A Pricing Policy for Scalable VOD Applications[1]

A. Krishnamurthy, T.D.C. Little, and D. Castanon

Multimedia Communications Laboratory
Department of Electrical, Computer and Systems Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
tdcl@bu.edu

MCL Technical Report 05-01-1995

Abstract–Many video applications are scalable due to human tolerance to degradation in picture quality, frame loss and end-to-end delay. Scalability allows the network to utilize its resources efficiently for supporting additional connections, thereby increasing revenue and number of supported customers. This can be accomplished by a dynamic admission control scheme which scales down existing connections to support new requests. However, users will not be willing to tolerate quality degradation unless it is coupled with monetary or availability incentives.

In this paper we propose a pricing policy and a corresponding admission control scheme for scalable VOD applications. The pricing policy is two-tiered, based on a connection setup component and a scalable component. Connections which are more scalable are charged less but are more liable to be degraded. The proposed policy trades off performance degradation with monetary incentives to improve user benefit and network revenue, and to decrease the blocking probability of connection requests. We demonstrate by means of simulation that this policy encourages users to specify the scalability of an application to the network.

Keywords: Video scaling, quality of service, pricing, protocols.

[1]In *Proc. 2nd IEEE Intl. Workshop on Community Networking*, Integrated Multimedia Services to the Home, Princeton, NJ, June 1995, pp. 139-146. This work is supported in part by Motorola through the UPR Program and the National Science Foundation under Grant No. IRI-9211165.

1 Introduction

The evolution of computing and networking technology in recent years has enabled the development and support of exciting new distributed multimedia applications (e.g., video-on-demand, distance learning, and video conferencing) which are anticipated to be available to end users on a large scale. Networks supporting Video-on-Demand (VOD) applications will allow users to retrieve and display huge amounts of video data from distributed file servers and sources in a real-time fashion. There are two approaches to the transfer and play-out of such data: transferring all the data ahead of time and then playing them out from local memory (e.g., as done by the HyperText Transport Protocol), or transferring data continuously while playing them out. The latter approach has many advantages, given the large amount of data most video applications generate, but is also more difficult to implement due to the real-time nature of VOD applications.

A common approach to guarantee adequate quality of presentation during delivery is to reserve sufficient network resources for each individual connection [3]. The problem of efficient allocation of valuable network resources is made significant by the large volume of data coupled with the bursty nature of compressed video. When resources are scarce, the data rate of the connection can be adapted by scaling the data stream [1]. Many video applications are scalable because of human tolerance to degradation in picture quality, frame loss and end-to-end latency, provided the quality of playout is above some perceptual threshold. Tolerance to degradation depends on both the application and the user. Video scalability can be translated to a reduction in resource requirements for the corresponding connections. For example, tolerance to large end-to-end latency allows data to be smoothed by buffering before transmission in the network. Similarly, tolerance to picture quality degradation allows encoding parameters to be modified to yield lower data rates. Thus, the application can specify a range of resource requirements (ideal and minimum acceptable) to the network during connection establishment.

Table 1 shows the ideal and minimum acceptable bandwidth requirements for four 10-minute M-JPEG encoded video sequences with the scaling parameters chosen randomly.[2] Q is the quality compression factor, d is the percentage of dropped frames, d_c is the number of consecutive frame drops allowed, D is the latency (ms), and b_h and b_l are the ideal and minimum acceptable bandwidth requirements (Mb/s). Here, we have assumed that scaling is performed at the source and that bandwidth is the only network resource under

[2]A uniform distribution was used and skewed towards the more probable scaling parameters, e.g., a quality factor in the range 30-125 was chosen with twice the probability as that in the range 125-250.

Table 1: Bandwidth Reuirements for M-JPEG Video Sequences

#	**Q**	**d**	d_c	**D**	b_h	b_l
3	75	7	2	40	3.94	1.16
4	200	12	4	140	3.40	0.45
1	50	2	1	10	3.32	1.59
3	75	6	2	30	3.94	1.20
2	150	11	3	100	2.24	0.55
1	30	1	1	1000	3.32	0.45
1	125	11	3	70	3.32	1.55
2	250	14	4	140	2.24	0.42
4	50	4	1	10	3.40	1.48
3	100	8	2	60	3.94	0.93
4	150	11	3	100	3.40	0.58
3	30	1	1	20	3.94	2.06

consideration; such an assumption is justified for single hop networks with sufficient buffering at the source and destination. We term this range of resource requirements the "admissible region"; if resource availability in the network is greater than the minimum acceptable, the connection can be admitted [4].

The "admissible region" can be translated to network gains by means of a dynamic connection establishment mechanism [4, 5] that allows renegotiation. If sufficient resources are not available to admit a connection, it can be scaled down within the specified range to enable connectivity while providing a quality above the specified threshold. Furthermore, existing connections may be scaled down to free up resources to admit new connection requests. Clearly, the employment of such a mechanism increases network connectivity, utilization and revenue. However, users suffer degradation when the application is scaled down. In the absence of any incentive to specify scalability, users will always demand the best possible quality. Furthermore, the network has no incentive to reserve resources to support connections beyond the minimum specified requirement. Such user and network behavior can lead to inefficient allocation of valuable network resources.

We propose a pricing policy for network resources to overcome these problems. The proposed policy provides monetary incentives to offset performance degradation to the user and makes the revenue earned by the network commensurate with the quality delivered. The policy encourages users to specify application scalability to the network. Furthermore, the network is provided with monetary incentives to support connections at higher than the minimum specified quality when resources are available.

While recent research efforts have focused on solving a number of technological issues in the networking and operating system arenas, surprisingly little work has been reported in the literature on the development of an appropriate pricing structure for scalable VOD services. Research on pricing issues has focussed on both connectionless transfers such as on the Internet [2, 6, 9], and reservation-based connection-oriented transfers [7, 8]. MacKie-Mason and Varian [6] propose a pricing policy which charges more during periods of congestion (when bandwidth is a scarce resource) and very little during periods of light load. Cocchi et al. [2] propose a scheme to maximize user satisfaction in a connectionless environment. In a reservation based connection oriented scheme, prices should be based on the resources reserved, and not on the actual volume of traffic transferred [7].

