this traffic be marked as OUT and admitted into the network where a differential dropping scheme would drop them if necessary before IN packets. But in our recent work [7] we have seen that it is better from an end-to-end performance perspective (especially for TCP flows) if the bottlenecks are distributed to the edges of the network where they are likely to be smaller. This is because buffer management schemes do not scale well with an increase in the number of contending TCP flows at the core. Therefore, it makes sense to send only a part of the excess traffic into the network marked with OUT tokens and hold the remaining excess traffic at the edge. The average rate limit provides a basis for determining how much excess traffic should be allocated OUT tokens. Specifically, the total pool of OUT tokens is simply the difference between the *average rate limit * contract term* and the *sum of the contracted (IN) volumes* of all customers. We then split this OUT token pool equally among the contending customers' excess traffic. Future work may explore other ways of sharing this OUT token pool.

Smart Market Implementation

In the smart market model a per-packet-charge which reflect marginal congestion costs is imposed. The price-per-packet varies dynamically depending on the level of congestion in the network. Users assign a "bid" value for each packet and this packet tries to make through the network. Each packet has a probability of being dropped depending on the current threshold (cutoff) value among the routers in the network. This threshold value depends on the level of congestion at the particular router, and is adjusted by that router. Finally, users pay the highest threshold value that it passed through, also called "market-clearing price".

The smart market scheme assumes that the customers are fed back the clearing price information immediately without any delay, which is not possible to implement on a real wide-area network. So, an approximation is needed. We use deterministic time intervals at ERs and IRs set to be larger than RTT as a way to handle this feedback problem i.e., customers get feedback from the network at the end of each time interval thereby they can make adjustments to their bid values and demands. So, the length of this time interval is a comparable measure to the length of contract term in DCC.

What should a customer do when she is fed back the current status of the network? Mac-Kie-Mason & Varian, (1993) [9] suggests that each customer should maximize their utility ($u(x) - D(Y) - px$) by selecting the best *x*, where *x* is the demand (number of packets to send), *u(x)* is the utility of the customer, *p* is the current price charged for a packet in the network, *Y* is the utilization of the network, and *D(Y)* is the delay experienced by the customer. We assumed a concave indifference curves between delay and packets sent and derived the utility function and the corresponding demand function (which identifies the number of packets to send in the next time interval) for the customer. It turned out that customers should adjust their demand inversely proportional to a positive power of the clearing price. In addition, since the value for x should reflect the budget constraint of the users, they would chose: $x = \frac{Budget}{p}$. After deciding the volume, the customer bids randomly between the clearing price and the maximum bid value that she can afford.

Performance Analysis

We conducted simulation experiments to compare the performance of the two models. Our objective was to evaluate model performance in terms of both technical efficiency and economic efficiency. The performance measures we use for both schemes are utilization, queue length, relative volume allocations, throughput, goodput and packet loss.

We used a simple network configuration in our analysis. The configuration includes a single bottleneck with a rate of 1Mbps. The bottleneck can be accessed by the customers through an edge router (corresponds to the provider). The customers send constant bit rate UDP traffic with fixed packet sizes (1000 bytes). The contract term in DCC and the length of the time interval for feedback in the smart market are set to be 0.4sec. Also, the length of the observation interval in DCC is set to 80ms. For both schemes, we ran three experiments with the parameters defined in Table 1. Simulation time for the experiments is 4secs. The first two experiments have two customers with equal (60 units each per term) and unequal budgets (25 and 60 units per term) respectively. The third experiment has three customers with unequal budgets (15, 25 and 35 units per term).

Tables 2 shows the average volumes and the total volumes allocated to the customers during the experiments. It shows that total volume allocated to all customers is significantly higher in the case of DCC. This indicates that DCC better utilizes the bottleneck. Figures 1, 2 and 3 plot the normalized values of volumes allocated to the customers. They indicate that the smart market model allocates the volume to the customer almost proportionally to their budgets, whereas DCC allocation is a little less proportional to the budgets. This implies that in comparison to the smart market model, DCC has a lower economic efficiency.

Figures 4 through 9 show the bottleneck utilization under the DCC and the smart market models in the three experiments. For the smart market model, we observe that there is a large transient period before steady state is reached. Figures 10 to 15 show the queue length at the bottleneck for the two schemes. For both schemes we observe a stabilized queue suggesting that both schemes are able to control congestion.

In summary, the experiments suggest that the DCC is better from a congestion management perspective because it achieves a higher utilization and a quicker convergence to steady state. Interestingly, this is achieved without seriously distorting volume allocations, which are in fact, close to those attained by the smart market scheme. Nevertheless, from a pure economic efficiency perspective, the smart market scheme appears to fare better. We have drawn these conclusions by looking at the queue lengths, utilization and mean volume allocations. The other metrics (throughput, goodput and packet loss) presented in Table 4, reiterates the congestion management benefits of DCC. In particular, though the number of packets dropped is similar, the aggregate throughput and goodput is markedly better in the case of DCC.

