A Comparative Study of Pricing Strategies for IP Telephony

Matthew Chapman Caesar, Sujatha Balaraman, and Dipak Ghosal
Department of Computer Science
University of California at Davis
Davis, CA 95616
E-mail:{caesar, ghosal}@cs.ucdavis.edu

Abstract—

In this paper we present a comparative study of a few simple but representative usage-based pricing strategies. For the current best-effort Internet, we introduce a QoS sensitive pricing mechanism which takes into account the fact that the quality of received audio degrades as the number of hops traversed by the audio packets increases. We compare the QoS sensitive pricing with flat pricing, congestion sensitive pricing, and a hybrid scheme that combines QoS sensitive and congestion sensitive pricing schemes. Our study is based on a two class user model; type 1 users pay any price for the best QoS and type 2 users request the best QoS at a cost that is less than some maximum price they are willing to pay. Experimental results show the following: i) The QoS sensitive pricing has the lowest blocking probability and also the lowest service distance (the average distance between the client and the servicing gateway) for type 1 calls of any of the schemes. However, it does so by forcing type 2 calls away from the home gateway and hence gives a lower QoS to these calls. ii) The congestion sensitive pricing scheme adapts to the current load at a gateway and hence is good at providing a low service distance to type 2 calls. Unfortunately, it does this by increasing the blocking probability of a type 2 call. iii) The combination of congestion sensitive and QoS sensitive pricing is quite effective and incorporates the best elements of both schemes: it has a very low blocking probability (close to that of QoS sensitive pricing) while retaining a low service distance of type 2 calls. However, at very high loads, the correlation between price and distance breaks down resulting in an increase in the service distance of type 1 calls.

Keywords—

IP Telephony, Gateway location protocol, Congestion sensitive pricing, QoS sensitive pricing, Blocking probability, Revenue.

I. INTRODUCTION

While Internet Telephony (IP Telephony) encompasses many different architectures and services, the key idea is the transport of real-time voice traffic over the Internet. IP Telephony architecture [4] allows the entire end-to-end path to be routed over the Internet. In this case, the endpoints are regular personal computers (PCs) that are equipped with IP Telephony software. IP Telephony architecture also allows one or both the endpoints to be connected to the PSTN (Public Switched Telephone Network). For these cases, a portion of the end-to-end path is routed over the Internet. This requires interoperability between the Internet and the PSTN which is achieved using gateways that act as application level interfaces between the two networks. These gateways are referred to as Internet Telephony Gateways (ITGs) [3].

In the PSTN, voice traffic is carried over dedicated circuits established using the Signaling System Number 7 (SS7) protocol [7]. As a result, the only delay suffered by the voice traffic in the PSTN network is the propagation delay which is fixed once the circuit has been established. On the other hand, the Internet is still inherently a best-effort network and provides no end-to-end bandwidth guarantees. Thus, transporting packetized voice over the Internet can result not only in variable delays but also losses which can result in poor audio quality at the receiver. Furthermore, these losses and delays can increase as the number of hops traversed by the voice packets increase.

One important feature of IP Telephony is that it provides a very rich signaling architecture that can extend up to the endpoints when they are PCs enabled with IP telephony software. This enables new services and as a result Internet Telephony is sometimes referred to as PANS (Pretty Amazing New Services) due to the various new features that it can support [11]. The signalling architecture along with the greater intelligence of the network equipment at the ends of the connection allows service providers to formulate and implement complex pricing models that can take into account both the dynamic congestion at the ITGs and the desired QoS of the call.

Pricing is an important method to implement access control to a limited resource. In the context of IP Telephony, pricing schemes can be employed both to control access to the ITG resources as well to reflect the QoS in the best-effort Internet. In this study, we consider the following simple but representative pricing schemes [9] [5] [6] [12]. First, we consider a flat pricing (FL) scheme in which the ITG advertises a fixed per unit time price for all calls. The second scheme utilizes congestion sensitive pricing (CS) in which the price advertised by the ITG is proportional to the congestion at the ITG. The third scheme is a QoS sensitive pricing (QoSS) policy in which the price is set inversely proportional to the distance in terms of the number of hops between the client and the ITG that services the call. Finally, we consider a hybrid scheme that combines the congestion and QoS sensitive pricing schemes (CSQoSS).