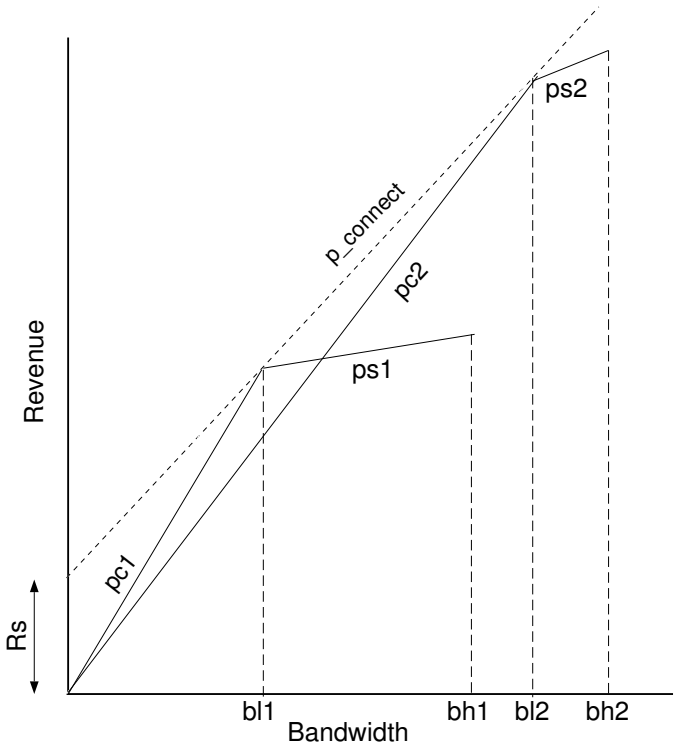
In current literature, scalability and pricing have been studied independently for the provision of integrated services. We contend that these issues complement each other for networks supporting VOD applications. Our study focuses on the relationship between performance and monetary issues from both the user's and network provider's perspectives. The formulation of a pricing policy encompasses consideration of a variety of social, regulatory, economic and performance issues. In our formulation, we concentrate on utilization and performance issues within the network and ignore other factors. We discuss the proposed pricing policy in Section II. Section III describes our simulation models and environment. We present our results in Section IV. Finally, Section V concludes the paper.

2 Proposed pricing policy

The employment of an appropriate pricing policy is essential if benefits from scaling gains are to be utilized. If all users were charged a fixed amount, users will always demand the highest possible quality of service. There is no incentive for them to specify application scalability. On the other hand, the network will always serve each connection at its lowest bandwidth, even if excess bandwidth is available to provide a better QoS. A suitable pricing structure should provide an incentive for the network to scale up connections and utilize excess bandwidth.

2.1 Pricing Policy

The pricing scheme should encourage users to specify the maximum possible scalability to the network when they maximize their individual benefit. The network can then use this scalability to maximize its revenue. Another objective in developing the pricing policy is to decrease the blocking probability of connection requests. A suitable pricing structure should provide an incentive for the network to scale up connections and utilize excess bandwidth. Furthermore, the revenue collected should be proportional to the amount of bandwidth reserved.



Rs - Fixed Connection Set-Up Cost
p_connect - Additional Price for Connection
pc - Price for Connection Admittance
ps - Price for Connection Scaling

Figure 1: Proposed Pricing Model

Considering these objectives, we propose a pricing structure as shown in Fig. 1 which plots the revenue obtained against bandwidth allocations for two connections with different requirements. The pricing structure has two tiers corresponding to connection set-up and scaling. The connection set-up cost consists of two fixed components: a connection set-up

charge (R_s), and a per-unit-bandwidth price ($p_{connect}$). Though the connection set-up prices are the same for all connections, each request sees a different price for connectivity, indicated by p_{ci} in the figure. The price component of the scalable region (p_s) is inversely proportional to the scalability of the connection (i.e., directly proportional to $\frac{b_l}{b_h}$), and is always lower than $p_{connect}$.

2.2 Admission Control

We assume that the network will employ an admission control algorithm to maximize its revenue.[3] In the proposed policy, each connection has two slopes associated with it, one for connectivity (p_c) and one for scalability (p_s). We call these slopes the "connectivity" and "scalability" slopes. Network revenue is maximized if bandwidth is allocated to connections in the higher slope regions. We consider two scenarios of admission control. In the *static scenario*, the admission control algorithm must choose the connections to admit from a group of requests and compute their bandwidth allocations. Note that $p_c > p_s$ always. To implement admission control, the network sorts the p_c and p_s values for the requests in decreasing order and allocates bandwidth starting with the largest value. If the value corresponds to a connectivity price, the connection is admitted with bandwidth b_l. If the value corresponds to scalability, the connection is scaled up to a bandwidth allocation of b_h specified in the corresponding request. This is done until there is no bandwidth to allocate or all connections have been scaled up. Note that this policy is heuristic rather than optimal. However, it is simple and leads to the optimal revenue in most cases.

In the *dynamic scenario*, requests for connection establishment and release are received by the admission control algorithm over time. On receiving a request, the algorithm attempts to admit the connection at the minimum bandwidth. If sufficient bandwidth is not available, the algorithm checks to see if the required bandwidth can be freed by scaling down existing connections. If this test fails, the request is rejected. If it passes, existing connections are scaled in order of increasing scalability slope until sufficient resources are freed. Once the connection is admitted, the algorithm attempts to scale it up at the expense of connections with lower scalability slopes. When a connection is released, existing connections are scaled up in order of decreasing scalability slopes.

[3] And therefore its profit, assuming fixed capacity and cost to provide this capacity.

2.3 Implications

We make the following observations about our proposed pricing structure which relate to the objectives of our pricing model:

- Connections with lower minimal acceptable bandwidth requirements are given higher priorities for connection admission (static scenario).
- Once connected, applications specifying higher scalability are charged a lower price but are also more likely to be scaled.
- Increasing the scalability of an application decreases its blocking probability.
- The network gains by scaling down applications to accept a new request.
- The network gains by scaling up applications when it has unused bandwidth.
- Increasing the price for connectivity (R_s and $p_{connect}$) decreases the blocking probability but may also decrease the revenue earned by the network.
- Users running applications with higher scalability are charged a lower price, while those paying a higher price are supported at high qualities.

3 Simulation Models and Environment

We now describe our models for user utility and cost and simulation scenarios and performance parameters.

