Evaluation of resource sharing benefits

Amit Gupta, Wingwai Howe,
Mark Moran, Quyen Nguyen
{amit,whowe,moran,nguyenq}@tenet.Berkeley.EDU
Tenet Group
University of California at Berkeley, &
International Computer Science Institute

TR-94-051

October 1994

Abstract

Current approaches to supporting real-time communication allocate network resources either to individual connections, or to aggregates of connections, based on type of traffic, protocol, or performance requirements. The first approach provides well-defined performance guarantees that are independent of other network traffic. The second approach may achieve higher utilization of network resources, but the expected performance is less well-defined since it is dependent on the behavior of unrelated (possibly unknown) connections. We previously presented *resource sharing* , a new approach that exploits known relationships between *related* connections to allow network resources to be shared without sacrificing well-defined guarantees. Resource sharing is very important for large conferences with a bounded number of *concurrent* speakers, resource requirements do not increase with the number of potential speakers. In this paper, we evaluate resource sharing benefits by analysis and by simulation. Results show that resource sharing leads to a large gain in the connection acceptance rate, and a significant reduction in the computational overhead associated with admission control.

1 Introduction

Many classes of applications, including distributed multimedia group communication [12] and traditional distributed processing, require or benefit from a network communication service that provides well-defined performance guarantees. A number of protocols and schemes have been proposed to provide real-time communication services [3, 5]. These schemes are usually connection-oriented, in that they allocate network resources (e.g., bandwidth, buffers, etc.) along the path data packets will travel.

Traditional real-time network systems (e.g., [4]) may over-allocate resources for two reasons: (1) they allocate resources based on a worst-case prediction of the actual traffic; and (2) they treat traffic on different connections independently when determining their resource requirements. One technique to improve utilization (and hence the connection acceptance rate) is to measure the actual traffic characterization of individual connections and to modify the connection traffic characterization dynamically [14]. Another technique uses performance measurements over aggregations of connections to predict future performance [3]. The first approach still over-estimates aggregate resource requirements, since it fails to capture the important relationships between connections (e.g. in a conference, usually only one speaker is active at a time). The second approach will indirectly capture these relationships, but it fails to provide protection against unrelated traffic, and depends on the assumption that current behavior adequately predicts future behavior for all connections in the aggregate. This assumption may not be valid when a single connection can have a significant effect on the performance of other connections, e.g., over low-capacity links. In addition, when measurements are not available (e.g. when a connection is first established), this approach must fall back to use independent traffic characterizations, as in the first approach.

In previous work [8], we proposed *resource sharing* as a "middle ground", by which *related* connections can *share* resource allocations in a controlled manner, so that network utilization is improved and performance guarantees of established connections are achieved. Resource sharing differs from techniques that rely on *statistical multiplexing* of (unrelated) network traffic, in that the network client specifies how traffic from related connections is multiplexed. As long as the aggregate traffic of these related connections does not exceed this specification, the network service guarantees that well-defined performance bounds will be met for individual channels. As the client specifies the related connections and their aggregate traffic, all sharing is completely client-controlled. Most importantly, performance guarantees are *not* dependent on the behavior of unrelated network traffic. Resource sharing is an important technique to provide well-defined performance guarantees for most large-scale, multi-party communication paradigms.

In this paper, we evaluate the performance of resource sharing, by analysis and by simulation. The results show that resource sharing leads to a large gain in the connection acceptance rate, and a significant reduction in the computational overhead associated with admission control. We have organized this paper in the following manner. Section 2 provides a brief description of the resource sharing techniques that we evaluate in this paper; the readers should read [8] for a complete description of the resource sharing technique. We provide analytical bounds on resource sharing benefits in Section 3, and describe the results of our simulation experiments in Section 4. Section 5 compares the simulation results with the analytical bounds. We discuss the related work in Section 6 and we conclude this paper with suggestions for future work in Section 7.

2 Background

In this section, we provide a very brief introduction to resource sharing in the context of the Tenet protocols; interested readers can find more information on resource sharing in [8], and a detailed overview of the Tenet protocols in [4, 10].

In the second generation of the Tenet protocols [7], the basic abstraction is the multicast real-time channel—a simplex, one-to-many connection at the network-layer. A conference with M sources and N destinations consists of M multicast channels to the same set of N destinations, where the destination set is referred to as the *Target Set* for that conference. A channel is characterized by traffic and performance specifications given by the network service client(s). The traffic specification bounds the data rate at which the channel may inject traffic into the network. The performance specification describes the performance bounds that the network guarantees to meet for compliant clients. A typical set of performance parameters includes bounds on the end-to-end packet delay (delay bound), on the variation in the end-to-end packet delay (delay jitter bound), on the probability that the delay bound will be met (timeliness bound) and on the packet loss rate due to buffer overflow (loss bound). The Tenet approach is connection-oriented and reservation-based: before data can be transmitted on a real-time channel, the channel must be established (i.e., resources must be allocated for the channel at each server along its route), so that the guarantees are supported. The required resources may not be available along one or more branches of the multicast tree, resulting in a partial failure of channel establishment.

We will now describe source-initiated channel establishment; channel establishment can also be receiver-initiated. Channel establishment is a fully-distributed two-pass process. In the first (forward) pass, a message from the source visits each node on the multicast route of the channel. At each intermediate node, the network makes tentative reservations and forwards the establishment message to the next node. In the second (reverse) pass, reply messages flow from all destinations to the source along the same route as was used for the forward pass, but in the reverse direction. The resource reservations are committed on the reverse pass.

Our resource sharing mechanisms provide the following functionality:

- *Client-service interface:* which the network client uses to inform the network of sharing relationships between channels. This interface defines the contractual agreement between the client and the network.

- *Admission control tests:* by which the network admission control tests use the information supplied by the client to perform local admission control tests on groups of channels, rather than on each channel individually. After the number of channels for a group reaches the sharing threshold at a particular server, additional member channels can be established without performing *any* additional admission control tests.

- *Protection:* by which the network ensures that network resources consumed by a group of channels do not exceed the resource allocation of that group.

See [8] for more details on the mechanisms we have designed to support resource sharing.

3 Analysis

For analytical evaluation of resource sharing benefits, we introduce a new metric: *allocation gain.* For a set of connections over a set of links, we define allocation gain as the ratio of the resource allocation required without resource sharing, to the resource allocation required when resource sharing is used. For example, an allocation gain of 4 means that under resource sharing, $\frac{1}{4}$ as many resources are required as without resource sharing. We chose allocation gain as the metric because resource sharing benefits accrue on a per-link basis; therefore, it is difficult to extrapolate the analysis to gains in overall call acceptance. In the next section, we will use a different metric called *acceptance gain* for evaluating the resource sharing benefits in the simulation experiments.