II. IP TELEPHONY ARCHITECTURE

Figure 1 shows a generic IP Telephony architecture and the manner in which it inter-operates with the PSTN system. The problem of selecting between ITGs is solved by the Gateway Location Protocol (GLP). The key entities involved in the GLP [4] include the Administrative Domain (AD), the User Agent, the ITG, and the Location Server (LS). The Internet is viewed as a collection of Administrative Domains (ADs) that are connected by multiple backbone networks. Each AD contains one or more ITGs, one or more LSs, and User Agents, which are customers that initiate calls. The Location Server (LS) [4] receives

---

This research was supported through NSF grants NCR-9703275 and ANI-9741668.

0-7803-6454-6/00/$10.00 (C) 2000

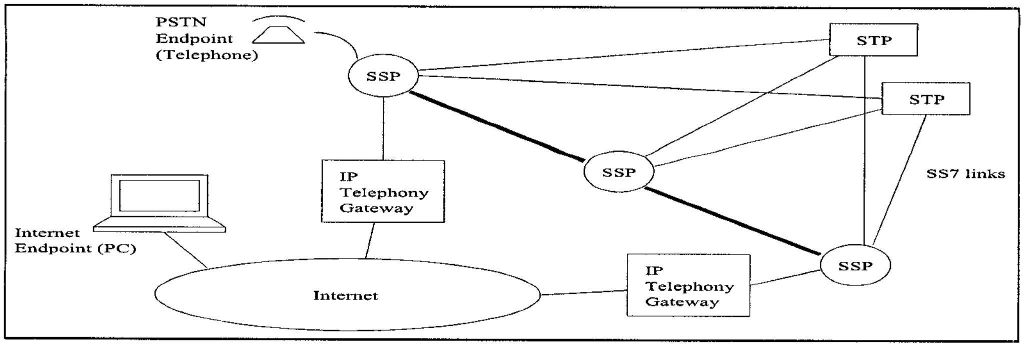

Fig. 1. A generic IP Telephony architecture.

information about ITGs from other LSs and sends information to other LSs about ITGs it knows about[2]. ITG information is propagated between these LSs in the form of ITG objects which contain multiple attributes including a range of reachable numbers and the next hop IP address. Additional information like cost, features supported, protocols, and codecs supported may also be part of the ITG object. Each LS maintains a database containing information about all the other ITGs.

III. PRICING SCHEMES

In this paper, we study a number of simple but representative pricing schemes[1]. The pricing schemes that we describe below operate at the call level, in the sense that negotiations between the user and ITG are made only once per call during the call setup phase [13]. Furthermore, all the pricing schemes considered in this study incorporate usage-based pricing. This implies that total cost of the call depends on the duration of the call[2]. Following are various pricing schemes considered in this study.

A. Flat Pricing (FL)

In this scheme the ITG advertises a fixed per unit time charge for a call. The cost per minute advertised by the ITG is the same and is set at 10 cents per minute. Since the average call holding time is assumed to be a negative exponential distribution with a mean of 3 minutes, the average cost of each call is 30 cents[3]. The ITG sends out updates about trunk state information only when it changes between the full and non-full states as opposed to sending out trunk state information at every state change.

B. QoS Sensitive Pricing (QoSS)

The longer the path traversed by a packet in the IP network, the greater is the delay and its chances of being dropped in the network. This implies the larger the distance (in terms of Internet hop counts) between the client and the servicing ITG, the lower will be the quality. The audio quality may also be reduced because of jitter, protocol errors, codec delays, and other factors that increase with distance [12][14]. The above factor is taken into account in the QoSS scheme, and the price advertised by the ITG is discounted for clients that are further away from the ITG[4]. We considered a topology in which the hop count between adjacent ITGs is uniformly distributed in the range (1..3). Before sending out an update to the LS in the neighboring AD, the LS decrements the advertised cost by a value of 1.5 cents per hop. The maximum advertised cost of a call is 45 cents at the originating ITG and the price is decremented up to a minimum of 15 cents. The ITG advertises only when the trunk state changes between full and non-full states.