3.1 User Utility Function

The *user utility function* is a measure of user satisfaction as a function of allocated resources. We assume a model of diminishing returns; the marginal utility to the user diminishes as a function of allocated bandwidth. User utility is non-zero only when the connection is admitted (i.e., the allocated bandwidth is greater than b_l). Furthermore, the marginal utility is zero when the allocated bandwidth is b_h, i.e., the utility does not increase with increasing

bandwidth allocation at this point. Formally, we define the user utility function for any bandwidth allocation x as:

$$U(x) = \begin{cases} u * (b_h * x - \frac{x^2}{2}) + U_c, & \text{if } b_l \leq x \leq b_h \\ 0, & \text{if } x < b_l \\ u * \frac{b_h^2}{2} + U_c, & \text{if } x > b_h \end{cases}$$

Here, U_c is an additive constant reflecting the utility for connectivity, and u is a constant. An example of a utility function ($U_c = 0, u = 1$) with $b_h = 3.94$ Mb/s and $b_l = 0.96$ Mb/s is shown in Fig. 2.

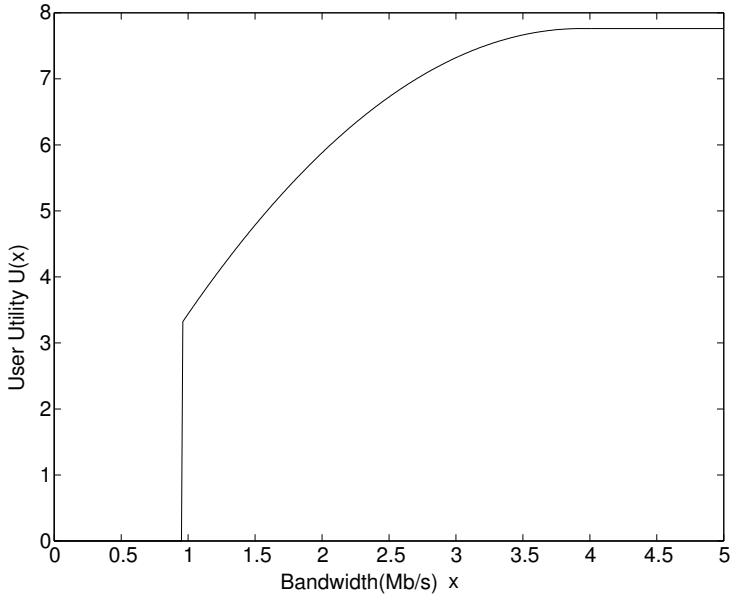


Figure 2: An Example Utility Function

3.2 Cost Functions

The user cost per connection is dependent on the pricing policy used by the network. In a *fixed cost* policy the cost for a connection is fixed and independent of the bandwidth allocated:

$$C(x) = \begin{cases} \text{C}, & \text{if } b_l \leq x \\ 0, & otherwise \end{cases}$$

In the proposed policy the cost consists of three components corresponding to the connection set up cost (R_s), price for connectivity ($p_{connect}$), and price for the scalable region

(p_s):

$$C(x) = \begin{cases} R_s + c * (p_{connect} * b_l + p_s * (x - b_l)), & b_l \leq x \\ 0, & b_l > x \end{cases}$$

where c is a constant.

3.3 Performance Metrics

Our objective is to demonstrate that the proposed policy encourages the user to specify the maximum possible scalability to the network during connection establishment. The user specifies bandwidth requirements so as to maximize benefit, which is defined as the utility derived minus cost paid.

$$B(x) = U(x) - C(x)$$

We aim to show that applying scalability by means of a dynamic admission control scheme leads to significant user benefit, network revenue and connectivity gains over a fixed non-scalable scheme (the fixed cost scheme). We consider percentage of blocked requests, aggregate user benefit and network revenue (aggregate user cost) as performance metrics in our analysis.

3.4 Simulation Scenarios

Simulations were performed in two scenarios. In the static scenario, all requests are assumed to arrive at the same instant. The number of requests was varied over different runs. In the dynamic scenario, requests arrive with a Poisson distribution, the rate of which was varied over different runs. The holding time of connections was fixed at 10 minutes. Performance parameters for the dynamic case were measured per unit time. The b_l and b_h parameters for the requests were chosen randomly from a database similar to the one in Table 1 created in advance. Simulations were performed for the fixed cost policy and our proposed scheme. The parameters for the cost and utility functions were chosen such that the user always benefits (i.e., the benefit is always positive) if the connection is admitted. We set $U_c = 500$ and $u = 12$ per unit time (min). In the fixed cost policy, $C(x) = 500$. For the proposed policy, $R_s = 200$, $c = 12$ per unit time (min), $p_{connect} = 1$, and $p_s = \frac{b_l}{b_h}$. The total available bandwidth was set at 100 Mb/s. The results were obtained by averaging over 100 runs.

3.5 User Preferences

While the network adopts admission control algorithms to maximize its revenue, the user tailors the resource requirement specification to maximize benefit. The user optimizes benefit by demanding bandwidth x such that

$$B'(x_{opt}) = 0$$

$$U'(x_{opt}) = C'(x_{opt})$$

With the fixed cost policy, $C'(x) = 0$, and user utility is maximized when

$$U'(x_{opt}) = 0$$

$$x_{opt} = b_h$$

Thus, the user always demands the maximum bandwidth from the network, and has no incentive to provide a scaling range. In this model, we assume that the user is not influenced by the probability of the request being blocked. When the network is congested, the probability of admission increases with a lower specification of bandwidth. This factor is considered by using a model where the user specifies less than the optimally calculated bandwidth. In the dynamic scenario, the user can optimize if information about the available bandwidth (b_a) is known. If the optimal bandwidth (b_h) is greater than the available bandwidth, the user specifies b_a.

In the proposed policy, the user will specify the scalable range to the network only if benefit is maximized. In a lightly loaded system, all requests are allocated the maximum bandwidth in the range, b_h, irrespective of the size of the range, while the incurred cost decreases with a higher scalability specification. Thus, in a lightly loaded system, user benefit is maximized by specifying the entire range since this minimizes the cost. If the system is heavily loaded, the user tries to minimize blocking probability. This is again achieved by specifying the entire range; a lower value of b_l decreases blocking probability. For intermediate loads, the optimum user specification is not so obvious. If the user ignores the possibility of being blocked and assumes that the bandwidth allocated will always be the minimum specified, the range specified, (b_{min},b_h) is a subset of the entire range (i.e., $b_l \leq b_{min}$), and

$$U(b_{min}) = u * (b_h * b_{min} - \frac{b_{min}^2}{2}) + U_c$$

$$C(b_{min}) = R_s + c * (p_{connect} * b_{min})$$

Note that the higher value in the range is always b_h, since a lower value will increase the cost without changing the utility, thus decreasing the benefit. We calculate the value of b_{min} by

$$U'(b_{min}) = C'(b_{min})$$

$$b_{min} = b_h - \frac{c}{u} * p_{connect} \tag{1}$$

if $b_{min} < b_l$, the user specifies the entire range (b_l,b_h), i.e., $b_{min} = b_l$.