Summary

We have proposed a dynamic capacity contracting (DCC) framework primarily inspired by the work of Clark [4] and Mac-Kie Mason & Varian [9], and the diff-serv architecture[1] which provides a platform for implementation. The distinguishing features of our work include the idea of short-term contracts, mechanisms to support congestion-sensitive pricing of such contracts, use of pricing as a tool for congestion management, and a pragmatic focus on deployment issues. We have also proposed a sample scheme in this framework to illustrate the potential of the framework and illustrate its comparative performance tradeoffs vis-à-vis the smart market scheme. We believe that the DCC framework could play a role in transition from today's completely flat-priced system towards a system that includes congestion-sensitive pricing for certain classes of service. We have also proposed a sample implementation model for the smart market scheme and explored the difficulties and issues in its implementation. Our on-going research in this area include the following:

- Expansion of the concept of contracting to point-to-anywhere contracts
- Exploring the concept of “bandwidth intermediary” to facilitate the mediation between customers and multiple providers which employ the DCC framework
- Improving the basic DCC scheme itself in several dimensions, notably in its relative volume allocations, dynamic estimation of budgets and demands and to support a rich variety of contracts
- Investigating the issue of how long “short-term” contracts can be while maintaining the congestion-sensitivity of the scheme.

References available upon request from the authors. Please send your requests to ravit@rpi.edu.

Experiment Number	Budgets of Customers		
	#1	#2	#3
1	60	60	-
2	25	60	-
3	15	25	35

Table 1: Parameters of the experiments.

	DCC				Smart Market			
Experiment Number	Customer			Total Volume	Customer			Total Volume
	#1	#2	#3		#1	#2	#3	
1	0.18	0.19	-	0.36	0.14	0.14	-	0.28
2	0.14	0.25	-	0.39	0.07	0.18	-	0.25
3	0.09	0.12	0.15	0.36	0.04	0.07	0.11	0.22

Table 2: Mean and total volumes (Mbps) allocated to customers in DCC and the Smart Market.

	DCC			Smart Market		
Experiment Number	Goodput (Mbps)	Throughput (Mbps)	Packets Dropped	Goodput (Mbps)	Throughput (Mbps)	Packets Dropped
1	0.964	0.966	33	0.700	0.748	42
2	0.958	0.958	40	0.615	0.663	43
3	0.944	0.946	35	0.537	0.583	40

Table 3: Performance metrics of the experiments for DCC and the Smart Market.

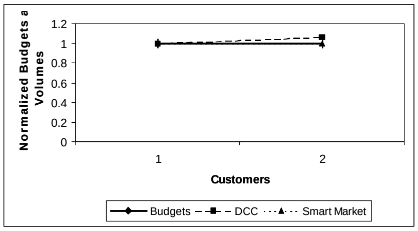

Figure 1: Normalized budgets and volumes of customers in Experiment 1.

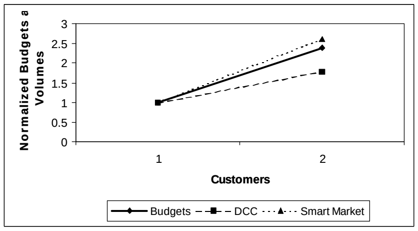

Figure 2: Normalized budgets and volumes of customers in Experiment 2.

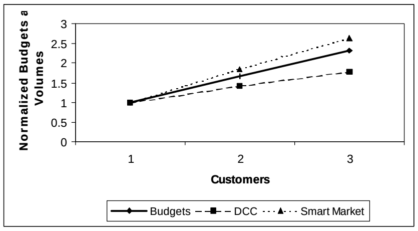

Figure 3: Normalized budgets and volumes of customers in Experiment 3.

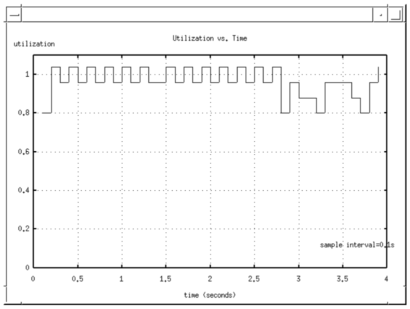

Figure 4: DCC Bottleneck Utilization in Experiment 1.

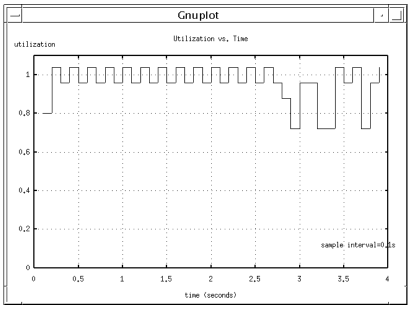

Figure 5: DCC Bottleneck Utilization in Experiment 2.