C. Congestion Sensitive Pricing (CS)

In this scheme, the price changes depending on the congestion at the ITG which is measured in terms of the average number of trunks in use. The utility function used by the gateway is described by the price-congestion curve shown in Figure 2. In this scheme trunk state information is sent out whenever the number of busy trunks crosses any of the price change thresholds shown in Figure 2. The ITG advertises a lower price when it has many trunks free, i.e., when it is under-utilized, and increases the advertised price as the number of trunks in use increases. Congestion sensitive pricing is implemented in PSTN in the form of time-of-day pricing. However, unlike in PSTN, congestion sensitive pricing in IP Telephony can be implemented much more dynamically.

An important aspect in this scheme is the manner in which the congestion is measured. A typical approach is the exponential averaging scheme [10]. Let M be the average number of trunks in use. If N is the current observed value, the average number of trunks in use is given by

$$M = \alpha M + (1-\alpha)N \qquad (1)$$

where α, which is referred to as the filter gain, is a constant between 0 and 1. M is recalculated every time a new connection is made to or dropped from the ITG. Based on the above definition of congestion, a utility function is used to specify how price changes with ITG congestion.

---

[1] Our implementation assumes a cooperative environment, in which a single pricing scheme may be implemented over all nodes in our network. We felt this was a useful environment to simulate, as this would correspond to an IP Telephony service provided by a large ISP with points-of-presence in a number of different geographical areas.

[2] It is important to note that when the call is made from a PC, one can imagine that the caller may be provided real-time information on the cumulative cost based on which the caller may limit the duration of the call [6]. In this study, we do not consider such a scenario.

[3] In our experimental setup the ITG advertises the total cost of the call assuming an average of 3 minutes call holding time.

[4] Note that while there is distance pricing in the PSTN, it is opposite of what we have considered here. In the PSTN, the larger the distances between the endpoints the higher the price. This pricing reflects the fact that infrastructure cost of the PSTN system is very high.

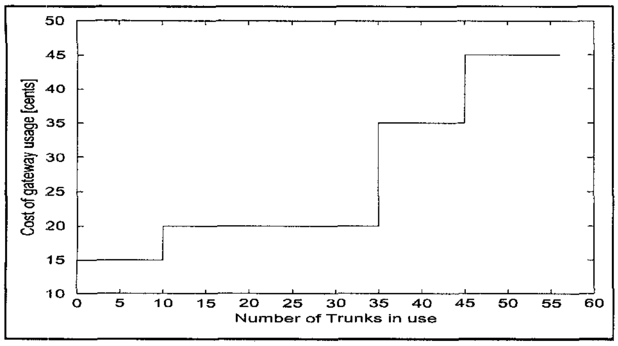

Fig. 2. Price curve used for CS.

D. *Quality of Service and Congestion Sensitive Pricing (CSQoSS)*

This scheme combines the features of CS and QoSS. Each ITG advertises the cost at regular intervals depending on its current load as shown by the price congestion curve in Figure 2. The cost is discounted by a function of the hop distance between adjacent LSs as in QoSS. As a result, the cost in this scheme reflects both the congestion at the ITG as well as the number of hops between the ITG and the client.

IV. THE EXPERIMENTAL ARCHITECTURE

The experimental setup used in our study adopts the architecture proposed in the GLP based on Border Gateway Protocol (BGP) [1]. Our overall architecture consists of 10 ADs. Each AD, as shown in Figure 3, consists of an ITG module, a LS module, and a user module. A single workstation was abstracted as an AD and all the modules were implemented as separate processes running on the same machine that interact with each other using the UNIX message queue (MQ) [8]. The experiments were conducted on 10 HP-Unix workstations connected to a Local Area Network (LAN).