3.6 Admission Control

The admission control algorithm for the proposed scheme has been described in the previous section. In the fixed cost case, the network obtains the same revenue for each connection, independent of the bandwidth allocated to it. The network maximizes its revenue by admitting connections with the smallest bandwidth requirements (b_h) first. To execute this admission policy in the static case, the network sorts the bandwidth specifications of connections in increasing order and admits connections starting with the lowest minimum acceptable bandwidths. In the dynamic case, the connection simply admits connections if it has sufficient bandwidth when the request is received.

4 Results and Discussion

In this section, we present the results of our simulations.

We first compare the proposed policy with the fixed cost policy in the static case. In this simulation, users ask for the maximum possible bandwidth (b_h) in the fixed cost policy, and specify the entire range (b_l, b_h) in the proposed policy. Fig. 3 shows the percentage of requests blocked as the number of requests increase.

We see that the fixed cost policy starts blocking much earlier than our policy. Both policies admit connections at b_h until the bandwidth is saturated. The fixed cost policy cannot admit connections beyond this point and blocks additional calls, while the proposed policy scales down existing connections to admit more connections. Furthermore, we observe that the slope of the curve is greater for the fixed cost policy. The proposed policy can still admit connections after it starts blocking because it admits connections at b_l which can be significantly lower than b_h. This is reflected in Fig. 4 which shows network revenue plotted against number of requests.