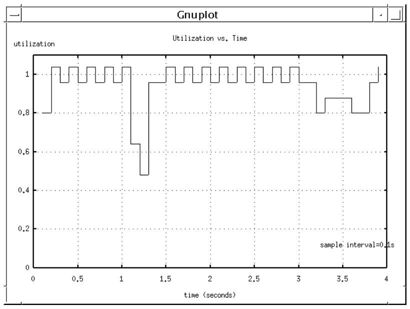

Figure 6: DCC Bottleneck Utilization in Experiment 3.

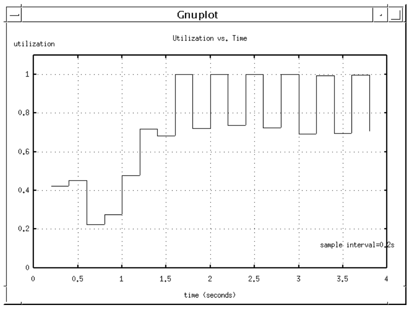

Figure 7: Smart Market Bottleneck Utilization in Experiment 1.

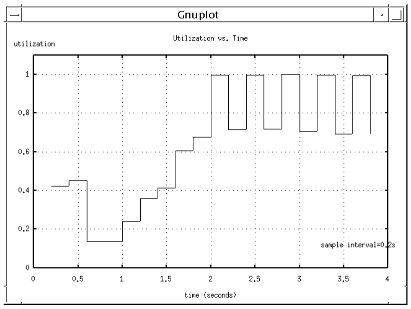

Figure 8: Smart Market Bottleneck Utilization in Experiment 2.

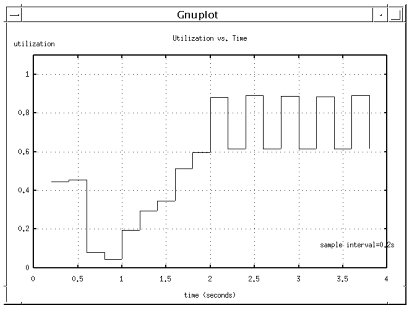

Figure 9: Smart Market Bottleneck Utilization in Experiment 3.

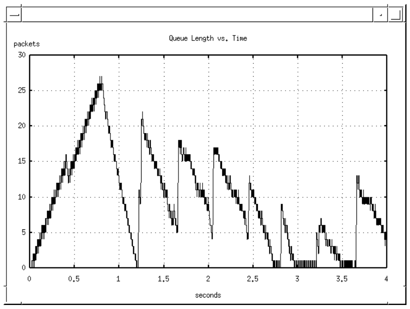

Figure 10: DCC Queue Length in Experiment 1.

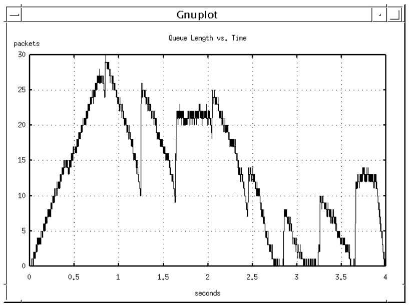

Figure 11: DCC Queue Length in Experiment 2.

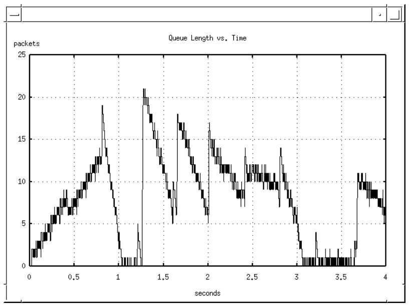

Figure 12: DCC Queue Length in Experiment 3.

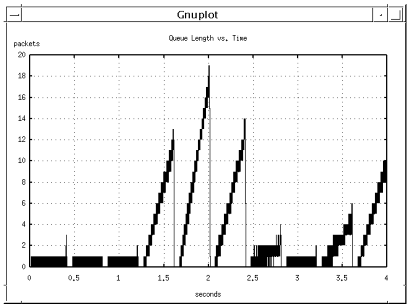

Figure 13: Smart Market Queue Length in Experiment 1.

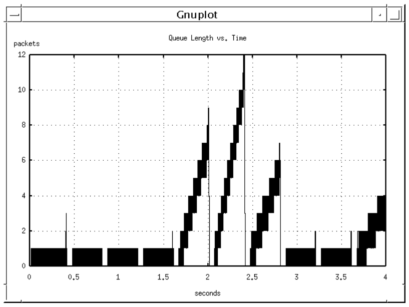

Figure 14: Smart Market Queue Length in Experiment 2.

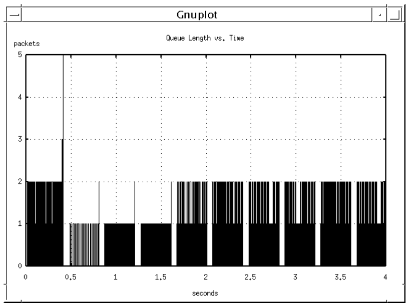

Figure 15: Smart Market Queue Length in Experiment 3.