A. *Gateway Module (ITG)*

The advertisements sent out by an ITG to the LS consists of a 5 tuple which includes i) phone numbers serviced by the ITG, ii) IP address of the ITG, iii) the AD identification number of the ITG, iv) number of available trunks (we assume that there are totally 56 trunks), and v) the current price of a call offered by the ITG. When these advertisements are sent out depend on the pricing scheme. We assume that a call can be serviced by any ITG. Furthermore, we assume that, unlike in PSTN, there is no difference between local calls and toll calls.

The user module connects with the ITG to establish a call. The ITG accepts an incoming call only if it has an available trunk. Otherwise, the user is informed and the call is dropped. On a successful connection setup, the user module and the ITG use a Real-Time Transport Protocol (RTP) [10] session to exchange voice packets. The ITG module also terminates the connection when the user completes the call and releases the trunk.

B. *Location Server Module (LS)*

The LS module is split up into a server module and a client module. The server module maintains information about all the ITGs, both local and remote, in a database. This information includes the price which an ITG is willing to offer a potential customer, the number of free trunks, and the number of hops between the LS and ITG. The database is built by the server from the advertisements that are received from the local ITG and from other LSs. The server module in the LS also receives requests from the user module for translating phone numbers to an appropriate ITG identified by an IP address. The server module looks up the information in its local database and sends back a suitable reply to the user request. Since this database is only updated via periodic advertisements, it is important to note that this database may not reflect up to date information about the ITGs represented therein.

C. *Client/User Module*

The client/user module simulates users who generate requests for call setup to particular phone numbers. We assume the call arrival process in each domain follows a negative exponential distribution with rate λ calls per second. We vary λ to study the sensitivity of the results under different offered loads. The call holding time is also follows this distribution with mean 180 seconds[1]. Each of these child processes then sends a request to the local LS to obtain the IP address of the ITG that will service this call. The client then tries to connect to the ITG, following the appropriate session initiation protocol. If the call is accepted, an RTP session is used to transfer voice traffic between the ITG and the client. When the call terminates, the ITG trunk is freed.

V. EXPERIMENTAL SETUP AND PARAMETERS

In this section, we present the user and server models, the details of the implementation of the various pricing models, and the performance measures used in comparing the various pricing schemes.

A. *User and Server Models*

This experimental study is based on a user model consisting of two types of users. Type 1 users are willing to pay any cost, but require the best quality of service. This implies that the servicing ITG should be as close as possible (in terms of hop count) from the user initiating the call. When a client makes a type 1 request, the LS will attempt to satisfy it by returning the IP address of the closest ITG that has trunks free. Type 2 users expect to get the best quality of service that is available for some maximum price that they are willing to pay. During connection setup, the user offers a particular price uniformly distributed in the range [15..45] and the LS finds the closest ITG that is offering the highest price equal to or less than the price offered by the user and is willing to service the call.

B. *Performance Metrics*

To compare the various pricing policies we use the following performance measures.

1. Call Blocking Probability: This is measured as the ratio of the total number of calls blocked to the total number of calls attempted. A call can be blocked due to two reasons:

(a) The call could be blocked at the LS. This could happen if the LS cannot find an ITG that can service the call. Note that the call could be blocked either because all ITGs are busy, i.e., no