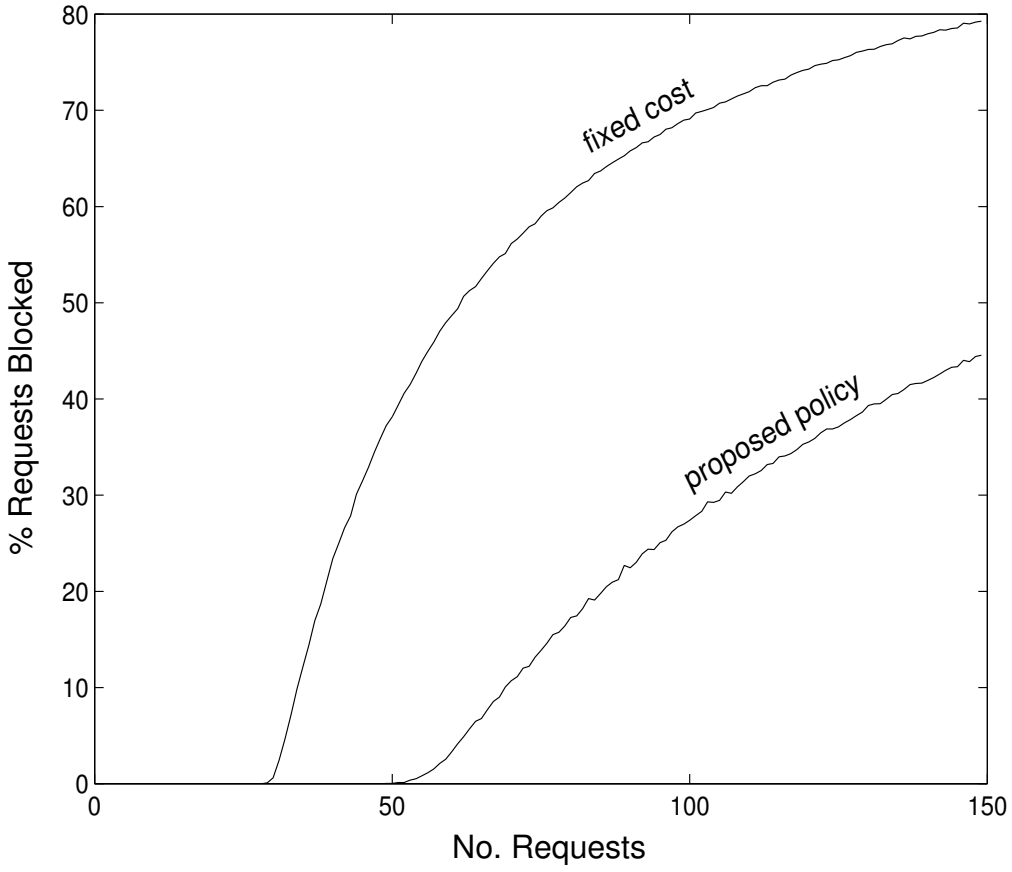


Figure 3: Percentage of Requests Blocked: Static Case

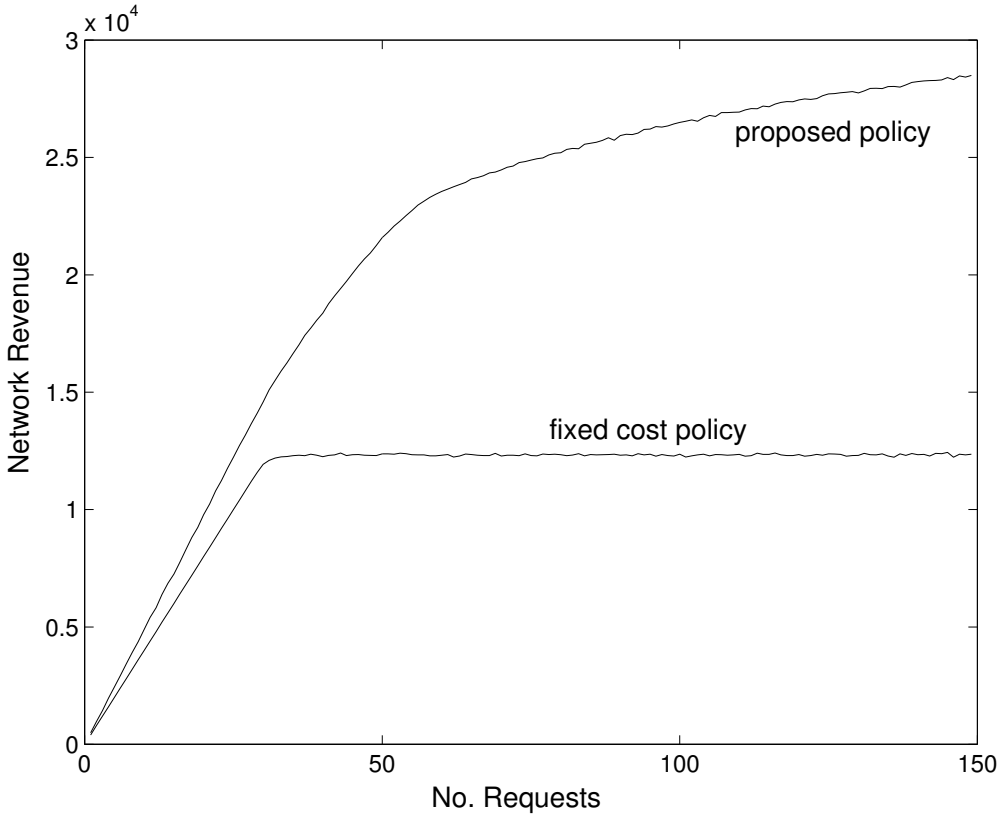


Figure 4: Network Revenue: Static case

Note that we are more concerned with the shapes of the curves than their locations. Either curve can be raised by increasing the cost parameters. The fixed cost curve flattens out when blocking starts, indicating that the number of connections increases only slightly once blocking starts.[4] With the proposed policy, the curve flattens at a higher number of requests (blocking starts later), and increases in the blocking region, indicating that a significant number of requests are still admitted. The loss of revenue due to scaling down connections is more than offset by gains due to admitting more connections. Finally, the user benefit curves are illustrated in Fig. 5.

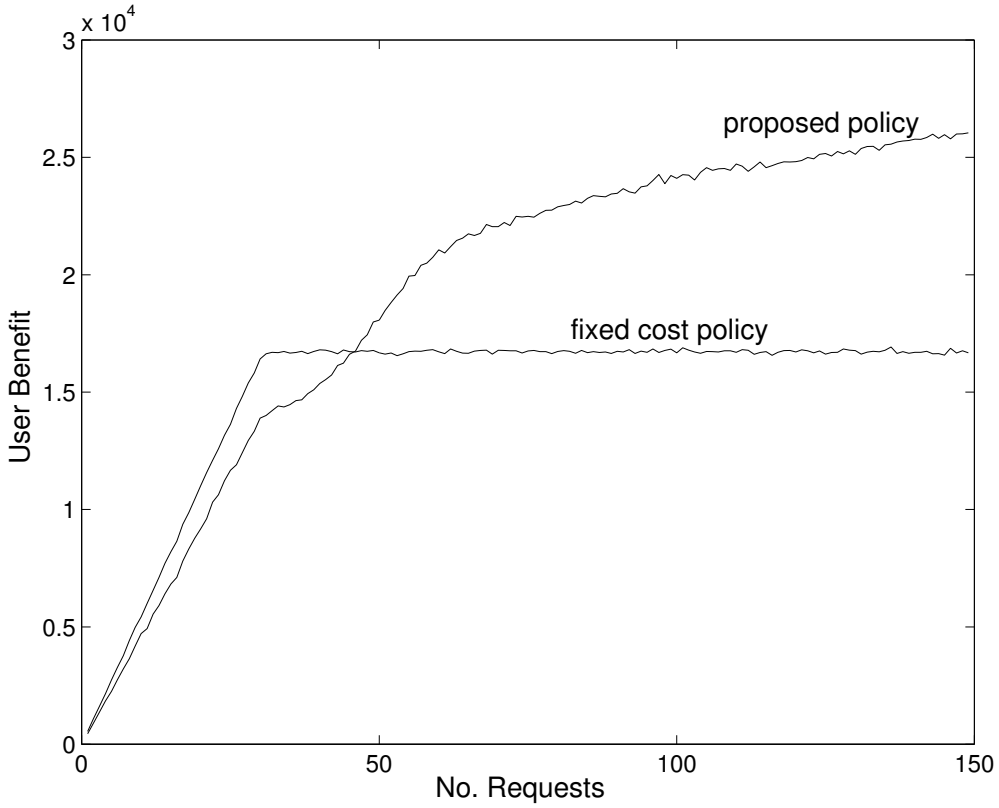


Figure 5: User Benefit: Static case

Again, the user benefit curve flattens out for the fixed cost policy. Note that the benefit may actually decrease in the heavily loaded region because connections with lower bandwidth requirements (and therefore utility) are admitted at the same cost. In our example, the curve remains flat because the lower benefit per connection is offset by an increase in the number

[4]Connections with lower requirements are admitted first, so more connections can be admitted.

of connections. In the proposed policy, the decrease in utility per connection due to scaling is offset by a decrease in cost.

If the user lowers the requirement to increase the probability of admission in the blocking region, more requests may be admitted in the fixed price case. However, the benefit per admitted user decreases because the decrease in utility is not offset due to the fixed cost structure. Depending on the actual cost, users may find it more beneficial not to request the connection, leading to a drop in network revenue. From these results, we conclude that the proposed policy uses application scalability for gains in network connectivity, revenue, and user benefit.

In the above experiments, we assumed that the user always provides the network with the entire range of scalability (b_l, b_h). Providing this range makes the application liable to scaling and consequently to performance degradation. Users will not provide this range unless it optimizes their benefit. In the lightly loaded region, all connections are supported at the b_h, irrespective of b_l. Users therefore maximize their benefit by providing the entire range, because this minimizes their cost. In the heavily loaded region, providing the entire range maximizes the probability of connectivity. In the moderately loaded region, user benefit may not be maximized by specifying the whole range. If users ignore the blocking factor, and assume that bandwidth allocation is always the lower end of the specified range, they optimize by specifying (b_{min}, b_h) where b_{min} is calculated as in (1). Fig. 6 illustrates user benefit with this user optimization ($b = 0$).