A simple example can show that under some conditions (very dense networks, adversarial routing etc.), resource sharing will not provide any useful gains in resource allocation. Consider a complete network, i.e. direct links connecting all pair of nodes and a routing algorithm that ensures that data packets go directly from the source to the destinations, with no intermediate nodes, under the assumption that no two senders reside on the same network node. In this case, no link carries data belonging to more than one channel; consequently, we cannot obtain any savings by introducing resource sharing.

However, we do expect that in sparser computer networks, we will obtain significant savings with resource sharing. We present two analyses to demonstrate these savings: the first analysis is for sparse networks while in the second analysis, we assume that for each conference, the network routing function creates a single undirected tree and returns routes along that tree for all connections that constitute that conference. We present useful results for resource sharing gains under these sets of assumptions.

3.1 Sparse network

Wide-Area-Networks (WANs) tend to be rather sparse; for example, the NSFNET backbone WAN has 32 nodes and only 35 links. In this analysis, we obtain lower bounds on resource sharing gains for networks with small cut-sets of links. It should be noted, that sparse networks usually have small cut-sets of links. For the purpose of this analysis, we assume that the conference participants are homogeneous and are uniformly and independently distributed among the network nodes. To keep the analysis general, we make no assumptions whatsoever about the routing algorithms.

Lemma 1 *Consider a network in which channels always traverse at least one link of a set of links L, with $|L| = l$. If a conference has m channels, then a lower bound, g, on the resource sharing allocation gain is given by*
$g = \frac{m}{2tl}$,
where t is the sharing threshold specified by the client.

The lemma follows from a simple counting argument: Without resource sharing, at least one reservation must be made over the links of the set L for each of the m channels, for a minimum of m reservations over the set L. With resource sharing, at most t reservations must be made (in each direction) for each link of set L. Since a link has only 2 directions, a maximum of $2tl$ reservations are required.

For a conference with 40 channels, $t = 2$, and $l = 5$, the allocation gain g exceeds 2.

Lemma 2 *Consider a network with a cut-set L of links that divide the network into a set of subnetworks, such that no subnetwork contains more than a fraction f of the nodes of the original network. If a channel has n destinations uniformly and independently distributed among the network nodes, then the probability that the channel traverses at least one of the links of the cut-set L is given by*
$prob \geq 1 - f^n$.

A channel must traverse a link of the cut-set L *unless* all destinations reside in the same subnetwork as the source. Assuming a uniform, independent distribution of destinations, the probability that a single destination resides in the same subnetwork as the source is at most f. Thus, the probability that all n destinations reside in the same subnetwork as the source is at most f^n.

As the number of destinations increases, the probability *prob* rapidly approaches 1; for example, with $f = 0.75$, $n = 16$, probability *prob* is greater than 0.99.

Theorem 1 *Consider a network with a small link-set L with $|L| = l$, where r is the probability that a channel traverses the link-set L. If a conference has m channels and the sharing threshold t, the following relation holds.*
$\log(\frac{1}{1-p}) = \frac{mr}{2}(1 - \frac{2tlg}{mr})^2$,
where p is the probability that the allocation gain is at least g.

We use Chernoff bounds on sum of independent Bernoulli trials (first described in [2]) to obtain lower bounds on resource allocation gains that result from resource sharing. This Chernoff bound states that for independent Bernoulli trials with $P[X_i = 1] = p_i, p_i \in (0, 1)$, and random variable X, where $X = \sum_{i=1}^{n} X_i$, and $\mu = \sum_{i=1}^{n} p_i > 0$,

$$P\left[X < (1-\delta)\mu\right] < exp(-\mu\delta^2/2),$$

where μ is the expected value of X. We re-arrange the expression to

$$P\left[X \geq x : x = (1-\delta)\mu\right] \geq 1 - exp(-\mu\delta^2/2).$$

Let X be a random variable representing the number of channels (of the m channels that comprise the conference) that traverse the link-set L. Here, $\mu = E[X] = mr$. From Lemma 1, the gain factor will exceed g, iff $X > x$, where $x = 2tlg$. Setting $x = (1-\delta)\mu$ yields, $\delta = (1 - \frac{2tlg}{\mu})$. Substituting μ and δ into the previous expression,

$$P\left[X \geq x\right] \geq 1 - exp\left[-mr(1 - \frac{2tlg}{mr})^2/2\right].$$

Let p denote the lower bound on the probability $P\left[X \geq 2tlg\right]$; thus,

$$p = 1 - exp\left[\frac{-mr}{2}(1 - \frac{2tlg}{mr})^2\right].$$

Rearranging the above expression, we obtain:

$$\log(\frac{1}{1-p}) = \frac{mr}{2}(1 - \frac{2tlg}{mr})^2.$$

For example, the probability p is greater than 0.95 for $g = 2$, with $m = 50$, $r = 0.99$, $l = 4$ and $t = 2$.

3.2 Tree-based routing

In this section, the analysis shows the strong relationship between routing and resource sharing gains. We adopt a simple routing strategy that attempts to increase the sharing gains by selecting the same *undirected* routing tree for all channels that belong to a particular conference. One way of looking at this tree selection process is that the routing system selects a spanning tree T for the network. Then for every channel, the appropriate *directed* subtree t, of T, that connects all destinations to the source is used. This behavior is exhibited by many current routing algorithms, including Core-Based-Trees [1]. In this analysis, we relax the constraints that we previously imposed on the network topology and on the distribution of destinations among the network nodes; thus, this analysis is applicable to arbitrary network and connection topologies.

To keep the analysis simple, we assume that, for a given conference, the routing algorithm selects the same forwarding tree, regardless of whether resource sharing is used. Admittedly, a routing algorithm for conferences that do not use resource sharing probably would not be tree-based, since such an algorithm would tend to cause congestion on the shared links. However, since our analysis merely calculates the resources *required* at each server rather than performing admission control (essentially assuming infinite resources are available) the assumption of tree-based routing does not adversely affect the analysis.

Consider the subset L of links that connect the destinations for the conference. It is easy to see that every channel will traverse all links in L exactly once in some direction. We can then obtain the following corollary of Lemma 1.

Corollary 1 *Consider any link l that belongs to the set of links L which connects the destinations of the conference as described above. If a conference has m channels and resource sharing threshold t, a lower bound g, on the allocation gain for each link i is given by*
$g = \frac{m}{2t}$.

For a conference with 40 channels with a sharing threshold of 2, the allocation gain exceeds 10.

Consider a link i of the spanning tree T that is used for routing channels for a given conference. The link i divides the tree T into two subtrees. Let the ratio of number of nodes in the two subtree be r_i, $r_i > 1$. Under the assumption that the destinations are uniformly and independently distributed among the nodes in the network, we obtain the following corollary of Lemma 2.