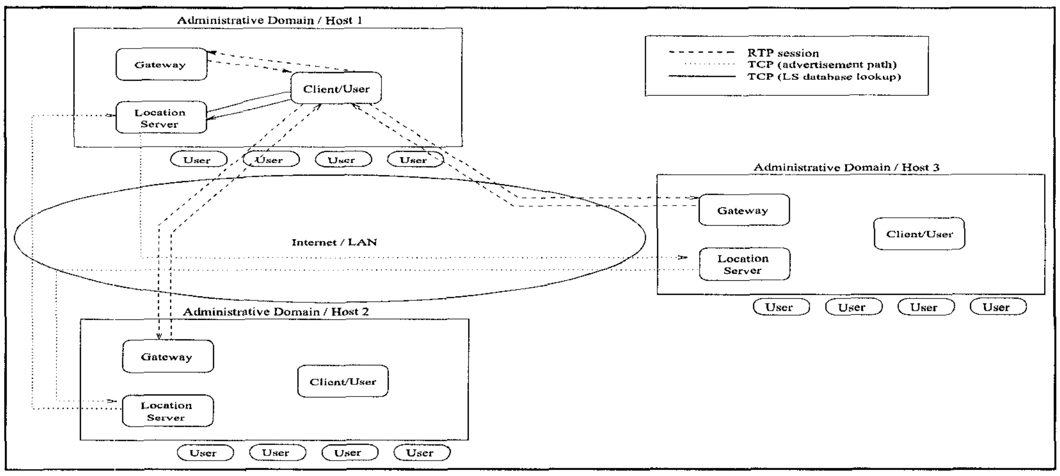

Fig. 3. The experimental setup.

free trunks at the ITGs (NFT), or because there is no ITG that is willing to service the call at the price offered by the user (OPR).

(b) The call could be blocked at the ITG (GRC). This could happen if the LS has outdated/incorrect trunk state information about the ITG. Thus, the LS initiates a call setup to an ITG which actually has no free trunks.

2. Service Distance: This is the average distance in terms of the number of hops between the user and the ITG that services the call.

3. Revenue: This is the sum of the revenue for each call summed over all the ITGs.

VI. RESULTS AND DISCUSSIONS

A. Blocking Probability

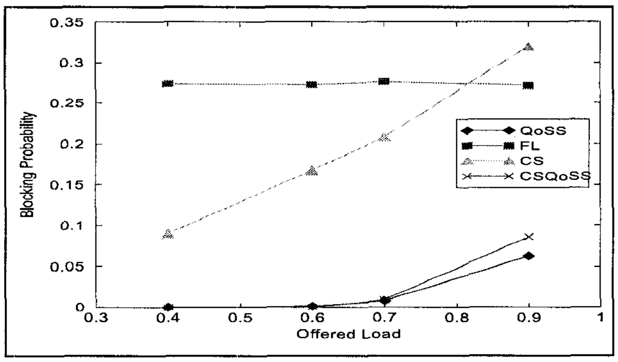

Fig. 4. Average blocking probability as a function of offered load.

Figure 4 compares the probability that a call will be dropped in our system under different system loads for each pricing scheme. Table I tabulates the various components of the blocking probability for different loads.

In the FL scheme the blocking probability of a call being dropped is about 27%, regardless of the offered load. Since the price of all calls are fixed at 30 cents and the price bid by a given type 2 call is uniformly distributed between 15 and 45, half of the type 2 calls are being dropped in this scheme. Since type 2 calls comprise half of the total number of calls, we see approximately 25% of all calls are being dropped. As a result, the effective trunk utilization is low and there are always free trunks available for type 1 calls. Hence, all calls dropped in this scheme are dropped due to overprice as can be seen in Table I.

In CS, the price varies dynamically with load. For low loads, the average price is low, allowing the system to satisfy more type 2 calls than in FL. As the system load increases, the number of calls dropped increases. As can be seen from Figure 4, the blocking probability for CS exceeds that for FL at a load of about 0.8. According to the utility function used for CS in this study (see Figure 2), the price exceeds FL's price of 30 cents when the average number of trunks in use becomes greater than 45, i.e., an offered load is greater than 45/56 = 80.4%. Table I shows that, as in FL, a significant portion of type 2 calls are blocked, and hence virtually no calls are dropped due to not enough trunks being free.

QoSS has the lowest blocking probability of all four schemes for any offered load. In fact, the blocking probability is almost zero for load less than 0.7. Our policy dictates that the price of a call at each node[5] is 45 cents, but for calls originating in other nodes the price is decremented by 1.5 cents per hop. Since we have 10 nodes in our system and the number of hops between nodes is an uniformly distributed integer in the range [1..3], there is always at least one ITG offering a price of 15 cents regardless of where a call originates. Hence, most type 2 calls are serviced. Furthermore, since the gateway picks the highest offered price in the network below the type 2 call's bid, type 2 calls are evenly distributed among the ITGs. As a result, the ITGs are load balanced with respect to type 2 calls and type 1 calls can typically be satisfied locally. Thus, in this scheme, the trunk utilization is very high and few calls are dropped because of no free trunks being available in the system.

