Dynamic Capacity Contracting: A Framework for Congestion-Sensitive Pricing[1]

Ranjeeta Singh, Murat Yuksel, Shivkumar Kalyanaraman, Thiagarajan Ravichandran
Rensselaer Polytechnic Institute, Troy, NY 12180
E-mail: ranje@cs.rpi.edu, yuksem@cs.rpi.edu, shivkuma@ecse.rpi.edu, ravit@rpi.edu

Abstract

In this paper, we propose a framework called **"Dynamic capacity contracting"** which may be implemented using the differentiated services architecture in the Internet. A particularly interesting aspect of this framework is its use in implementing congestion-sensitive pricing. The central idea of congestion-sensitive pricing is such that, based on congestion monitoring mechanisms, a network could raise prices and vary contract terms dynamically. We have developed a congestion-sensitive pricing scheme in the proposed framework. We also present a preliminary performance evaluation of technical and economic efficiency aspects of this scheme and position it relative to prior work by Clark and Mac-Kie Mason/Varian. Another contribution of this paper is to model the smart market scheme in a packet-based simulation for the purpose of understanding architectural, implementation constraints and performance.

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

1. Introduction

As the Internet grows in size and heterogeneity, it is useful to consider new tools and techniques for managing congestion. *Technical* tools such as differentiated services [1], TCP/IP improvements like SACK [11], over-provisioned ISP cores have been developed and deployed in the Internet. However, *economic* measures like responsive pricing have been proposed [4], [6], [8], [9] but not deployed, though recent work by Edell and Variaya [14] is an important step in this direction. A major impediment in the deployment process has been the minimal "best effort" service model of the IP, which does not provide a standard mechanism to specify packet-forwarding behavior other than the "best-effort" service. In other words, several of the proposed schemes have remained in the theoretical domain due to lack of models for implementing them using IP.

[1] This work was funded by NSF Contract number ANI9819112 and an Rensselaer Outstanding Young Faculty Award

However, this scenario has recently changed as the Internet Engineering Task Force (IETF) has standardized two approaches to support service differentiation. The first approach is called Integrated Services ("int-serv") [3]. The second approach called Differentiated Services ("diff-serv") [1] is expected to provide scalable service discrimination without the need for per-flow state signaling at every hop. Though the two approaches can coexist and interoperate, it is expected that the latter (diff-serv) will be the choice of ISPs and backbone inter-network providers. In this paper, we focus on developing a pricing framework that utilizes the advanced traffic management features offered by the differentiated-services architecture.

The rest of the paper is organized as follows. First, we summarize the control mechanisms and traffic management features of the diff-serv architecture that are relevant to Internet pricing. Next, we describe Dynamic Capacity Contracting: a ***flexible and dynamic framework for implementing a range of pricing schemes within the differentiated services architecture.*** We then describe a congestion based pricing scheme developed using this framework and position it relative to other pricing proposals in literature. Further, we present some performance results to demonstrate the potential of this framework. We have developed an implementation model of Mac-Kie Mason/Varian's (MMV) smart-market scheme and compare the performance of our scheme with this scheme. We would like to re-emphasize that main focus of this paper is to address pragmatic new ideas in Internet pricing and related technical and deployment issues. However, we do not focus on complete refinement of the schemes or thorough performance analysis. This is a topic for future work.

2. Control Mechanisms in Differentiated Services Architecture

There are two types of routers in differentiated services (diff-serv) model: interior routers and edge routers. The interior routers perform only a small set of highly optimized packet

forwarding functions and classify packets based on the IP header field (known as “per hop behavior” (PHB)) whereas, the edge routers and nodes perform complex control and stateful data plane functions. More details on the diff-serv architecture can be found in [1].

Nichols, Jacobson and Zhang [12] have proposed a control architecture that includes a bandwidth broker for handling signaling and Service Level Agreement (SLA) negotiations between customers and providers. The bandwidth broker is a software intermediary that is aware of the policies concerning the user and liaisons between the user and possibly multiple providers. Common Open Policy Service (COPS), a policy-based architecture, has also been proposed for the control and provisioning of differentiated services networks [2]. The differentiated services model combined with the proposed signaling/control schemes allows economic concepts such as "*price-based service differentiation*" and "*contracting*" to be incorporated within IPv4 albeit with the following constraints.

First, diff-serv is built around the concept of *aggregation*, and hence it is important to formulate service properties, which are not destroyed due to the operations of aggregation and de-aggregation.

Second, the asymmetry between edge routers and interior routers dictates *that stateful mechanisms* like accounting and pricing must be primarily *concentrated at the edge* with minimal (preferably stateless) participation of interior routers.

Our work attempts to leverage this framework and simplify the pricing schemes to be implementable under these constraints.

3. Related pricing proposals

Among the proposed pricing proposals***, flat-rate pricing*** [2], is the most common mode of payment today for bandwidth services, and is popular for several reasons. It has minimal

accounting overhead, and encourages usage. However, flat rate pricing has problems. During congestion, the marginal cost of forwarding a packet is not zero, and flat pricing does not offer any (dis)-incentive for users to adjust their demand, leading to potential ``tragedy of commons" [6]. But there is an interesting debate by Odlyzko [13] who suggests that flat pricing and over-provisioning is a sustainable strategy given the falling costs of bandwidth and available lead-time for network provisioning. Our view is that there is no congestion issue if capacity and provisioning speed exceeds demand. But we believe that there exist several niches (e.g. international links, tail circuits to remote markets, peering points or complex meshed networks) where bandwidth, even though technically available *cannot be added fast enough.* This is because the company probably does not own the links (or jointly-owns it with a partner), or the part of the network is so large that carrier-class upgrades take time (upgrade-cycles of one year are quite common).

Two prominent pricing proposals are: 1) to regulate usage by imposing a fee based upon the amount of data actually sent (called ***usage-based pricing***) and 2) use a fee based upon the current state of congestion in the network (called ***congestion-sensitive pricing***). On the commercial side, ISPs are starting to sell OC-3 (155 Mbps) access rapidly, but with usage-based pricing on port-usage (not on an end-to-end basis). Usage is typically measured over 5 min intervals and a monthly charge is assessed based upon the average of these measurements after removing the outliers. The rates vary around $600-800/Mbyte/month. *The problem with usage-based pricing is that usage costs are imposed regardless of whether the network is congested or not.* Further, it does not address the congestion problem directly, though it does indirectly make users more responsible for their demands. Also, some users may not like ***a posteriori*** pricing unless it is a very small part of their overall expenditures. Odlyzko's arguments [13] assert that today end-users indeed see

bandwidth as a small part of their IT operational expenditures, though we believe the same is not the case for small and regional ISPs.