As expected, user benefit is lower when the entire range is specified ($b = 1$) in the lightly and heavily loaded regions. We see that the benefit is greater in the moderately loaded region when the user tries to optimize. The user may take the blocking factor into account by reducing the optimal value of b_{min} in (1) as long as it is above b_l. We define the backoff factor b such that

$$b_{min,new} = b_{min} - b * (b_{min} - b_l)$$

Fig. 6 shows the benefit curves for different values of b. The envelope of these curves is the optimal user benefit curve and would be obtained if the user could calculate the optimal range knowing the load in the network taking the blocking factor into account. Thus, in a static scenario, a region exists where an optimizing user does not specify the entire range to the network. We note that this region is small and the gains may not be significant enough to offset the complexity and overheads introduced by by the optimizing algorithm. The existence of this region in the dynamic scenario is investigated next.

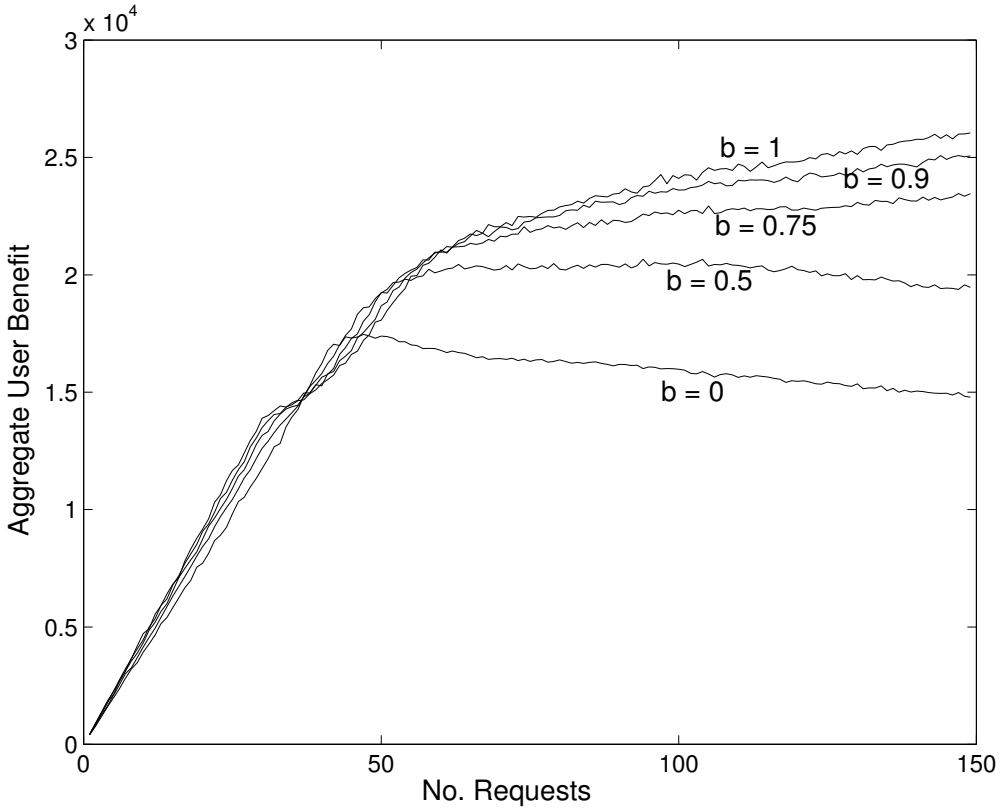


Figure 6: User Benefit with User Optimization: Static case

Though the study of the static scenario provided us with insight into user behavior and system performance, the dynamic scenario models an actual network more closely. In the dynamic case, the network has make a decision on connection admittance when the request is received. Once a connection is admitted, it has to be allocated at least the minimum specified bandwidth for its duration. We expect a degradation in performance as compared to the static case since the network cannot rank connections before deciding which ones to admit. We assume that in the fixed cost case, the available bandwidth b_a is known. If b_a is less than b_h, the connection demands the larger of b_a and b_l. In the proposed policy, the entire range (b_l, b_h) is specified.

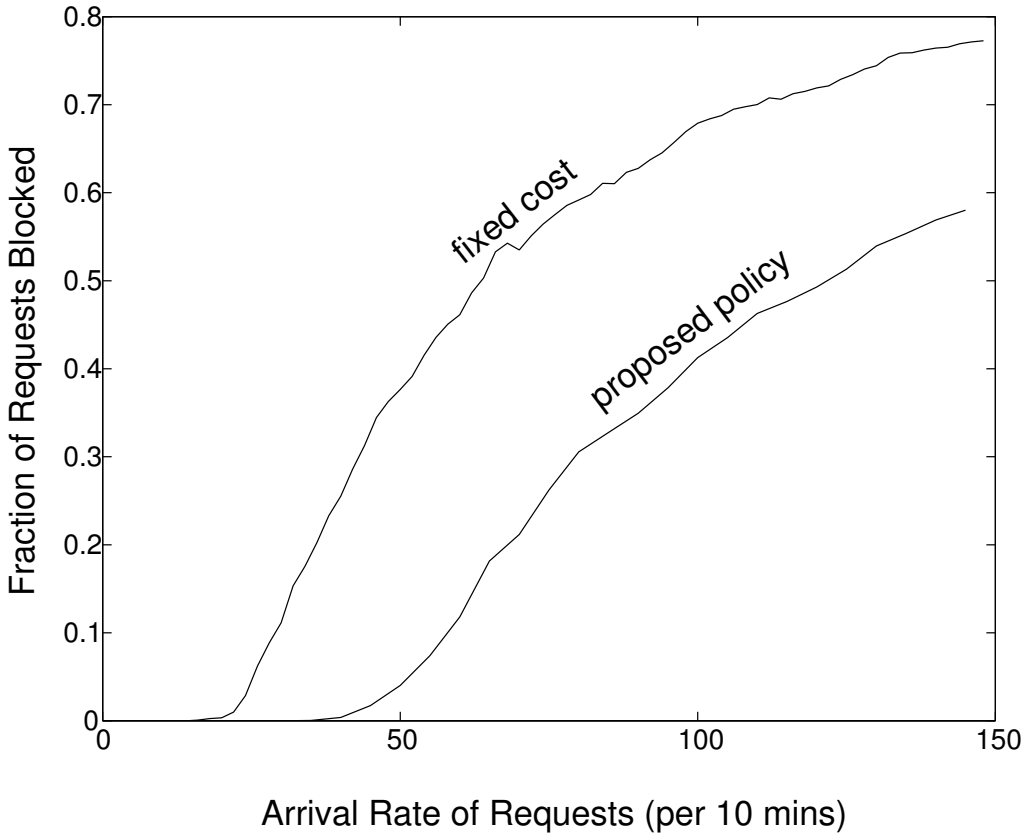


Figure 7: Percentage of Blocked Requests (Dynamic Case)

Fig. 7 shows that the fixed cost policy starts blocking at a much lower number of requests than the proposed policy. In both cases, requests are admitted as they are received until there is no more bandwidth left. Beyond this, the fixed cost policy blocks requests, while the proposed policy scales down the admitted connections to free up bandwidth for

the new request. The network revenue increases in the blocking region for the proposed policy because connections are still admitted if sufficient bandwidth is released by scaling existing connections. In the fixed cost case, revenue flattens out as the number of admitted connections increases only slightly. This is illustrated in Fig. 8.