Corollary 2 *If a channel has n destinations, the probability f that a randomly selected channel will traverse the link l is given by*

$$\begin{aligned} f &= 1 - \left(\left(\frac{r_i}{1+r_i}\right)^{n+1} + \left(\frac{1}{1+r_i}\right)^{n+1}\right) \\ &= 1 - \frac{r_i^{n+1}+1}{(1+r_i)^{n+1}} \\ &> 1 - \left(\frac{r_i}{1+r_i}\right)^{n} = 1 - \left(1 - \frac{1}{1+r_i}\right)^{n} \\ &> 1 - e^{\frac{n}{1+r_i}} \end{aligned}$$

For example, $f > 0.999$ for $n > 10 * (1 + r_i)$.

The following corollary follows from Theorem 1 and Corollary 1.

Corollary 3 *If a conference has m channels and the sharing threshold t, the following relation holds for all links that belong to the routing tree for that conference.*

$\log(\frac{1}{1-p}) = \frac{mf}{2}(1 - \frac{2tg}{mf})^2$,
where p is the probability that the allocation gain is at least g, and f is as given by Corollary 2 above.

3.3 Examples

In this subsection, we illustrate the above analytically-derived bounds with some graphs; we have derived these graphs from Corollary 3.

For these graphs, we set sharing threshold $t = 2$, probability p of Corollary 3 as 0.99 and probability $f = 0.999$ unless otherwise stated.

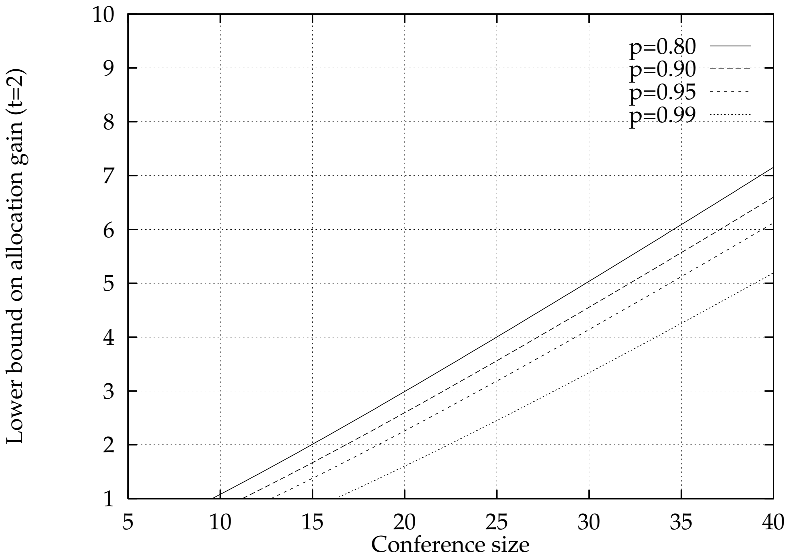

Figure 1: Analysis – Allocation gain for constant sharing threshold

In Figure 1, we fix sharing threshold at 2, and obtain allocation gains as we vary the conference size; we obtain curves for p varying from 0.8 to 0.99, where p is the probability that the allocation gain will exceed its bound.

As expected, at fixed probability p, the lower bound g on allocation gain increases almost linearly with the conference size; also, as the probability p increases (thereby we are getting closer to guaranteeing that the allocation gain will exceed the lower bound), the value of the lower bound g decreases for the same value of conference size.

In Figure 2, we fix the probability p of exceeding the sharing threshold at 0.99, and obtain allocation gains as we vary the conference size; we obtain curves for sharing threshold t varying from one to five.

As expected, for a given sharing threshold, the allocation gain increases almost linearly with the conference size; also, as the sharing threshold increases, the lower bound on allocation gain goes down.

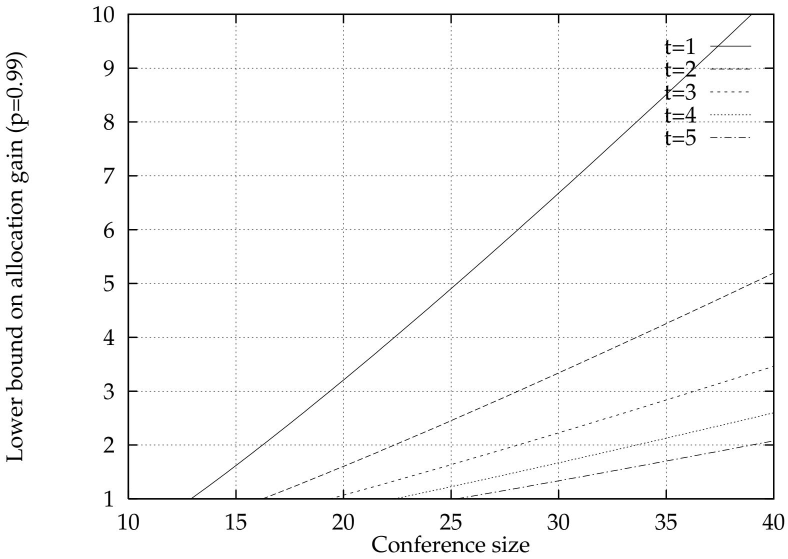

Figure 2: Analysis – Allocation gain for varying sharing threshold

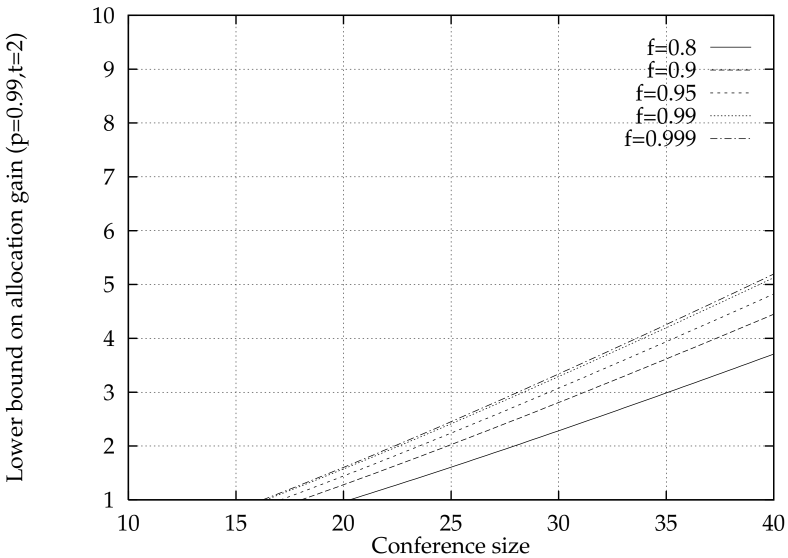

Figure 3: Analysis – Allocation gain for constant probability of exceeding the gain bound

