# A Performance Study of Multicast Routing Algorithms for ATM Networks

Tim Harrison  Carey Williamson
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, CANADA S7N 5A9
email: {harrison,carey}@cs.usask.ca

## Abstract

*This paper addresses the problem of multicast routing in ATM networks. Formal experimental methods are used to evaluate the relative performance of three simple multicast routing algorithms on simple mesh-based networks, using call-level simulation.*

*The performance study focuses on the scalability of multicast routing algorithms with respect to network size, the level of multicast traffic on the network, and the size of multicast groups.*

*The results show that the choice of a multicast routing algorithm has a major impact on network performance. Shared link algorithms, which route multicast calls along common link channels whenever possible, improve the overall call blocking performance, reduce the call blocking bias against large multicast groups, and scale much better with network size. However, the improved call blocking performance for multicast calls comes at the expense of increased call blocking for unicast calls.*

## 1. Introduction

*Multicast* is the selective broadcast of a message from a source node $S$ to a group $G$ of $N$ recipients, where the group $G$ can consist of an arbitrary subset of the total nodes in a network. Multicast is distinct from *unicast*, which provides point-to-point (*i.e., one-to-one*) communication, and *broadcast*, which provides *one-to-everyone* communication.

The goal in multicast transmission is for a sender to only have to send *one* copy of each message, regardless of how many group members there are. In other words, rather than making multiple copies of each message and sending one copy, in turn, to each recipient, the sender relies on the network itself to make the copies, as needed, at branch points in the *multicast delivery tree*, so that one copy is delivered to each recipient. A related goal is that messages should *only go where they are needed* in the network. Again, the sender relies on the network to achieve this: the network should route a message to each member of the group, and avoid routing any messages into portions of the network where no intended recipients reside.

Multicast thus has several key advantages over broadcast. These include: savings in network bandwidth, since messages are routed only where they need to go; control of information distribution, for the same reason; and fewer incoming messages for most network nodes, since fewer irrelevant messages arrive.

Unfortunately, *optimal* multicast routing in an arbitrary network is not computationally feasible: it is precisely the *Steiner Tree* problem [7], which is known to be NP-complete.

Fortunately, in most computer networks, it is sufficient to find *good* multicast routes, even if the routes are not provably optimal. Heuristics thus play a central role in most computer network routing algorithms, for both unicast and multicast traffic. In fact, there are several proposed routing techniques for multicast network traffic [3, 4, 9, 11, 12].

Efficient multicast routing is important because, even though the number of multicast calls in a network may be quite low, the volume of traffic carried by multicast calls may be quite high. For example, current (unpublished) measurements of Internet traffic suggest that audio and video multicasts on the Internet multicast backbone (MBONE) [2, 5] can account for up to 40% of the bytes transferred, but only 1% of the calls.

Currently, the MBONE is the only major use of multicast in computer networks, and its use is seriously hampered by its large bandwidth requirements. Many parts of the Internet are unable to participate in the MBONE for this reason. Asynchronous Transfer Mode (ATM) networks [10] promise to deliver the bandwidth needed for multimedia applications such as video conferencing and distance education. However, deployment of this new technology is not yet widespread. Furthermore, ATM networks, as currently defined, primarily support point to point connections, not point to multipoint. Thus, even though ATM networks at present are small and relatively simple when compared to the Internet, it is not yet clear how or if Internet multicast routing algorithms will migrate to the ATM environment. Different (*i.e.,* simpler) multicast routing techniques may be needed for ATM networks, at least initially.

This paper presents a performance study of three simple multicast routing algorithms for an ATM network environment with relatively few nodes. They are N-Unicast, Conference Call, and Core-Based Trees [1]. The evaluation of multicast routing algorithms is done using call-level simulation, with call blocking probability and links used per successful call as the primary performance metrics.

The purpose of the performance study is to assess the relative efficiency and scalability of the three approaches. More specifically:

1. What are the performance differences between the different multicast routing algorithms, and at what level of multicast load do these performance differences become significant?
2. How well do multicast routing algorithms scale with network size and multicast group size?
3. What heuristics are most useful in developing efficient multicast routing? For example, does link sharing make any difference? How well do these heuristics work on different routing algorithms?

A secondary purpose of the paper is to illustrate the use of formal experimental methodology in the design of an experiment and the analysis of results [8].