Other schemes that have been proposed are: MMV's ***smart-market scheme*** [9], Gupta's ***priority pricing*** scheme [6] and Kelly's ***proportional fair pricing/rate-allocation scheme*** using the concept of shadow prices [8]**.** *Shadow price* is a measure of the total marginal congestion cost that an *increase* in x_i (where x_i denotes the person i's use of network resources) imposes on the users. Economists say that the shadow price "internalizes" the externality by making the user face the costs that he imposes on other users. Though Kelly's scheme has strong fairness characteristics, it is hard to deploy on the differentiated-services model in the Internet because it requires both end-system and network support similar to the smart-market scheme. [10]

Clark [4] proposed an "***expected capacity****"* allocation scheme where users pay a price for an assurance of delivery of a given volume of traffic. Note that this scheme is not congestion-sensitive. This "contracting framework" requires simple negotiation between a customer and a provider and does not require end-system support. Though it does require some network upgrades, it was appealing to both ISPs and users, and as a result inspired the development of the differentiated-services architecture itself.

In summary, while the congestion-sensitive proposals do not have a clear deployment or service-assurance models, Clark's model is not congestion-sensitive. Our work is an effort to provide a middle ground between these approaches.

4. The Dynamic Capacity Contracting (DCC) Framework

In this section, we describe the proposed "Dynamic Capacity Contracting" (DCC) framework. This framework extends Clark's model to incorporate short-term contracting and adds mechanisms to make it congestion-sensitive. The proposed framework 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 contracting (at per user or per session levels and not at a per packet level).

The reason we need short-term contracts for congestion-sensitive pricing is because 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 we can revise the price based upon congestion measures.

In summary, 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):

$$Service(S) = f(Volume\ (V)\ ,\ Term\ (T)) \qquad (4.1)$$

We assume that the user can send up to the maximum volume negotiated within the term of the contract. 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 information in the latter case is non-trivial. We make one simplification to equation (4.1) 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).

5. A Sample Scheme in the Dynamic Capacity Contracting Framework

This section develops a simple scheme in the framework, which we also use in our performance evaluations. We want to emphasize that this is only the first and simplest of a set of schemes than can be developed in this framework. More complex schemes could be designed to

meet other complex objectives. Specifically, the scheme sets up short-term contracts between a "customer" component and a "provider" component (which sits at the enterprise-ISP boundary). The customer initiates this process with a request for the table of short-term contracts available at the provider: a "send table" request. In response, the provider computes the entries of the table (price per unit volume, P_v), maximum available capacity (bottleneck capacity * 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 $P_v = \frac{\sum B_i}{\min(average\ rate\ limit, link\ capacity) * Term}$, where ΣB_i is the estimated total budget of all customers for the contract term.

5.1 Average Rate Limit: A Congestion-Sensitive Parameter

The ***average rate limit*** is an important parameter, which is calculated over ***the contract duration (or contract term)*** and is based upon a measure of congestion in the network. This parameter, in other words, is what captures the "congestion-sensitivity" of the contracting and pricing scheme. The basic idea is that 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 period**" or a "**congestion epoch**".

Identification of "***congestion epochs***" is a non-trivial task, especially if it needs to be done on an edge-to-edge basis in a network, transparent to both the end-systems and the interior routers of the network. However, this problem has been solved by another group of our team recently [7]. The key idea in that work is to maintain state on a per-edge-to-edge aggregate basis, and identify congestion using ***observation intervals*** at the egress node which are sized to be larger than the

maximum round trip time of any edge-to-edge path. The observation intervals of the edge-to-edge congestion detection scheme are in fact the preferred subdivisions of the contract-term (see figure below).

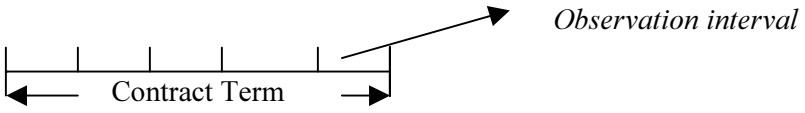

Figure 5-1: Visualization of observation intervals in contract duration.

Moreover, the egress edge upon detecting such *congestion epochs* feeds back the measured edge-to-edge aggregate output rate (μ). This fed back rate is an important factor determining the "**average rate limit**" because we intend to construct the average rate limit as the mean of rate-limits in every observation interval. Specifically, in every congested epoch, we assume that the rate limit is β*μ, where β is a fixed fraction between 0 and 1 (set to 0.9 in this paper), and μ is the observed edge-to-edge output rate fed back by the egress node. The reasoning behind this setting is explained in [7], and is specifically shown to be robustly stable from a congestion control perspective.

During observation intervals where *congestion is not seen*, we assume that the rate limit increases using an additive increase policy. Specifically, the rate limit is incremented by $\Delta = 1$ packet/RTT. The average rate limit is simply the mean of each of the rate limits in the observation intervals. The rate limit for the first observation interval in a contract term is initialized to the average rate limit in the previous contract term, and the very first rate limit is assumed to be the access link rate. Unlike traditional rate-based congestion control, we assume in this paper that the "rate limit" is not *directly* enforced, but is used *indirectly* to calculate pricing and let the user reduce its demand based upon the price.

The performance difference in the two approaches from a congestion-control perspective occurs because pricing of short-term contracts is done at a larger time-scale, which limits the

fidelity of control. Moreover, the level of control is also a function of how well the price can be set relative to the real demands and budgets of users, i.e., its effectiveness is also a function of the demand and budget *estimation* capabilities of the pricing scheme. In this paper, we assume a simple static model of the estimation of demands and budgets. Future work will explore real economic issues in this regard and issues surrounding how large the contract term can be in order to be truly "congestion-sensitive" while balancing the transaction costs of the user.

5.2 Customer Model

We assume a simple customer model for our initial proof-of-concept. 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 its available budget. The demand curve is assumed to be a simple hyperbolic curve between price and corresponding demand (volume), i.e. the volume that is contracted by a customer is calculated as (B_i/P_v). Here, B_i is the budget of the customer. But we bound the contracted volume by a maximum volume, V_{max} permitted by the provider (say to avoid access link congestion). In such a case, any leftover budget is carried over to the next term. We also assume that the customer has equal default budget allocations per-contract term (which may differ from the allocations of other customers). Our pricing model *assumes that the provider is able to estimate at least the sum of all the budget allocations of its customers per contract term*, if not every individual customer budget. Such aggregate budget estimation in the future would need to model budget changes by individual customers across contract terms, and the entry/exit of customers from the pool.

In our simple modeling we assume that V_{max} is set to the bottleneck capacity times the contract term. In other words, we assume that an estimate of (the possibly remote) bottleneck capacity is known. Again, this aspect could use better estimation in future work.