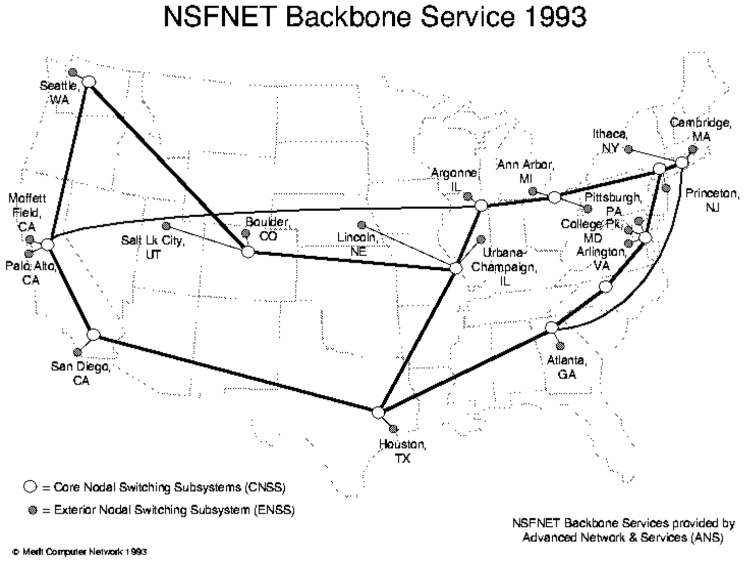

Figure 4: The NSFNET Network

In Figure 3, we fix sharing threshold at 2, and the probability of exceeding the allocation bound at 0.99 (i.e. 99% of the time, the actual allocation gain will exceed g). We obtain allocation gains as we vary the conference size; we obtain curves for f varying from 0.8 to 0.999, where f is the probability given by Corollary 2.

As expected, for fixed probability f, the lower bound on allocation gain increases almost linearly with the conference size; also, as the probability f increases (implying that the link is more likely to be traversed by most connections), the lower bound on allocation gain increases.

4 Simulations

In the previous section, we provided analytical bounds for allocation gain due to resource sharing. In this section, we present the results of our simulations with resource sharing. We ran these simulations on *Galileo* [9], an object-oriented real-time network simulator. Our goal was to make the experiments realistic so that the results obtained can be confidently transposed to our resource sharing implementation in the Tenet Protocol Suite 2 [7]. For this, the network topologies that we used in the simulations are based on two real wide-area networks – the NSFNET backbone network, and XUNET [6], a high-speed ATM network that spans across North America from Bell Labs in New Jersey to Berkeley, California.

Simulation workload and evaluation metrics

In all the experiments, the sources and destinations for the conferences were uniformly and independently distributed among the network nodes. We ran the same simulations with and without resource sharing; in this section, we denote by RS the experiments with resource sharing and by non-RS the experiments without resource sharing. For simplicity, we only considered homogeneous workload, where all channels provide identical traffic specifications, and all destinations specify identical performance requirements. We chose traffic parameters that might represent a medium-quality video stream (peak rate of 1 Mb/sec), and delay bounds consistent with interactive communication:

- Deterministic Delay Bound $D = 400$ ms ;
  Jitter Bound $J = 16$ ms;
- Minimum Packet Interarrival Time = 8 ms;
- Maximum Packet Size = 8 Kb;
- The sharing threshold equals 1, unless otherwise stated.

We performed many sets of experiments, each time varying one of three workload parameters: number of concurrent conferences, number of participants per conference, and the sharing threshold. For each of these workloads, several trials have been run, and the results obtained are averaged. In the first set, we varied the number of conferences, while keeping the number of participants in a single conference (*conference size*) fixed at 10. In the second set of experiments, we varied the conference size, and fixed the number of conferences at 50. We chose this workload to overload the system slightly, since the benefits of resource sharing are greatest under high network utilization.

The main metric for performance evaluation is:

$$\text{Destination Acceptance Rate} = \frac{\text{Number of Destinations successfully established}}{\text{Number of Destinations attempted}}$$

Here, *destinations* refer to the recipients in a multicast channel. For the remainder of this paper, *acceptance rate* will refer to the *destination acceptance rate*, unless otherwise stated. We also define another metric, called *acceptance gain* to be the ratio of acceptance rates with and without resource sharing respectively.

In addition, we are also interested in the timeliness and computational cost of channel establishment, for which we use a different metric: the computational overhead associated with admission control. In the simulations, we use the admission control tests for the EDD scheduling discipline [5]. The time complexity of these tests at a given node is $O(n)$, where n is the number of accepted reservations at that node. Establishment overhead is computed as the sum of the values of n at each node along the route of the channel.

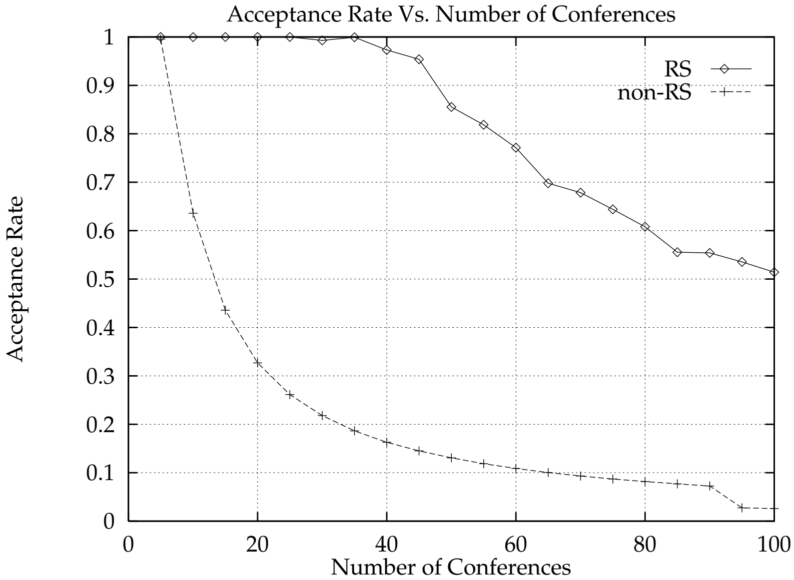

Figure 5: NSFNET – Acceptance Rate Vs. Number of Conferences

4.1 NSFNET

In this set of experiments, we ran our simulations on the backbone of the NSFNET network; we only included the core nodes (CNSS) of the network shown in Figure 4.

Results

- Acceptance rate

  (i) Increasing the Number of Conferences
  With resource sharing, the acceptance rate is consistently higher than without resource sharing across a wide range of the number of conferences (from 5 to 100). From Figure 5, when the network is heavily-loaded (40 or more conferences), the destination acceptance rate with RS is at least 6 times higher than that for non-RS.

  (ii) Increasing the Conference Size
  With RS, the acceptance rate remains close to 1 even when the size of conferences approaches the size of the network while the acceptance rate for non-RS degrades substantially as the conference size increases. This behavior is expected because with resource sharing, the amount network resources allocated to a conference is bounded by the sharing threshold, regardless of the size of the conference. This experiment illustrates the scalability and the importance of resource sharing for large multi-party applications.