As load increases, we find that statistical fluctuations cause all trunks at certain nodes to fill up due to sudden influxes of type 1 calls at those nodes. As can be seen in Table I, this change in system state is reflected both as overprice drops (OPR) and drops due to no free trunks available in the system (NFT). This can be explained by noting that the system can exist in one of

---

[5] We will use the word node to refer to an ITG or an AD, depending on context.

TABLE I

COMPARISON OF DIFFERENT TYPES OF BLOCKING PROBABIL ITIES (P(OPR/NFT/GRC) → FRACTION OF BLOCKING PROBABILITY DUE TO OPR/NFT/GRC

Pricing Strategy	Load = 0.4		Load = 0.6		Load = 0.7		Load = 0.9	
	P(OPR)	P(NFT)+ P(GRC)	P(OPR)	P(NFT)+ P(GRC)	P(NFT)	P(OPR)+ P(GRC)	P(OPR)	P(NFT)+ P(GRC)
FL	0.273766	0	0.272511	0	0.276497	0	0.271299	0
QoSS	0	0	0.000705	0.000846	0.001501	0.006502	0.022505	0.038989
CS	0.089851	0.00022	0.167094	0.000285	0.207998	0.00026	0.379687	0.000202
CSQoSS	0	0	0	0.000563	0.000605	0.009081	0.023752	0.062298

the following three states: (1) When there is at least one ITG with no free trunks and at least one ITG with one or more free trunks. In this case no incoming type 1 call will be dropped. (2) When none of the the ITGs have free trunks. In this case all incoming calls will be dropped. (3) When all ITGs have one or more trunks free. In this case no type 1 calls will be dropped.

OPR drops of type 2 calls can only occur when the system is either in State 1 or State 3. Also, drops due to NFT can only occur if the system is in State 2. The system spends most of its time in State 3, and judging from the relative frequency of OPR and NFT type drops we can see that it spends more time in State 1 than State 2 (as GRC drops form a very small percentage of the blocking probability).

Under high load the blocking probability for CSQoSS is higher than that of QoSS (by about 30%). The reason for this is because of the way we chose to combine CS and QoSS. Recall that we select a starting price for any node K to be defined by the price curve as used in CS. We then decrement the price for other nodes by 1.5 cents per hop from K. Under low load, we will be able handle all type 1 calls, just like in QoSS. However, the big difference here is that the highest price in our system will still be quite low: for load = 0.4 it will be only 20 cents. Hence we will be allowing more type 2 calls to be handled locally.

Under high load the system state behavior of CSQoSS is similar to that of the QoSS scheme. However, unlike QoSS, if a statistical fluctuation suddenly drops the load at a certain ITG, the advertised price will drop and more type 2 calls will be directed toward this node. In this manner, type 2 calls now have a method to "sneak in" and acquire a set of trunks. In QoSS we are imposing a strict correlation of hop count and price charged to the customer. In CSQoSS this correlation breaks down and type 2 calls are served with a low service distance at a low price. This makes an ITG unable to handle any spikes in incoming traffic: any drops in load will be quickly replaced by calls from other ITGs. However, any sudden increase in load quickly saturates all trunks in the ITG. Note this will not happen in QoSS, where any drops in utilization at a node free up a set of trunks which are then available in case any sudden spikes occur in incoming traffic to the node. As can be seen in Table I, this is indeed what is happening: the system becomes more likely to enter State 2; thereby causing more calls to be dropped due to NFT.

B. Service Distance

Figures 5 and 6 show the average service distance for type 1 and type 2 calls and further serve to clarify the discussion in the previous section. In FL the average service distance for both type 1 and type 2 calls is zero. Since many type 2 calls are being dropped due to OPR, we can easily satisfy the remainder of both types of calls at the home ITG.