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 like the smart market model [9].

5.3 OUT-profile Token Allocation: Excess Best-effort Traffic

The next arising question is what happens if the customer sends more traffic than contracted as IN-profile traffic. Clark's model suggests that this traffic be marked as OUT (referring to "out-of-profile") 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 could explore other ways of sharing this OUT token pool. In summary,

$$\textit{Out token rate per customer} = (\mu - \textit{total_in_rate})/(\textit{number of customers})$$

The packets thus conditioned in a congestion-sensitive manner, enter the network and proceed through a series of interior routers in the differentiated services network till they reach the

egress edge router. Similar to Clark's model, we expect interior routers to provide support for service differentiation by using a ***priority drop algorithm*** proposed by Clark known as ***RIO*** (Random Early Drop, with IN/OUT marking) [4]. RIO is a simple extension of the well-known RED drop algorithm [5], with different parameter sets for IN packets and OUT packets.

In summary, our scheme has been designed to use pricing and dynamic capacity contracting as a new dimension in managing congestion, as well as to achieve simple economic goals. The key benefits of our scheme are:

a) a framework for congestion-sensitive pricing (not usage-based or flat-priced)
b) does not require per-packet accounting anywhere (works at a contract term granularity)
c) provides deterministic service assurances like Clark's model, and
d) does not require upgrades or software support anywhere in the network except at logical boundaries (or edge nodes)
e) uses price and dynamic capacity contracting as a truly new dimension in managing network congestion.

In this sense it is well positioned as a pragmatic intermediate approach between Clark's expected capacity approach [4] and MMV's smart-markets approach [9]. The next section develops a packet-based network simulation of the smart-markets scheme in order to perform a meaningful comparative analysis. We discovered that this task itself was non-trivial and warrants discussion.

6. The Smart Market Scheme Implementation

This section presents our implementation of the smart-market pricing scheme, which was proposed by MMV. The smart market imposes a per-packet-charge, which is reflective of incremental congestion costs. The price-per-packet varies dynamically depending on the level of congestion in the network. In the smart market, users try to send their packets depending on the

level of congestion in the network and their per-packet utility levels. In other words, users will value each packet depending on importance of this packet for them. Users assign a "bid" value for each packet and this packet tries to make through the network. So the smart-market does not provide guaranteed service to users. 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, user pays the highest threshold value that it passed through, also called "market-clearing price". Please refer to [9] for further details of the smart market.

The implementation challenges of the smart market scheme are three fold: a) How to communicate the parameters required by customers to make a bid? b) How to balance the implementation requirements on core and edge nodes in a diff-serv framework? c) How to deal with parameter sensitivity, non-linearity and stability issues on a packet-based simulation? Our contribution is to address these issues and to develop a comparable model of the smart market scheme.

6.1 Implementation of the Smart Market in Diff-Serv Framework

We propose the following changes to adapt the smart market scheme over a diff-serv framework. The sender sets the bid value, b, in the packet and sends it to the network. The packet passes through an ingress edge node and series of Interior Routers (IRs), each of which has a threshold value τ. IRs simply drop the packet if it does not satisfy the condition of $b \geq \tau$. If the packet satisfies the condition, it is placed into the queue sorted according to its bid value. Note that this is a priority queue, and not FIFO. Therefore, it may reorder packets, which is not suitable for TCP. We assume UDP flow in all of our simulations.

If the packet goes through the network i.e. it reaches the egress edge node, then accounting is done for this customer according to the clearing price of the packet. The clearing price is written in an additional field of the packet header. This clearing price field is updated at each IR to the maximum of the prior value of the field and the current threshold value, τ, at that particular IR bottleneck. In other words, if the value of the threshold, τ, is greater than the value of the clearing price field of the packet, the value of τ is copied into that field. Else, the field is left unchanged. When the packet reaches the egress edge router, it contains the maximum of the threshold at all the IRs it passed through. The egress node then charges that clearing price to the owner of the packet.

So far everything in the smart-market can be adapted to the diff-serv architecture, albeit with a new field in the packet header. However, the information required by the customers in order to define their utility functions and bids require new feedback mechanisms. Also, the smart market scheme assumes that the customers are fed back such information immediately without any delay, which is not possible to implement on a real wide-area network. So, an approximation is needed.

We propose the use of deterministic time intervals at ERs and IRs set to be larger than RTT as a way to handle this feedback problem. Further, the customer sends a "probe-packet" (in addition to data packets) to investigate the current status of the network at deterministic time intervals. This probe-packet goes through IRs and finds the clearing price. IRs update the probe-packet's clearing price field as if it is a data packet, but do not drop this packet. The egress edge node receives this probe-packet and sends a feedback to the customer containing current clearing price of the network. Next, the customer makes adjustments to bid values and demand (number of packets to send) depending on the feedback and her available budget.

Note that if we want the IRs to treat the probe-packets and the feedback packets just like data packets, they must have bid values as high as possible to ensure that they won't be lost and

will encounter minimum delay. That means there has to be a maximum value for the bids of the data packets, which is a deviation from the smart market because it does not impose any limiting value for the bids. Alternatively, the IRs of the network have to behave differently for these probe and feedback packets. However, this will increase the processing time of a packet at the IRs, i.e. each packet will be checked whether it is a data packet, probe or feedback packet. We choose to normalize the bid values into a range (e.g. 0 to 1) and hence define a maximum value for the bids, i.e. 1. Once normalization (mapping to [0,1]) is done, there must also be a way of reversing this mapping back. What is going to be the actual money in dollars to charge for a clearing price of, for example, 0.75? We currently leave this question, which is important for the service providers, unanswered.

6.2 Simulation of the Smart Market Scheme

We simulated the smart market in *ns* according to the ideas presented in the previous section. One big question was how to calculate the threshold value, τ, at IRs. The authors of [9] determine the congestion price of a packet as $p = \frac{n}{K} D'(Y)$, where n is the total number of customers in the network, K is the capacity of the network, Y is the utilization of the network, and $D(Y)$ is the delay experienced by a customer. Although $\frac{n}{K}$ is not constant for a real network, we assumed it to be a constant for simplicity. The IRs also maintain deterministic time intervals at the end of which they calculate the rate of change in the delay, and update the threshold value, τ, by mapping this threshold value, τ, to the interval [0,1].

Implementation of the customers was the most problematic part of this scheme, because the smart market scheme does not define a demand model. In other words, what should a customer do when she is fed back the current status of the network? We propose an approximation to MMV's

utility function. MMV proposes that each customer should maximize $u(x) - D(Y) - px$ by selecting the best *x*, where *u(x)* is the utility of the customer, and *p* is the current price charged for a packet in the network.