The performance study is conducted on simple mesh-based network topologies. While not necessarily representative of realistic network scenarios, the mesh-based networks do provide two key advantages. First, they provide a network scenario in which optimal routing distances can be computed *a priori*, providing a baseline for evaluating the performance of other routing heuristics. Second, they provide a simple enough environment to interpret and fully understand the results of the performance study.

The remainder of this paper is organized as follows. Section 2 describes the multicast routing algorithms evaluated in the paper. Section 3 describes the simulation experiment designed to evaluate the multicast routing algorithms. Sections 4 and 5 present the results of the performance study. Finally, Section 6 concludes the paper.

## 2. Multicast Routing Models

This section introduces three multicast routing algorithms: N-Unicast, Conference Call, and Core-Based Trees. Initially, the discussion of the algorithms assumes no link sharing in multicast routing, *i.e.,* recipients of multicast calls are routed independently, without sharing links when multiple recipients of a call happen to traverse the same network link. Near the end of this section, the use of link sharing is discussed.

### 2.1. Problem Statement and Assumptions

The purpose of a multicast routing algorithm is to find a routing path from a designated source node $S$ in a network to a designated group of recipient nodes in the network. The performance objectives in multicast routing are to route the call as efficiently as possible (*i.e.,* using as few network links as possible), and to keep the call blocking probability as low as possible.

In our work, the set of multicast recipients is known at the time of call arrival, and remains fixed for the duration of the call. If all intended recipients can be reached successfully without exceeding the network capacity on any network link, then the call is accepted. Otherwise, the call is blocked (rejected), recorded as such, and discarded by the network.

In this paper, we focus our attention only on *greedy* approaches to multicast routing: route the arriving call if a path exists to accommodate the call right now. Otherwise, reject the call. This paper considers three such approaches to multicast routing: N-Unicast, Conference Call, and Core-Based Trees.

### 2.2. N-Unicast Model

The N-Unicast routing algorithm achieves multicast transmission by using $N$ separate unicast transmissions, one to each recipient. That is, the source makes $N$ copies of each message, and sends one copy to each group recipient using unicast. An example of N-Unicast is electronic mail sent to a mailing list in the Internet: a separate copy is routed to each recipient on the distribution list, regardless of where in the network they reside.

N-Unicast is thus *not* a multicast routing algorithm at all, since it violates the goals of multicast that were outlined in Section 1. Nevertheless, it does provide a useful baseline for comparison: if a proposed multicast routing algorithm $A$ provides performance results that are *worse* than N-Unicast, then clearly $A$ is not a practical multicast routing algorithm.

### 2.3. Conference Call Model

The Conference Call routing model uses a specialized conference hub, statically located in the network, as a rendezvous point for the sender and the recipients. This model is used in the telephone system for (voice) conference calls. Participants must dial in to a pre-assigned number on a pre-assigned conference hub (bridge). For economic reasons, conference bridges are usually available only in selected (large) metropolitan areas. As a result, sub-optimal routing can occur when receivers are clustered elsewhere in the network.

In our (non-shared link) version of the Conference Call model, the source of a multicast call sends *one* copy of a message to the conference hub. The hub node is responsible for replicating the message, and delivering one copy separately to each of the $N$ recipients (using N-Unicast). In our simulator, the number of hubs can be scaled with network size (*e.g.,* 1 hub node in a small 4 x 4 mesh, and 4 equally spaced hub nodes in a large 10 x 10 mesh).

The purpose of the multiple hubs in the larger meshes is twofold. First, the use of regional hubs roughly equalizes the number of source nodes serviced by a single hub regardless of mesh size. Second, regional hubs help prevent the formation of a traffic hot spot at the center of the mesh. For any given multicast call, only one of the regional hubs is used, chosen based on the location of the multicast source node. Only multicast calls (not unicast calls) need to route via the conference hubs. Corner nodes or edge nodes are never used as hubs.

### 2.4. Core-Based Trees Model

The Core-Based Trees (CBT) [1] model is similar to the conference call model in that it uses a hub (core) to effect the delivery of multicast messages. However, rather than using a static hub, the CBT model chooses the location of the hub *dynamically* for *each* multicast call, based on where the source and recipients reside. The source then sends one copy of each message to the hub (core), and the hub is responsible for replicating the message and sending a copy separately to each group member (using N-Unicast).