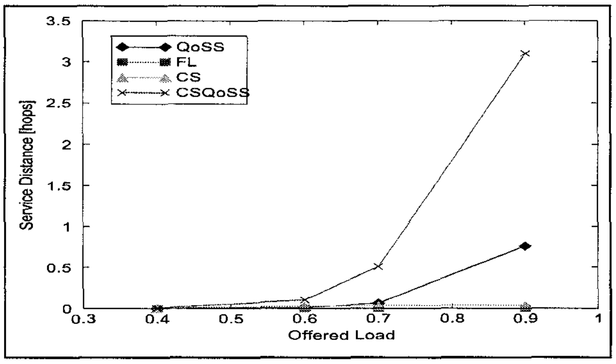

Fig. 5. Average service distance of Type 1 calls as a function of load.

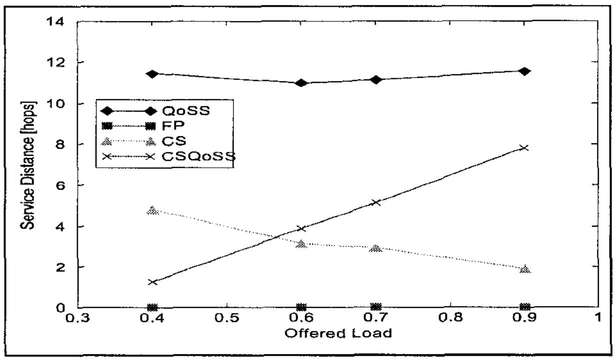

Fig. 6. Average service distance of Type 2 calls as a function of load.

In CS, the average service distance is quite low for both types of calls. Again, we are dropping a significant portion of the type 2 calls, therefore there are many trunks available to service incoming type 1 traffic as well as type 2 traffic with a sufficiently high bid at the home ITG. One may note that the service distance of type 2 calls decreases with increasing load. This is because we have chosen to implement type 2 traffic such that it chooses the maximum price smaller or equal to that of the user's bid. Hence, under low load, the first ITG that begins advertising a price higher than 15 cents will have the majority of the traffic directed at it. Under higher load the picture is different: there is likely to be a good spread of prices so type 2 calls are more likely to be satisfied locally.

The characteristics of the service distance in QoSS is substantially different from that of CS. Like CS, we see a very low type 1 service distance. However, the service distance of type 2 calls are very high. This is because the type 2 calls are uniformly distributed in the network. The tradeoff between CS and QoSS is the following: the CS scheme offers a lower average total service distance at the expense of a higher blocking probability. From Figure 6 we see that the average service distance for type 2 calls remains fairly constant across load. This is because price does not vary as load changes. At high loads we notice an increase in service distance of type 1 calls. This results from the fact that the home ITG becomes fully utilized and calls must be forwarded to neighboring ITGs.

It is quite interesting to see that CSQoSS has a better service distance for type 2 calls than QoSS (and a better blocking probability than that of CS). The large service distance for type 1 calls (and the comparatively small service distance for type 2 calls) show that some type 2 calls displace type 1 calls, thereby causing a lower QoS for those type 1 calls.

C. Total Revenue

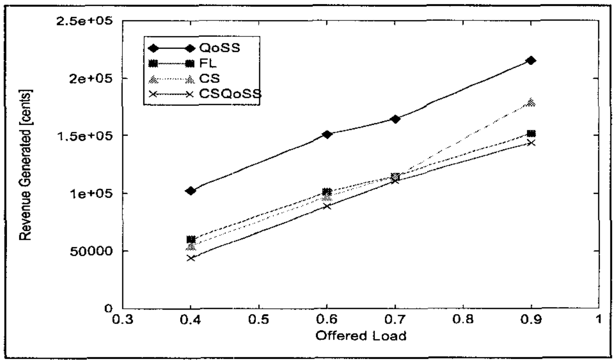

Fig. 7. Total revenue as a function of load.