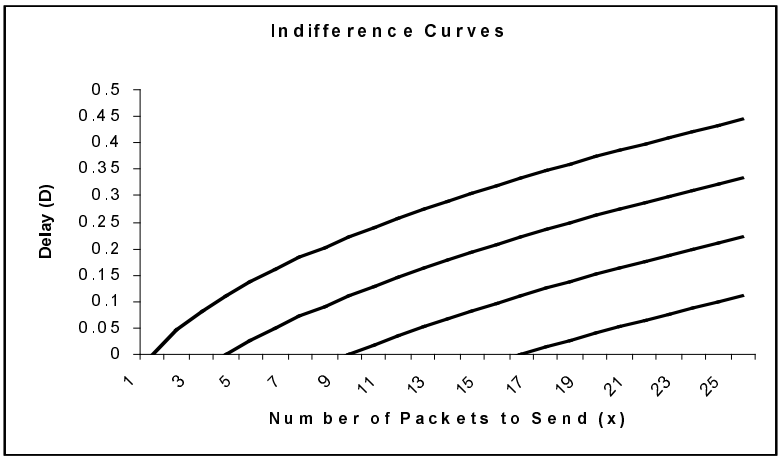

Figure 6-1: Sample Indifference Curves for delay and number of packets to send.

We first modeled indifference curves for delay, *D*, with respect to the number of packets to send, *x*. So, to be satisfied at the same level, the customer must be sending more packets having less marginal delay. This can be represented with a concave function instead of a convex function, which is the general assumption for indifference curves.

The intuitive idea is that the number of packets to send must be inversely proportional to the price-per-packet currently advertised by the network. Several indifference curves can be found to represent the case, but all of them must be concave as in Figure 5-1. In our implementation, we used the demand function:

$$x = (aD + b)^2 \qquad (5.1)$$

Utility function for these indifference curves can be defined in different ways. We selected the constant b in (5.1) as the utility function, because it perfectly orders the indifference curves from left to right, i.e. the very left indifference curve gets the lowest utility value, and the curves at

right get higher and higher utility values. By using this idea, we can get the utility function in terms of x and D as

$$u(x, D) = \sqrt{x} - aD \tag{5.2}$$

which leads to marginal utility of

$$u'(x, D) = \frac{\partial u(x, D)}{\partial x} = \frac{1}{2\sqrt{x}}. \tag{5.3}$$

Since customers have to find the best value of x to maximize $u(x, D) - px$, they must equate their demand (number of packets to send, x) to the marginal utility. This results in

$$p = u'(x, D) = \frac{1}{2\sqrt{x}}. \tag{5.4}$$

Notice that customers know the value of the current clearing price when they get the feedback from egress nodes. So, they want to find the best value of x to use in the next interval. In other words, customers should adjust the number of packets to send (i.e. their demand) for the next time interval as specified in (5.5).

$$x = \frac{1}{4p^2} \tag{5.5}$$

This implies that customers should adjust their demand inversely proportional to the clearing price, which makes sense. Additionally, in our implementation, we introduce a constant factor to (5.5) to normalize the units.

We have defined the procedure of finding the best value of x for the customers. However, it might be the case that x is not *affordable* with the current budget of the customer. So, the customer has to take the budget constraint into account. Once the customer finds the best value of x, she can now see whether she can afford sending that amount of packets with the current clearing price. If her budget is not enough for that, then she just decreases the value of x to the appropriate

value, i.e. $x = \frac{Budget}{p}$. Finally, after finding the affordable value of x, the customer bids randomly between the clearing price and the maximum bid value that she can afford. Actually, this bidding procedure is an issue, but we propose the use of random bidding within the affordable range of the customer for simplicity. A more detailed discussion of our implementation is available in the tech report [15].

7. Performance Analysis and Comparisons

We present a preliminary comparative evaluation of Dynamic Capacity Contracting (DCC) and our smart market implementation. We use simple configurations in our analysis (see Figure 6-1 and Figure 6-2).

The configuration for DCC has a single bottleneck and two edge-to-edge aggregate flows corresponding to two customers. The short-term contracts are set up between the customer and its connected ingress edge router. As described previously, the interior router implements RIO. The observation intervals at the ingress and egress are set to 80 ms, which is the maximum edge-to-edge round-trip time. The duration of the short-term contract is set to 0.40 sec. The measures we use for both schemes are utilization, queue length, relative volume allocations, throughput, goodput and packet loss.

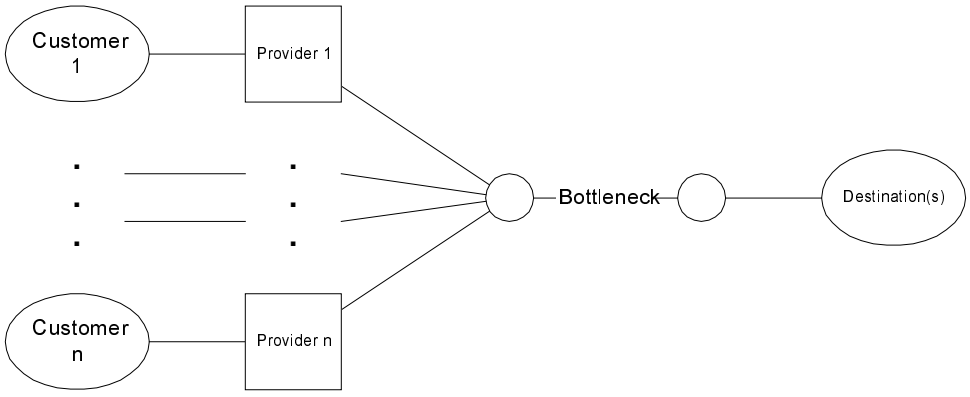

Figure 6-1: Configuration of the experiments made for DCC.

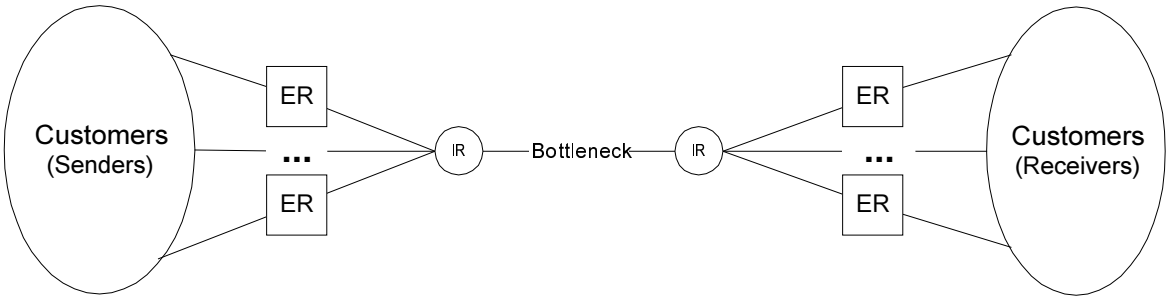

Figure 6-2: Configuration of the experiments made for the Smart Market.