  (iii) Varying the Threshold
  In Figure 7, we vary the sharing threshold in the sharing specification. In the

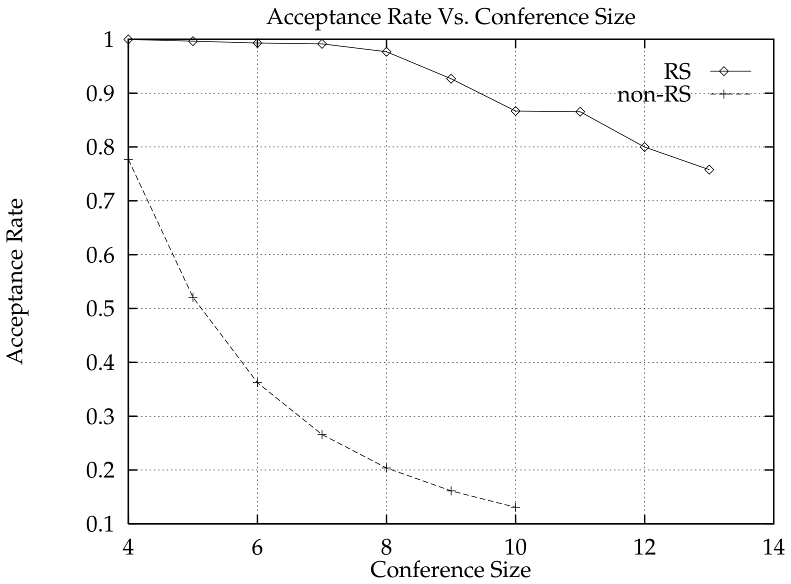

Figure 6: NSFNET – Acceptance Rate Vs. Conference Size

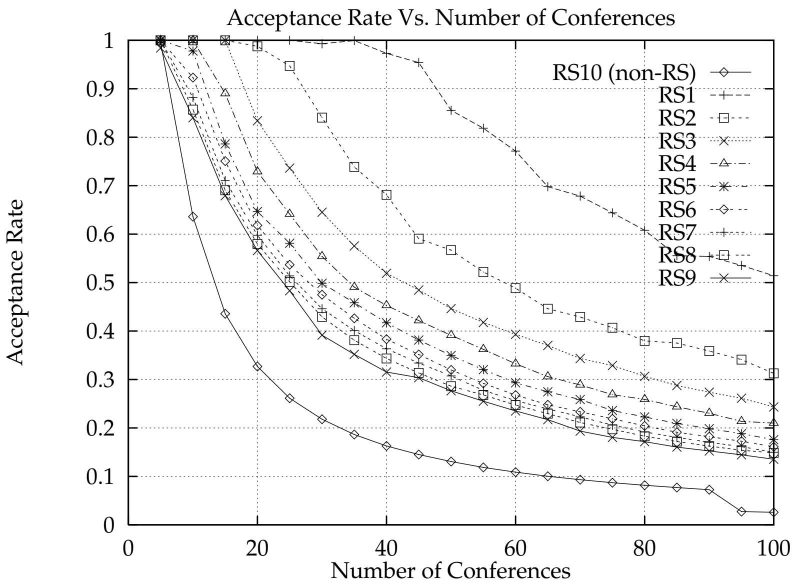

Figure 7: NSFNET – Acceptance Rate with Different Threshold Value

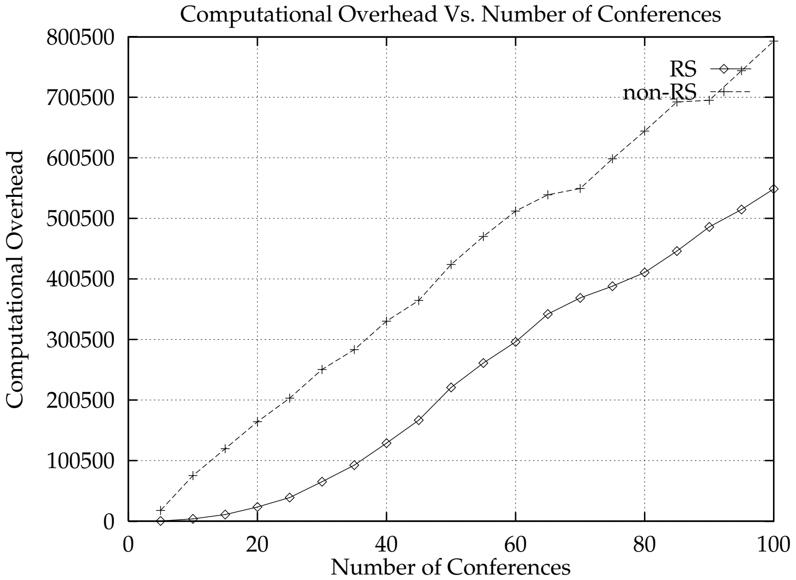

Figure 8: NSFNET – Computational Overhead Vs. Number of Conferences

graphs, we follow the notation that RSi refers to RS with sharing threshold of i; RS1 denotes the case of sharing threshold = 1; RS2 denotes the case of sharing threshold = 2 etc. Sharing threshold equal to the conference size (RS10 in Figure 7) amounts to turning resource sharing off. Figure 7 shows that with lower sharing threshold, the resource sharing gain is significantly higher, and that RS consistently out-performs non-RS.

- Computational Overhead

  According to Figure 8 and Figure 9, the admission control computation overhead for RS is always smaller than the overhead for non-RS. These results indicate that resource sharing does not add to the establishment overhead; indeed, resource sharing tends to *decrease* the establishment overhead. As described in Section 2, when the number of admitted channels in a sharing group reaches the sharing threshold at a server, the admission control mechanisms at *that server* will accept subsequent channels of the same group without performing any additional admission control tests.

4.2 Tree topology

We ran several simulation experiments with a tree-based topology (in which we added a few new nodes to the XUNET network topology of Figure 10).