In our implementation, the location of the hub is determined by averaging the X and Y coordinates of all group participants (source and recipients) and rounding off to the nearest node in the mesh. With this approach, corner nodes in the mesh can never be chosen as hubs. Edge nodes can be chosen as a hub only if a large number of group participants lie along the same edge of the mesh. Thus the vast majority of hub nodes chosen will be interior nodes of the mesh. In every case, the hub nodes will always be relatively central to the source and recipient nodes.

### 2.5. Link Sharing

The multicast routing algorithms described thus far have routed from source to destination without considering that two or more routes from a source or hub (within the same multicast call) may traverse the same network link. That is, they do *not* fully exploit the true potential of multicast as described in Section 1. With link sharing, multicast calls that use common links in routing to recipients of the group require only *one* unit of channel capacity on those common links, regardless of how many recipients use that link.

In this study, the Conference Call and Core-Based Trees routing models were simulated both with and without the link sharing heuristic.[1] While the non-shared link versions of these algorithms are not very realistic, including them in the simulation study has several advantages. First, they make the initial task of simulation validation easier. Second, they provide a baseline for comparison and interpretation of simulation results. Third, they make it possible to quantify the effects of the link sharing heuristic separately from other effects, such as static versus dynamic hub placement. Finally, they provide an indication of the performance limitations of existing first-generation ATM networks, which do not (yet) support ATM multicast.

## 3. Experimental Design

### 3.1. Overview

The objective of this research is to evaluate the relative performance of ATM multicast routing algorithms. The evaluation is done using call-level simulation on simple mesh-based networks. The routing models evaluated are N-Unicast, Conference Call, and Core-Based Trees. Modified versions of the latter two models that use link sharing whenever possible are also evaluated.

[1] By its definition, the N-Unicast algorithm does not permit link sharing. Thus it is not included in the link sharing experiments.

## 3.2. Workload Generation

A workload generator was used to construct the randomly generated call workloads for the simulation study. Each simulation run used 100,000 calls.

The input parameters to the workload generator were the mesh size (D), the percentage of multicast traffic, the mean multicast group size (g), the mean call holding time, and the mean call interarrival time.

The generated calls were represented as lines of data in a file as shown in Figure 1. The purpose of using workload files was to provide *exactly the same workload* for each routing algorithm. Each line of data corresponded to a unicast or multicast call. The information contained in each call consisted of the call start time in seconds, the duration of the call in seconds, the number of the node that originated the call, the number of group recipients, and the list of recipients. Each of the five (logical) columns of data was generated using an independent random number stream.

```
16708.941   122.551   31   1  20
17591.751   174.804   24   1  15
17638.253   389.275   17   8  24 15 3 25 5 33 30 35
17714.681   893.331   22   1  4
17835.537   416.170   11   1  26
18561.058   744.052    2   5  20 35 8 9 4
```

Start Time (sec.) | Duration (sec.) | Origin | Group Size | List of Recipients

**Figure 1. Example of a Call Workload File**

The following assumptions were made about the workload: call arrivals are Poisson; call holding times are exponentially distributed; all calls require 1 unit of channel capacity; sources and recipients are picked uniformly at random; and multicast group sizes are geometrically distributed.

Most of these are standard modeling assumptions, which, while not necessarily representative of traffic on real networks, served to simplify the simulation experiment significantly.

Several other assumptions require further elaboration. First, a geometric distribution for the multicast group size was chosen so that the model would generate a large number of small multicast groups, and a small number of large multicast groups. The input parameter specifies only the mean for the multicast group size distribution. Second, the network model assumes that a recipient is able to receive multiple calls simultaneously, and that a source can receive calls while sending. Third, the source and recipients for a given call must all be distinct. Finally, multicast calls with fewer than two recipients were not permitted.

## 3.3. Performance Metrics

The performance metrics for the simulation evaluation of multicast routing algorithms were:

- Call Blocking Probability (CBP). Call blocking was used as one indicator of how efficient a call routing model was. Both the overall (cumulative) value of this metric for all (unicast and multicast) calls and the individual values broken down according to recipient group size were recorded. The cumulative value was useful for comparing routing models, while the detailed values identified group size dependency in the call blocking behaviour.
- Average links used per call. The average number of links used per successful call in unicast and multicast calls was another indicator of how efficient a call routing model was. The data from this metric was also broken down by recipient group size.