The configuration for the smart market scheme also has a single bottleneck at the link connecting the IRs. Each customer has to access the bottleneck through an ER. The length of the time interval used by ERs and IRs for measurement in the smart market is set to 0.40sec (same as the contract term in DCC). In both DCC and the smart market scheme, the bottleneck rate is 1Mbps, and customers send constant bit rate UDP traffic with fixed packet sizes (1000 bytes). For both of the schemes, we run three experiments with the parameters defined in Table 6-1. 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 different budgets (15, 25 and 35 units per term).

In Figure 6-9 and 6-15, which are utilization, and queue length versus time respectively for experiment 1 of DCC, we observe relatively low and bounded queue lengths and high utilization, which proves that DCC is able to control congestion. From Figure 6-3, which is volume allocation versus time of DCC, we see that volume is not allocated to the two customers strictly proportional to their budgets. In summary, the DCC scheme achieves the congestion management goals of queue length and utilization, but is not very good in terms of economic goals of fairness or relative volume allocations.

Experiment Number	Number of Customers	Budgets of Customers			Simulation Time
		Customer 1	Customer 2	Customer 3	
1	2	60	60	-	4sec
2	2	25	60	-	4sec
3	3	15	25	35	4sec

Table 6-1: Parameters of the experiments.

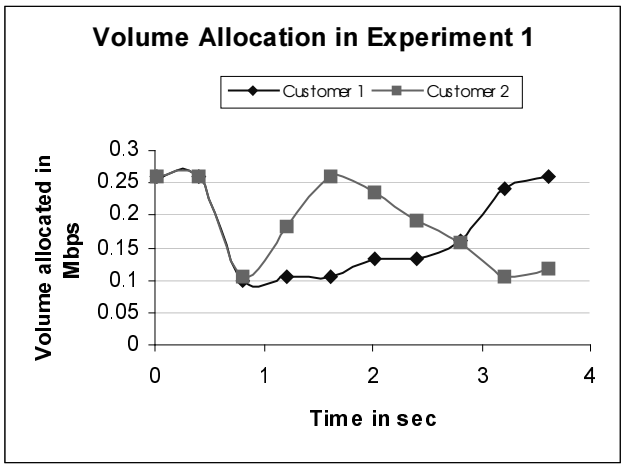

Figure 6-3: DCC Volume Allocation in Experiment 1.

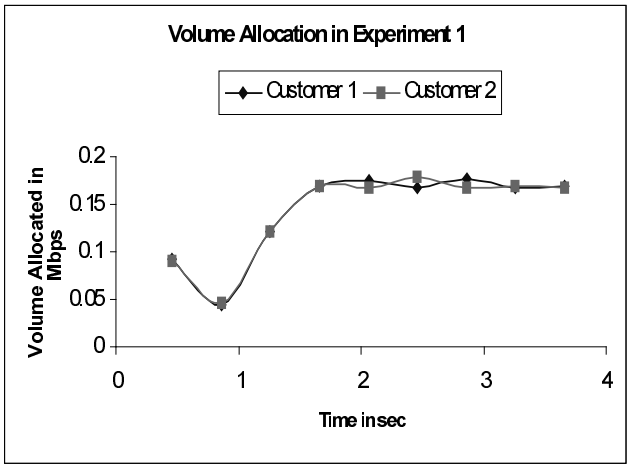

Figure 6-4: Smart Market Volume Allocation in Experiment 1.

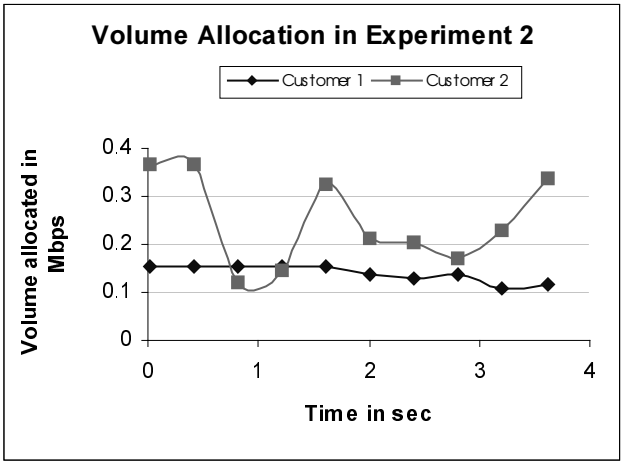

Figure 6-5: DCC Volume Allocation in Experiment 2.

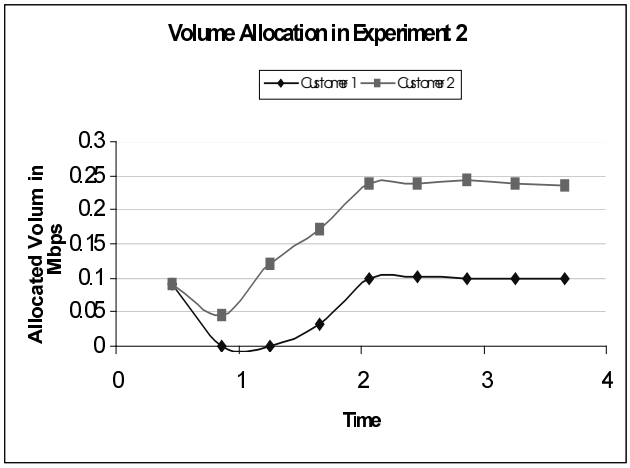

Figure 6-6: Smart Market Volume Allocation in Experiment 2.

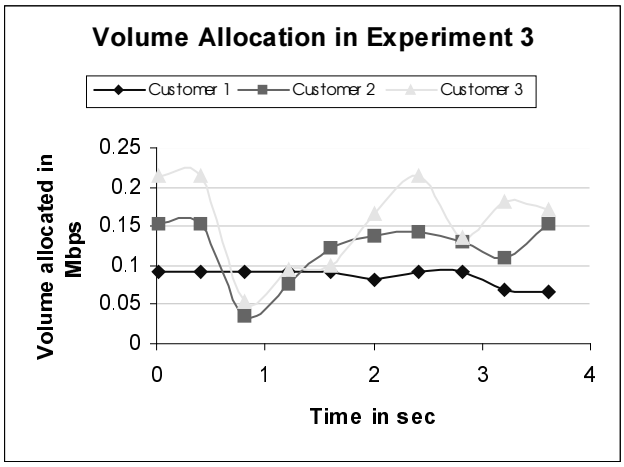

Figure 6-7: DCC Volume Allocation in Experiment 3.