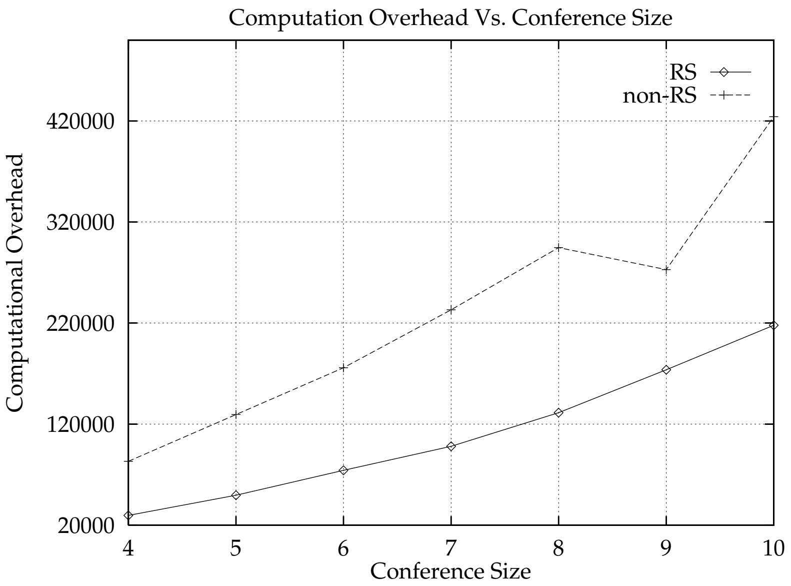

Figure 9: NSFNET – Computational Overhead Vs. Conference Size

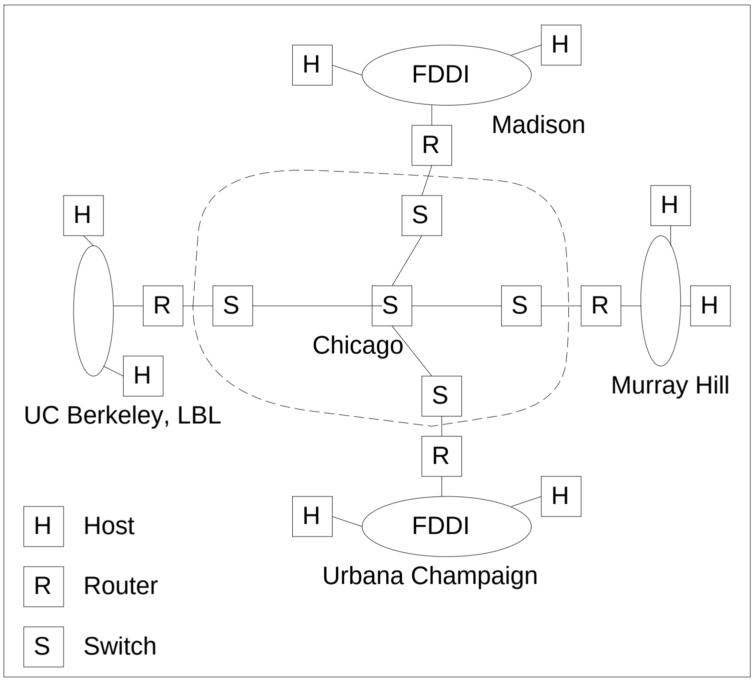

Figure 10: The XUNET Network

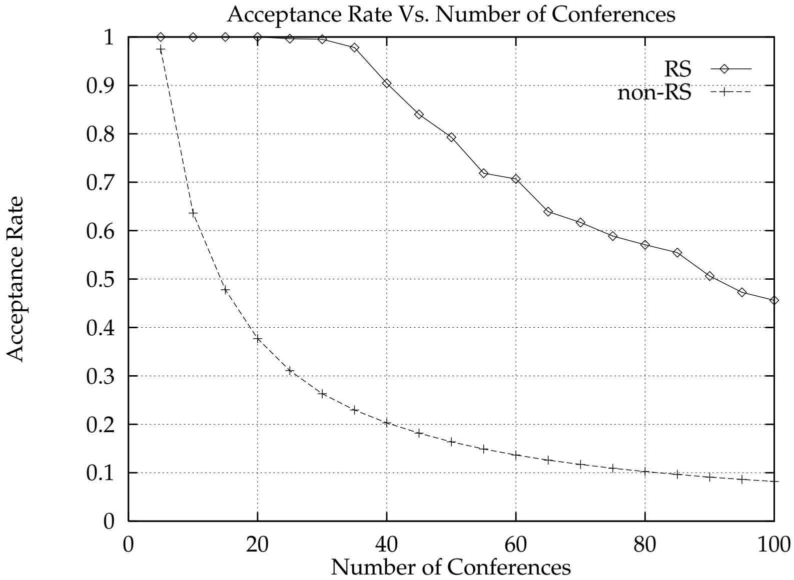

Figure 11: XUNET – Acceptance Rate Vs. Number of Conferences

Results

- Acceptance rate
  As in the previous section, we compare the acceptance rates obtained with and without resource sharing for a number of different experiments. The graphs (Figure 11 - 13) show how the system performance changes as we vary three parameters:

  (i) the number of conferences (Figure 11) ;

  (ii) the conference size (Figure 12) and

  (iii) the sharing threshold (Figure 13)

  The results are similar to the results that we obtained with the NSFNet topology. Resource sharing is shown to yield a higher acceptance rate.

- Computational Overhead
  Figure 14 and 15 show the computational overhead of RS and non-RS. The results are again very similar to those from the NSFNet in that RS in general reduces the computational overhead during channel establishment.

5 Comparison: simulation vs. analysis

In this section, we compare the simulation results for XUNET (tree topology) with the analytical bounds described in Corollary 1 of Section 3. Please observe that the analytically-derived *allocation gain* relates to the resource allocations at a link