In Figure 7 we compare the total revenue generated from each of the four schemes. For FL, the revenue increases linearly with load. This is not surprising to find since in this model prices do not fluctuate with load: since a call is equally likely to be dropped in this scheme, the total revenue will be a function strictly based on the number of calls coming into the system. In CS we find a similar effect: the more traffic coming into the system increases the number of calls that are handled and we therefore see an increase in the total revenue generated by the system. The total revenue generated by this scheme is higher than that of FL for high load because we are increasing the price of a call at each of the ITGs. The total revenue is lower than that of FL for low load for the same reason. Interestingly enough, QoSS gives the highest total revenue of all the four schemes. This is because the system is always providing a wide range of prices for incoming calls. For example, if a type 1 call comes in it will likely be handled at the home ITG which will allow the system to charge the maximum possible amount for the call. Further, most type 1 calls will not be dropped because there should be many trunks free at the home ITG to handle these calls, as previously mentioned. In addition, we are dropping very few type 2 calls, and of those which are not dropped we can charge a price very close to the bid (rounded down to the closest tier in the price curve). CSQoSS provides a smaller revenue for the system. As explained before, this is because the cost/distance relationship breaks down in this model. We are allowing more type 2 calls to take place at the home node, thereby displacing type 1 calls and decreasing the amount we charge for them. Since type 1 calls are willing to pay any price, we want to handle them at the home ITG in order to generate maximum revenue. CSQoSS handles this poorly by forcing these calls away and allowing type 2 calls into the home ITG.

VII. CONCLUSION

The results show that QoS sensitive pricing maximizes resource utilization while providing higher QoS to user who are willing to pay for it. CSQoSS did not perform as well as plain QoS sensitive pricing, particularly at high load. The key observation was that random fluctuations in traffic can break down the correlation between cost and distance imposed by the plain QoS pricing. This results in poor QoS to user who are willing to pay higher price.

There are many items that need further investigation. Firstly, there are different ways to combine CS and QoSS into a single pricing model; these approaches need to be compared to determine their relative strengths and weaknesses. Secondly, the study needs to be extended to realistic network topologies. Thirdly, the study needs to be carried out using more realistic user models. Finally, it would be of some interest to simulate a competitive network where pricing schemes can differ at each node.

REFERENCES

[1] D. Hampton, D. Oran, H. Salama, and D. Shah, "The IP Telephony Border Gateway Protocol Architecture," *Internet Draft*, Internet Engineering Task Force, February 1999, Work in progress.
[2] J. Rosenberg and H. Schulzrinne, "A Framework for a Gateway Location Protocol," *Internet Draft*, Internet Engineering Task Force, June 1999, Work in progress.
[3] J. Rosenberg and H. Schulzrinne, "Internet Telephony Gateway Location," *IEEE INFOCOM*, Francisco, California, March/April 1998.
[4] J. Rosenberg and H. Schulzrinne, "The IETF Internet Telephony Architecture and Protocols," Technical Report, Columbia University, 1999.
[5] J. Walrand and P. Varaiya, *High-Performance Communication Networks*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.
[6] R. J. Edell and Pravin Varaiya, "Providing Internet Access: What we learn from the INDEX Trial," Index Project Report 99-010W, University of California, Berkeley, April 27, 1999.
[7] T. Russell, *Signalling System Number 7*, McGraw-Hill Series on Computer Communications, 1995.
[8] W. Richard Stevens, *Unix Network Programming, Volume 1, Networking API: Sockets and XTI*, Prentice-Hall, 1998.
[9] G. Huston, *ISP Survival Guide: Strategies for Running a Competitive ISP*, Wiley, 1999.
[10] J. Kurose and K. Ross, *Computer Networking, A Top-Down Approach Featuring the Internet*, Addison-Wesley, 2000.
[11] H. Schulzrinne (Edited), Digest of papers in Special Issue on Internet Telephony, *IEEE Network Magazine*, May/June 1998.
[12] A. Rayes and K. Sage, *Integrated Management Architecture for IP-Based Networks*, IEEE Communications Magazine, April 2000.
[13] B. Li et. al., *QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges*, IEEE Communications Magazine, April 2000.
[14] M. Hassan and A. Nayandoro, *Internet Telephony: Services, Technical Challenges, and Products*, IEEE Communications Magazine, April 2000.
[15] D. Bergmark and S. Keshav et. al., *Building Blocks for IP Telephony*, IEEE Communications Magazine, April 2000.