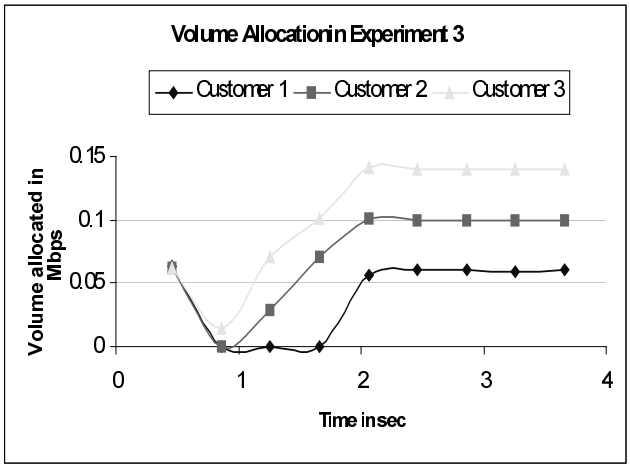

Figure 6-8: Smart Market Volume Allocation in Experiment 3.

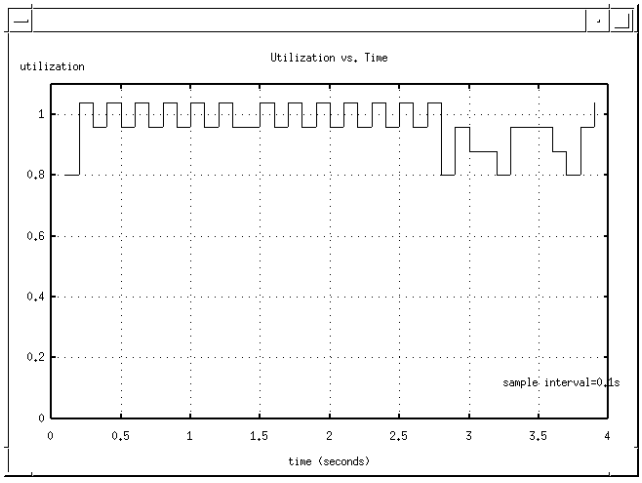

Figure 6-9: DCC Bottleneck Utilization in Experiment 1.

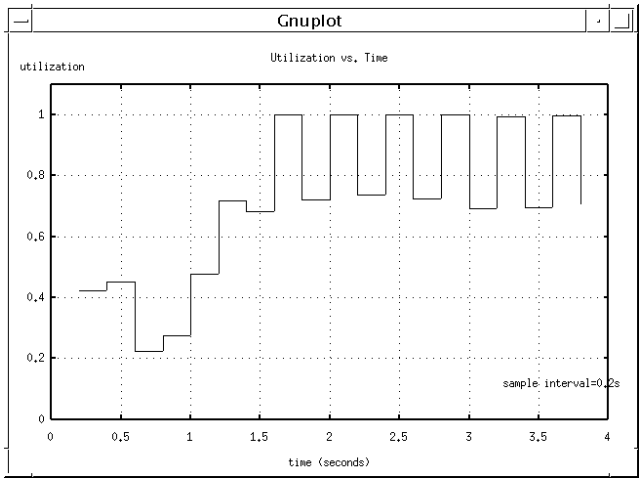

Figure 6-10: Smart Market Bottleneck Utilization in Experiment 1.

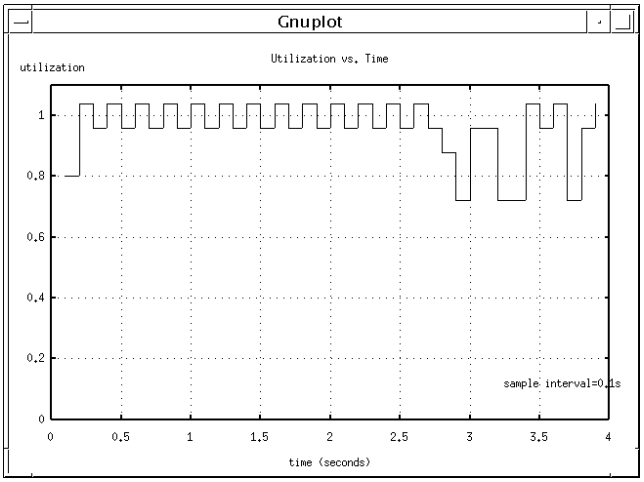

Figure 6-11: DCC Bottleneck Utilization in Experiment 2.

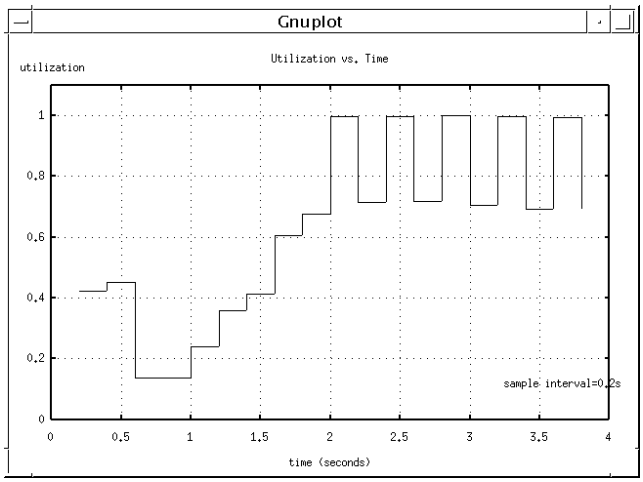

Figure 6-12: Smart Market Bottleneck Utilization in Experiment 2.

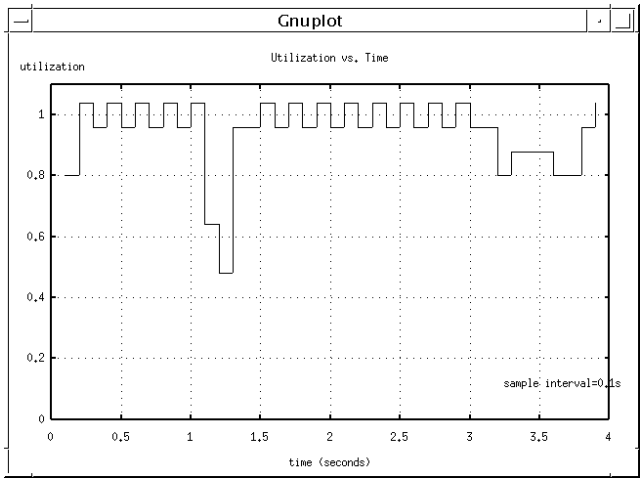

Figure 6-13: DCC Bottleneck Utilization in Experiment 3.