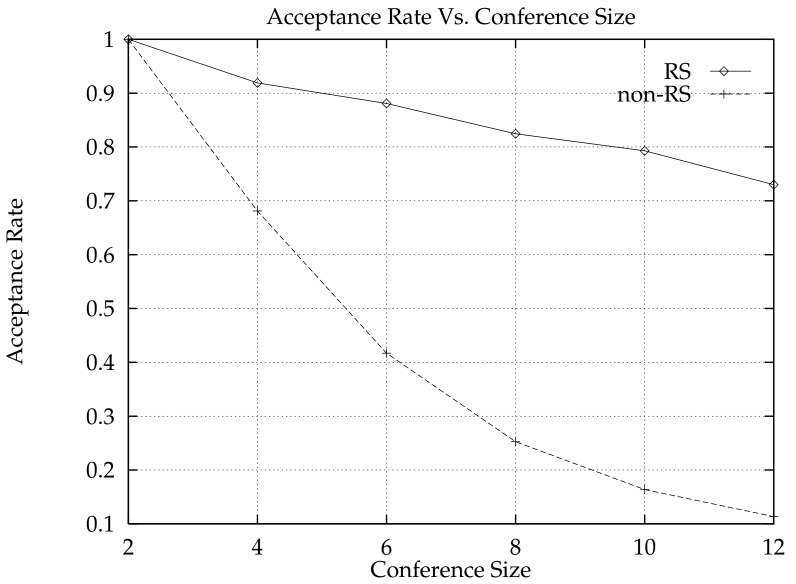

Figure 12: XUNET – Acceptance Rate Vs. Conference Size

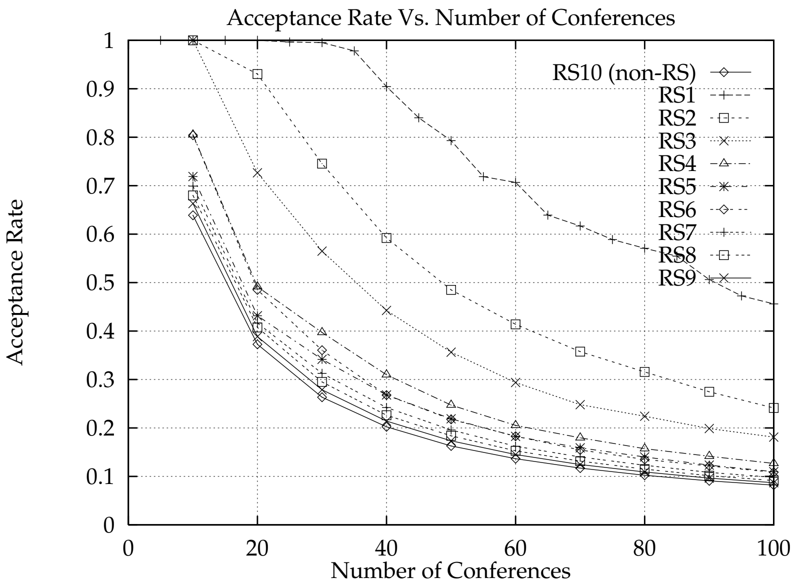

Figure 13: XUNET – Acceptance Rate with Different Threshold Value

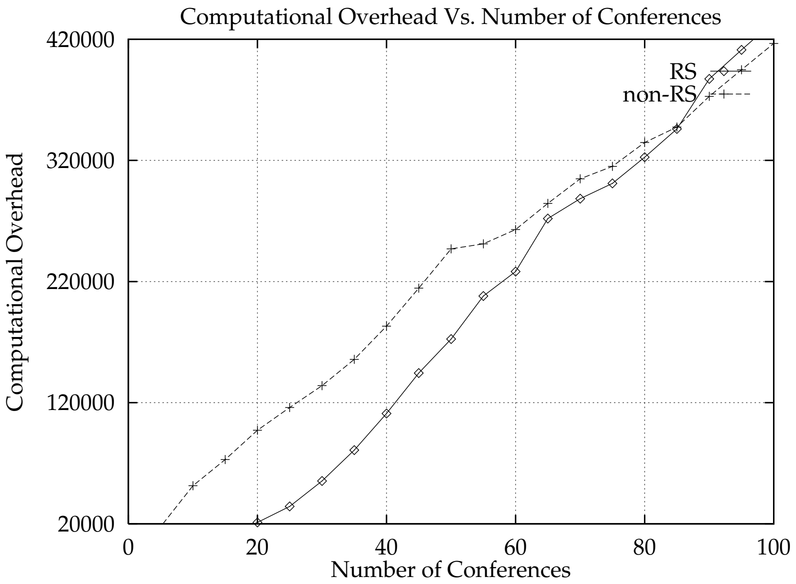

Figure 14: XUNET – Computational Overhead Vs. Number of Conferences

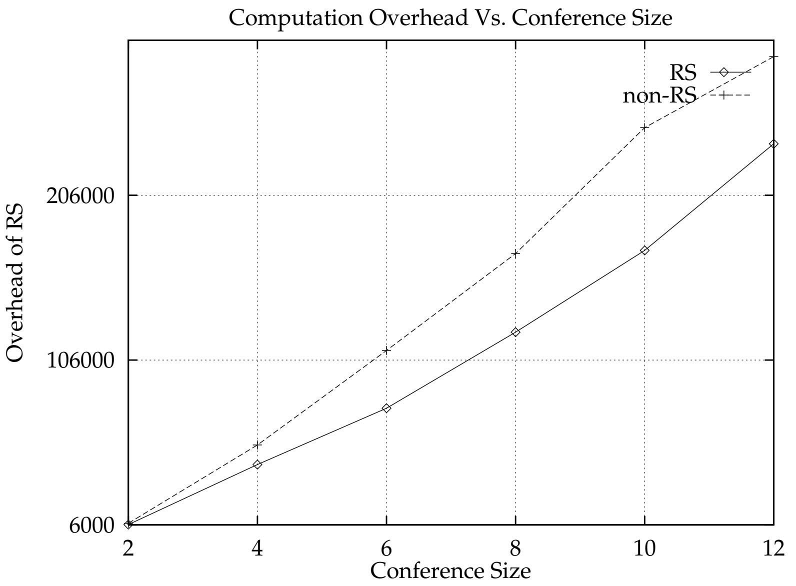

Figure 15: XUNET – Computational Overhead Vs. Conference Size

in the network, while the simulation-based *acceptance gain* provides an assessment of resource sharing benefits for the overall network. However, one can intuitively see that acceptance and allocation gains are well-correlated: if resources are saved, the network will be more capable of admitting future channels, thus increasing the acceptance rate and consequently, acceptance gain.

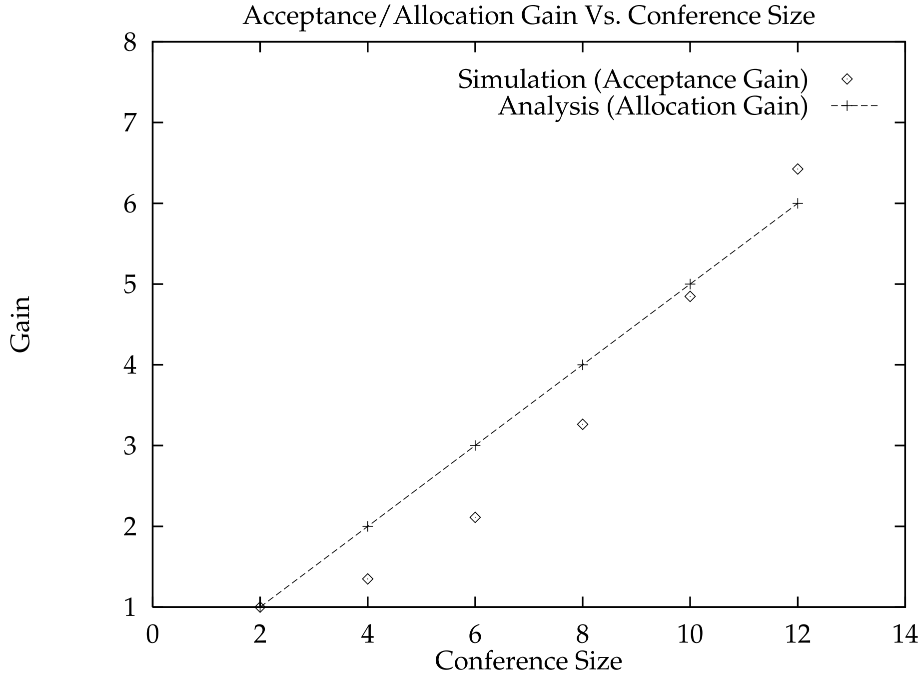

Figure 16: XUNET – Simulation vs. Analysis

We computed the *acceptance gain* from the two curves in Figure 12 by taking the ratio of the acceptance rate for RS with the acceptance rate for non-RS. In Figure 16, this acceptance gain is plotted along with the allocation gain $g = \frac{m}{2t}$ formulated in the analysis. As the graph shows, the simulation results correspond fairly well to the analytical values.