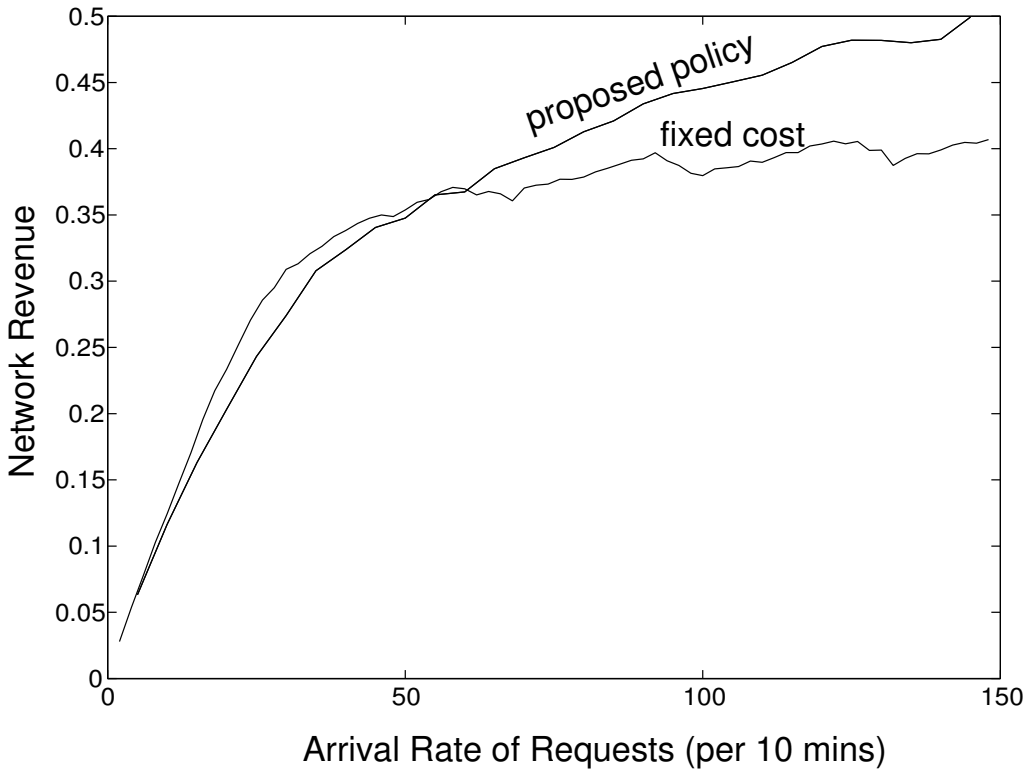


Figure 8: Network Revenue: Dynamic Case

A similar trend is observed in the user benefit curves shown in Fig. 9. In the fixed cost case, few additional connections are accepted after blocking starts. The utility of admitted connections decreases while the cost stays the same, so the utility flattens out. In the proposed policy, the loss of utility when a connection is scaled is offset by the decrease in cost. Furthermore, new connections are still accepted in the blocking region.

Finally, we examine user preferences for the proposed policy in the dynamic case. Fig. 10 shows user benefit curves for three user preferences.

Curve B corresponds to a policy where the user specifies the entire range (b_l, b_h). Curve A corresponds to the specification of (b_{min}, b_h), where b_{min} is calculated using (1). The user ignores the probability of blockage in the specification of minimum bandwidth, and this curve