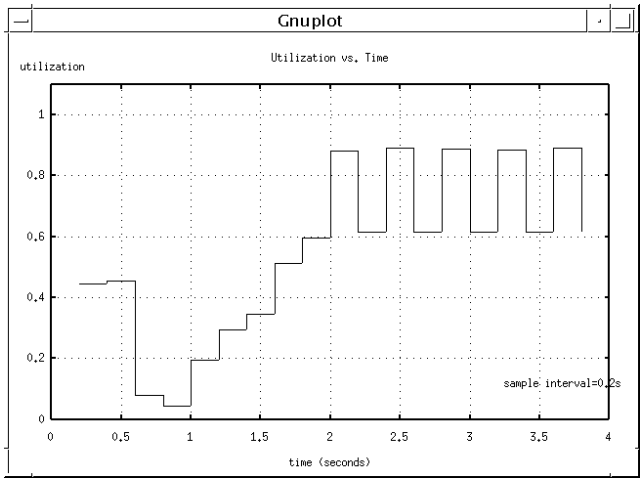

Figure 6-14: Smart Market Bottleneck Utilization in Experiment 3.

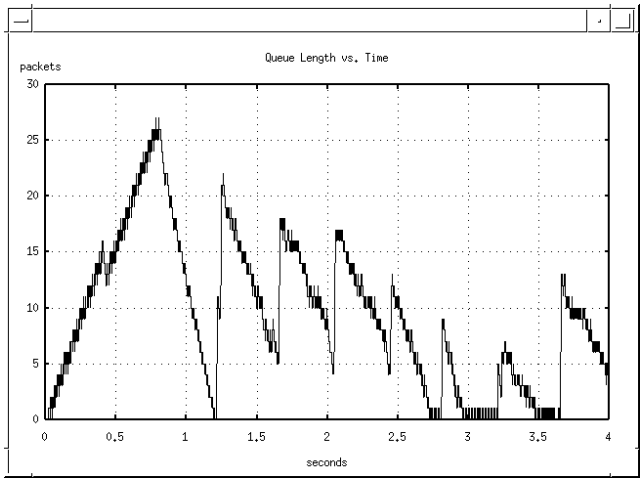

Figure 6-15: DCC Queue Length in Experiment 1.

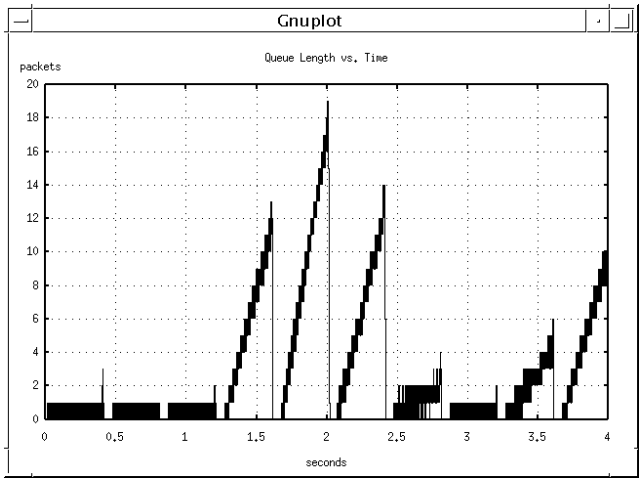

Figure 6-16: Smart Market Queue Length in Experiment 1.

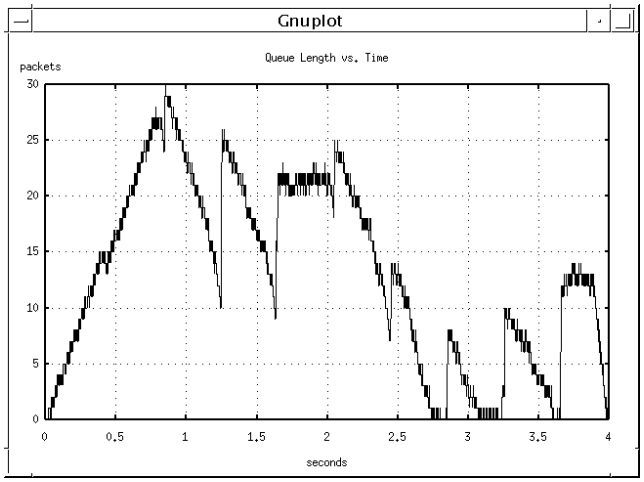

Figure 6-17: DCC Queue Length in Experiment 2.

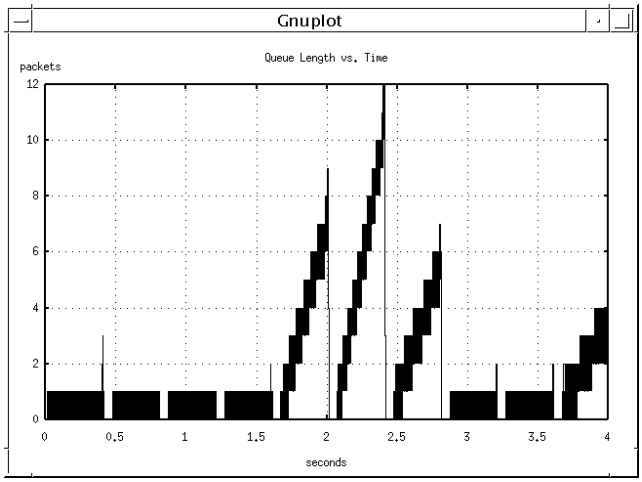

Figure 6-18: Smart Market Queue Length in Experiment 2.

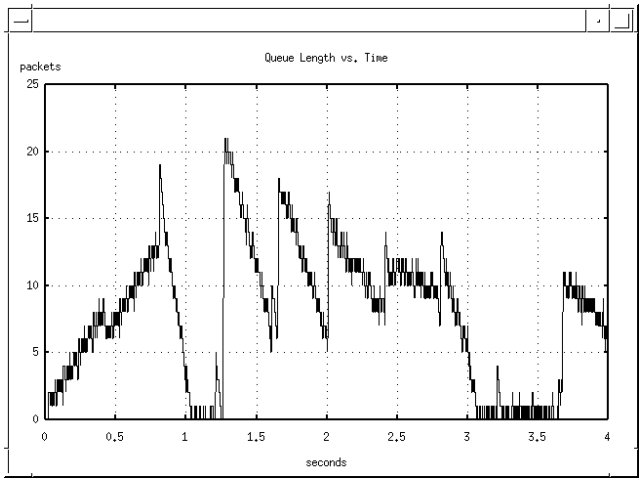

Figure 6-19: DCC Queue Length in Experiment 3.