6 Related work

A number of researchers have proposed techniques for providing *probabilistic* and *statistical* guarantees [16, 17] to improve the connection acceptance rates. Instead, resource sharing aims to improve the connection acceptance rate for *deterministic* guarantees by exploiting known relationships between channels. There are only two other related research efforts that we are aware of: the UCSD *filters* [15] and RSVP, the resource reservation protocol [18].

Pasquale et al.have proposed a stream *filter* that is "... an executable module which may be placed on a port, and implements a function which takes a specific set of streams associated with that port and produces a new stream" [15]. These filters perform an *application-level* transformation of one or more streams. A multiplexing filter could perform a function similar to the resource sharing mechanisms described in this paper by taking in streams from upstream nodes and multiplexing them on to a single output stream. It should be noted, however, that the design in this paper does not require application-level modules within the network. We are not aware of any implementation or any further research on these filters.

RSVP is a protocol by which receivers may reserve resources for *flows* (roughly equivalent to channels) that they receive. Along with reservation requests, receivers may specify a *reservation filter*, that describes which flows can use the reservation. The *Wildcard Filter* style corresponds most closely to the model of resource sharing described in this paper, in that the reserved resources may be used to forward data from any source in a given group. Other filtering styles are *Fixed* , in which the receiver specifies a list of enabled sources that *cannot be changed*, and *Dynamic*, in which receivers may dynamically alter the list of enabled sources.

The most important difference between the approach taken by RSVP and the resource sharing mechanisms described here for the Tenet scheme is that in RSVP the receiver determines the level of reservation, while in the Tenet scheme the reservation level is determined by the sender. Since many data streams cannot be scaled arbitrarily without serious degradation in perceived quality (e.g. [13]), we argue that the source should specify the resource requirements (at least in terms of bandwidth). Receiver control over bandwidth requirements can be obtained by using layered coding schemes [11] and putting each layer in a separate sharing group. This approach will be studied in future work.

7 Conclusions and suggestions for future work

We have presented a scheme for resource sharing for guaranteed performance connections in computer networks. The scheme provides fully-distributed low-overhead techniques for implementing resource sharing. We have evaluated the performance of our schemes by analysis and by simulations. Results show that resource sharing is very useful in saving network resources. It achieves both higher connection acceptance rate and lower computational cost for admission control (than without resource sharing), while still providing guaranteed performance to the clients, independent of the behavior of other unrelated traffic.

We understand that routing affects the resource sharing benefits; we are currently exploring the relationship between routing and resource sharing. We expect that this research will help us design specific routing techniques that increase resource sharing benefits. We are also investigating the interactions between resource sharing and advance reservations. Also, a key component of real-time communication service is the traffic specification model which the client uses to characterize the channel traffic. A number of traffic models have been proposed in literature; we have to evaluate resource sharing benefits under different traffic specification models.

Acknowledgement

We want to thank Domenico Ferrari, who guided us in this research. Thanks are due to Riccardo Bettati, Andreas Hoffman, Bruce Mah, Steve Mccanne, Ron Widyono, Brian Whetten, Raj Yavatkar and Hui Zhang for their suggestions and comments on an earlier draft of this report.

This research was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing official policies, either expressed or implied, of the U.S. Government or any of the sponsoring organizations.

References

[1] Tony Ballardine, Paul Francis, and Jon Crowcroft. Core Based Trees (CBT): an architecture for scalable inter-domain multicast routing. In *Proceedings of SIGCOMM 93*, San Francisco, CA, September 1993.

[2] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Math. Stat.*, 23:493–509, 1952.

[3] David Clark, Scott Shenker, and Lixia Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proceedings of ACM SIGCOMM'92*, pages 14–26, Baltimore, Maryland, August 1992.

[4] Domenico Ferrari, Anindo Banerjea, and Hui Zhang. Network support for multimedia: a discussion of the Tenet approach. Technical Report TR-92-072, International Computer Science Institute, Berkeley, California, October 1992. Also to appear in *Computer Networks and ISDN Systems.*

[5] Domenico Ferrari and Dinesh Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[6] Alexander G. Fraser, Chuck R. Kalmanek, A.E. Kaplan, William T. Marshall, and R.C. Restrick. Xunet2: A nationwide testbed in high-speed networking. In *Proceedings of INFOCOM'92*, Firenze, Italy, May 1992.

[7] Amit Gupta, Wendy Heffner, Mark Moran, and Clemens Szyperski. Multi-party real-time communication in computer networks. In *Proceedings of 4th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Lancaster, UK, November 1993.

[8] Amit Gupta, Wingwai Howe, Quyen Nguyen, and Mark Moran. Scalable resource reservation for multi-party real-time communication. Technical Report TR-94-050, International Computer Science Institute, Berkeley, California, October 1994.

[9] E. W. Knightly and G. Ventre. Galileo: a tool for simulation and analysis of real-time networks. In *Proceedings of IEEE 1993 International Conference on Network Protocols*, pages 264–271, San Francisco, CA, October 1993.

[10] Bruce Mah. A mechanism for administration of real-time channels. Master's thesis, Tech. Report UCB/CSD-93-735, University of California, Berkeley, CA, March 1993.

[11] Steve McCanne, October 1994. Personal communication.

[12] Mark Moran and Riccardo Gusella. System support for efficient dynamically-configurable multi-party interactive multimedia applications. In *Proceedings of Thrid International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, CA, November 1992.

[13] Pramod Pancha and Magda El Zarki. A look at the MPEG video coding standard for variable bit rate video transmission. *Proc. IEEE INFOCOM '92*, May 1992.

[14] Colin Parris, Hui Zhang, and Domenico Ferrari. Dynamic management of guaranteed performance multimedia connections, April 1993. to appear in ACM Journal of Multimedia Systems.

[15] Joseph Pasquale, George Polyzos, Eric Anderson, and Vachaspati Kompella. The multimedia multicast channel. In *Proceedings of Third International Workshop on Network and Operating Systems Support for Distributed Audio and Video*, San Diego, CA, November 1992.

[16] Dinesh Verma. *Guaranteed Performance Communication in High Speed Networks*. PhD dissertation, University of California at Berkeley, November 1991.

[17] Hui Zhang and Ed Knightly. Providing end-to-end statistical performance guarantees with interval dependent stochastic models. In *ACM Sigmetrics'94*, Nashville, TN, May 1994.

[18] Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: A new resource reservation protocol. *IEEE Networks Magazine*, 31(9):8–18, September 1993.