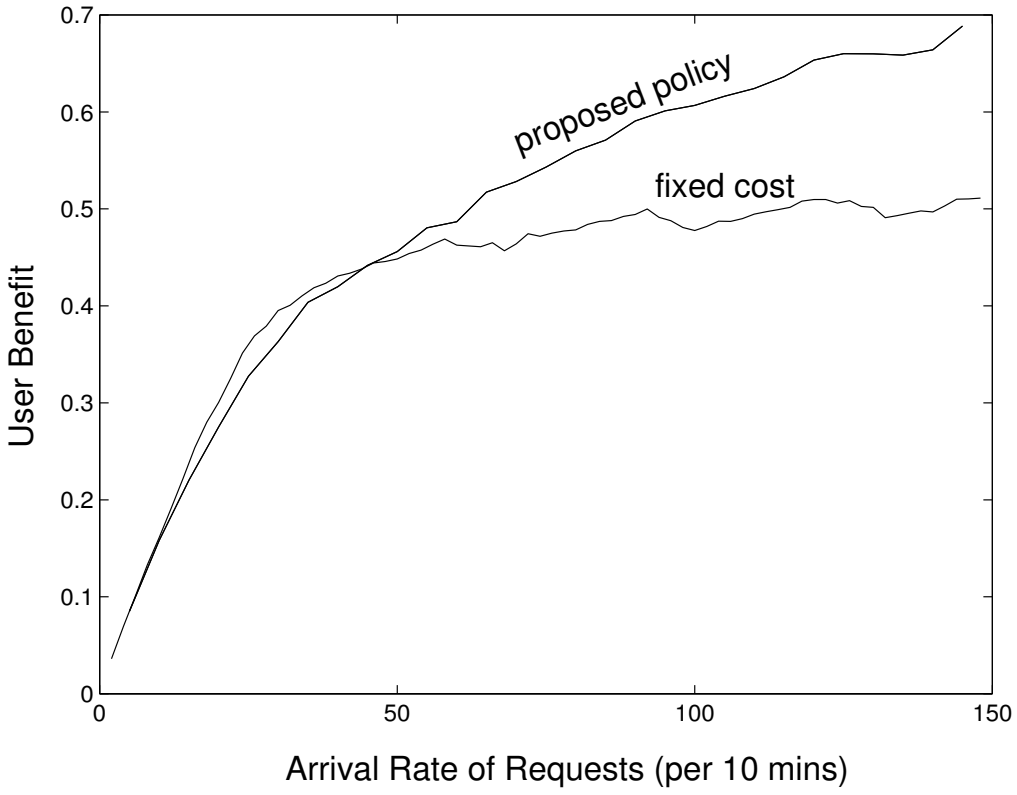


Figure 9: User Benefit: Dynamic Case

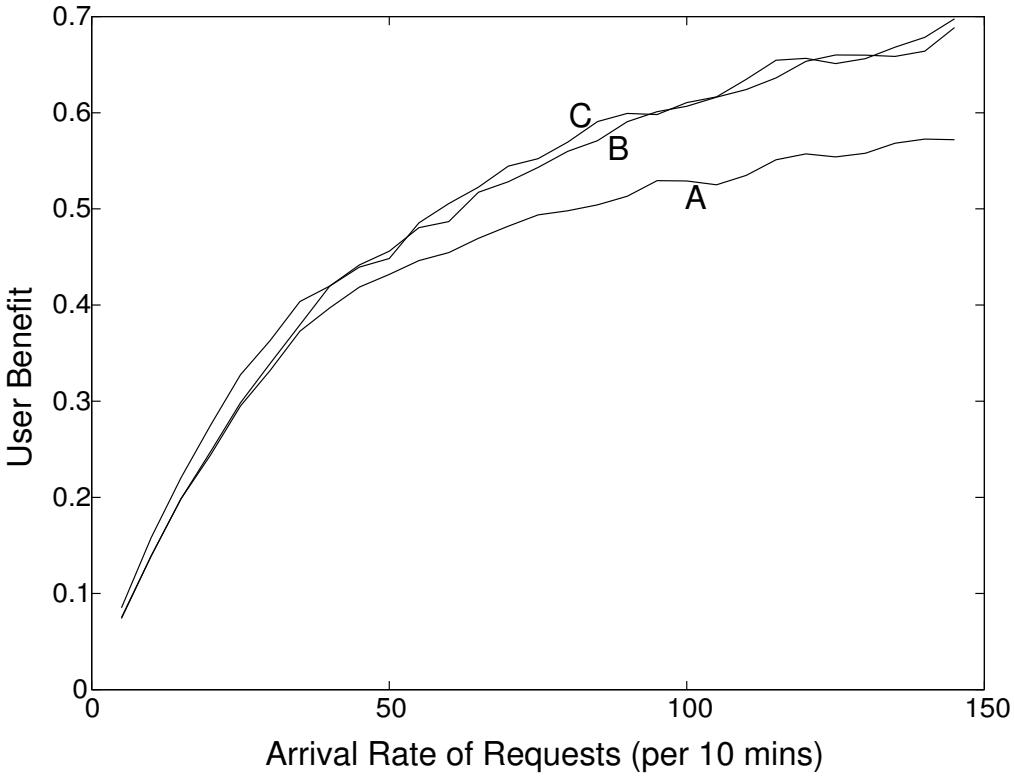


Figure 10: User Benefit With User Optimization: Dynamic Case

is clearly suboptimal. The third policy (Curve C) assumes that the user has information about the available bandwidth b_a and specifies (b_m, b_h) where $b_m = max((min(b_{min}, b_a)), b_l)$. That is, the optimal bandwidth is specified only if it is less than the available bandwidth. Otherwise, the available bandwidth is specified, lower bounded by b_l. Curve C is optimistic in that the user gets information on maximum bandwidth available for admission. We observe that Curve B is close to Curve C, suggesting that users should provide near full scalability. This implies that even if the user tailors the specification based on knowledge about available bandwidth, the benefit does not improve significantly in the blocking region.[5] Thus, the user achieves close to optimal benefit by specifying the entire range of scalability to the network.

5 Conclusions

Most video applications are scalable. The network can apply this scalability to improve connectivity and revenue by means of a dynamic resource reservation protocol. However, users suffer from performance degradation when the connection is scaled down. Thus, users will not specify scalability to the network unless there is an incentive to do so. In this paper we proposed a pricing policy which provides users with monetary incentives to specify scalability. We also proposed a corresponding dynamic admission control scheme which the network uses to maximize its revenue. Our simulation results demonstrate that the proposed pricing policy encourages users to specify application scalability to the network during connection establishment by increasing their benefit. We also show that this policy coupled with the admission control scheme improves user utility, network revenue, and network connectivity over a fixed cost scheme which does not consider application scalability.

References

[1] Delgrossi, L., C. Halstrick, D. Hehmann, R. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media Scaling for Audio Visual Communication with the Heidelberg Transport System," *Proc. ACM Multimedia 93*, Anaheim, CA, August, 1993, pp. 99-104.

[2] Cocchi, R., S. Shenker, D. Estrin, L. Zhang, "Pricing in Computer Networks: Motivation, Formulation, and Example," *IEEE/ACM Trans. on Networking*, Vol. 1, No. 6, pp. 614-627.

[5] Note that B is optimal in the lightly loaded region.

[3] Ferrari, D. and D.C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, pp. 368-379, April 1990.

[4] Krishnamurthy, A. and T.D.C. Little, "Connection-Oriented Service Renegotiation for Scalable Video Delivery," *Proc. International Conference on Multimedia Computing and Systems* Boston, MA, May 14-17, 1994, pp. 502-507.

[5] Krishnamurthy, A., "A Dynamic Resource Reservation Protocol," Ph.D. Dissertation, in progress, Boston University, 1995.

[6] MacKie-Mason, J.K., and H. Varian, "Pricing the Internet, *Public Access to the Internet*, Prentice-Hall, Englewood Cliffs, New Jersey.

[7] Parris, C. and D. Ferrari, "A Resource Based Pricing Policy for Real-Time Channels in a Packet-Switching Network," *Tech. Report, ICSI*, TR-92-018, March, 1992.

[8] Parris, C., S. Keshav, and D. Ferrari, "A Framework for the Study of Pricing in Integrated Networks," *International Computer Science Institute TR-92-016*, ICSI, Berkeley, California, 1992.

[9] Shenker, S., "Service Models and Pricing Policies for an Integrated Services Network," Technical Report, Palo Alto Research Center, Xerox Corporation.