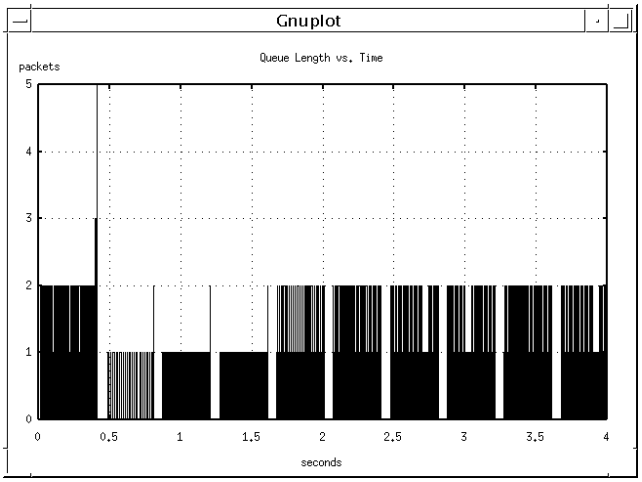

Figure 6-20: Smart Market Queue Length in Experiment 3.

	Dynamic Capacity Contracting			Smart Market		
Experiment	**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 6-2: Performance metrics of the experiments for DCC and the Smart Market.

In Figure 6-10 and 6-16, which are utilization and queue length versus time respectively for experiment 1 of the smart market scheme, we observe that there is a large transient period before steady state is reached. This transient is necessary to damp out the oscillatory behavior caused by parameter sensitivity of MMV's formulas for the smart market. From Figure 6-4, which is volume allocation versus time of the smart market scheme, we see that volume is equally allocated to the two customers. In other words, the smart market scheme is able meet its economic goals of relative volume allocations in proportion to budgets, at the expense of a larger transient convergence time and a low average utilization both during transient and steady state.

By looking at the corresponding graphs of the other two experiments (experiment 2 and 3) we can see a continuation of the same trend in the results. DCC is better from a congestion management perspective because it achieves small queues, high utilization, and a quick convergence to steady state. However, we also observe that volume allocated in case of DCC is not always proportional to budget but this is an area, which points to room for further improvement. The smart market scheme allocates volumes in strict proportion to budgets of the customers and meets the economic efficiency goals, whereas it falters on congestion management goals, especially of high bottleneck utilization.

We have drawn these conclusions by looking at the queue lengths, utilization and volume allocations. The other metrics (throughput, goodput and packet loss), which are presented in Table 6-2, 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.

8. 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 differentiated services 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 deployability. 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 such types of schemes could play a role in transitioning from today's completely flat-priced system towards a system that includes congestion-sensitive pricing for certain classes of service. We have proposed a sample implementation model for the smart market scheme and explore the difficulties and issues in converting this model to implementation. Notably, this scheme requires support in all nodes of the network. Future work will include:

- Expansion of the concept of contracting to point-to-anywhere contracts
- Exploring the concept of bandwidth intermediary to facilitate the mediation between customer and multiple providers by leveraging the dynamic capacity contracting framework. The design of such agents for scalability and integration of policy and budget constraints is also an open topic.
- The basic DCC scheme itself needs to be improved in several dimensions, notably in its relative volume allocations, dynamic estimation of budgets and demands and to support a rich variety of contracts.

References

[1] S. Blake et al, *"An Architecture for Differentiated Services"* IETF Internet RFC 2475, December 1998. Available from ftp://ftp.isi.edu/in-notes/rfc2475.txt

[2] J. Boyle, et al, *"The COPS (Common Open Policy Service) Protocol",* IETF Internet draft. Available from http://www.ietf.org/internet-drafts/draft-ietf-rap-cops-07.txt

[3] R. Braden, D. Clark, S. Shenker, *"Integrated Services in the Internet Architecture: an Overview",* Internet Request For Comments (RFC) 1633, June 1994. Available from http://www.cis.ohio-state.edu/htbin/rfc/rfc1633.html

[4] D. Clark, *Internet cost allocation and pricing, in Internet Economics, Eds McKnight & Bailey, MIT press,* 1997.

[5] S. Floyd, V. Jacobson, *"Random Early Detection gateways for Congestion Avoidance"* IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.

[6] A. Gupta, D. O. Stahl, A. B. Whinston, *Priority Pricing of Integrated Services Networks, Internet Economics, Eds McKnight & Bailey, MIT press,* 1997.

[7] D. Harrison, S. Kalyanaraman, *"Edge-to Edge traffic control: A new overlay congestion control architecture for the Internet"*, available from http://www.ecse.rpi.edu/Homepages/shivkuma .

[8] F. P. Kelly, A. K. Maulloo, D. K. H. Tan, "*Rate control in communication networks: shadow prices, proportional fairness and stability"*, Journal of the Operational Research Society 49 (1998), 237-252. Available from http://www.statslab.cam.ac.uk/~frank/rate.html

[9] J. K. MacKie-Mason, H. R. Varian, *Pricing the Internet, in Public Access to the Internet*, Kahin, Brian and Keller, James, ed., University of Michigan, Boston, Massachusetts, May 1993.

[10] J. K. MacKie-Mason, H. R. Varian, *"Pricing the congestible network resources"*, IEEE J. Selected Areas Comm. 13 (1995), 1141-1149. Available from http://www.sims.berkeley.edu/~hal/people/hal/papers.html.

[11] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, *"TCP Selective Acknowledgement Options"* Internet Request For Comments (RFC) 2018, October 1996. Available from http://www.cis.ohio-state.edu/htbin/rfc/rfc2018.html

[12] K. Nichols, V. Jacobson, L. Zhang, *"A Two-bit Differentiated Services Architecture for the Internet"*, Internet Draft, <draft-nichols-diff-svc-arch-00.txt>, December 1997. Available from http://www.ietf.org/internet-drafts/draft-nichols-diff-svc-arch-00.txt

[13] A. Odlyzko, *"A modest proposal for preventing Internet congestion"*. Available from http://www.research.att.com/~amo/doc/networks.html, 1997.

[14] J. Richard, Edell, P. Pravin, Varaiya, *"Providing Internet Access: What we learn from the INDEX Trial",* submitted to IEEE networks magazine. Available from https://www.INDEX.Berkeley.EDU/reports/99-010W

[15] M. Yuksel, S. Kalyanaraman, *"Implementing the Smart Market"*, Technical Report available from http://www.cs.rpi.edu/~yuksem/SM.doc, 2000.