## 3.4. Factors and Levels

The primary factors and corresponding levels used in the experiment were as follows:

| | |
|---|---|
| Multicast (MC) routing model: | N-Unicast (N-UNI) |
| | Conference Call (CC) |
| | Core-Based Trees (CBT) |
| | CC & Link Sharing (CC-LS) |
| | CBT & LS (CBT-LS) |
| Mesh size: | 4 x 4, 6 x 6, 8 x 8, 10 x 10 |
| Percent MC traffic: | 10%, 20% |
| Mean MC group size: | 4, 8 |

In addition, three non-factor parameters were used in the experiment. The link capacity for all network links was set at 5 channels (active calls) per link. The mean call duration was 500 seconds. The mean call interarrival time was chosen relative to the mean call duration to scale the average number of active calls in the network ($D^2/2$) with the size of the network ($D$). Note that this results in different workloads for each mesh size. Thus it does not make sense to compare call blocking performance for a given routing algorithm across different mesh sizes, but it does make sense to compare the call blocking probability of different routing algorithms for the same mesh size (since they have exactly the same workload).

## 3.5. Experimental Design

A full factorial experimental design was used [8]. A total of 80 test runs were performed, corresponding to five routing models, four mesh sizes, two levels of multicast traffic, and two settings for the mean multicast group size. Sixteen workloads were utilized, four for each size of mesh. Each workload was run on all five routing models.

Only a subset of the experimental results are reported in this paper. Complete results appear in [6].

### 3.6. Other Issues

The simulation run length was chosen to be 100,000 calls. This value was chosen to achieve a balance between statistically significant results and reasonable computation time. The simulation models were started in an initially empty network *i.e.,* there was no preloading of the network with initial calls. Finally, the method of *batch means* [8] was used to obtain confidence intervals on simulation results.

All performance differences reported in Sections 4 and 5 for the three routing algorithms are statistically significant. While confidence intervals are not shown in the graphs reported in this paper (to reduce cluttering), means and standard deviations for call blocking results are reported in tabular form.

## 4. Results: No Link Sharing

This section presents the simulation results for the non-shared link multicast routing models. The results for these models are characterized by call blocking and link usage behaviours that readily allow comparisons between these models. In contrast, the shared link routing models had distinctly different call blocking and link usage characteristics. The simulation results for the latter models are discussed in Section 5.

### 4.1. N-Unicast Model

In the N-Unicast routing model, the source of a multicast call makes a separate unicast transmission to each of the N recipients of the call. This section presents the simulation results for the N-Unicast algorithm for 10% multicast traffic when the mean multicast group size (g) is 4 recipients.

#### 4.1.1. Call Blocking Results

Table 1 summarizes the call blocking results for the N-Unicast algorithm. For the 10% multicast traffic workload, the N-Unicast routing model resulted in an average call blocking probability of 1.0% on a 4 x 4 mesh. The average call blocking for the other mesh sizes (for different workloads) was 1.5% for a 6 x 6 mesh, 2.6% for an 8 x 8 mesh, and 4.8% for a 10 x 10 mesh.

The overall call blocking probability, however, provides very little insight into the routing behaviour of the algorithm. Further insight is gained by separating out the call blocking results for unicast calls and multicast calls, as shown in Table 1. Unicast calls had a much lower call blocking probability than multicast calls. Even further insight can be gained by analyzing the call blocking results for each (randomly generated) recipient group size within the workload file.

Figure 2 illustrates a breakdown of the call blocking probability as a function of the number of recipients for each call generated. Large multicast groups clearly had a much higher call blocking probability than small groups. For example, unicast calls on a 4 x 4 mesh had a call blocking probability of 0.3%, while multicast calls with groups of size 2, 5, and 10 had call blocking probabilities of 0.7%, 3.4%, and 36.9%, respectively. The bias against large multicast groups makes sense intuitively: the larger the multicast group, the less likely it is that all group members can be reached successfully when routing the call. If all recipients cannot be successfully reached, the call is blocked (the "all or nothing" approach).

For a given multicast group size, the probability of call blocking usually increased with mesh size. The reason for this behaviour is that the scaling of workload with mesh size results in an increase in network congestion.

#### 4.1.2. Link Usage Results

The N-Unicast routing model was found to require an average of 3.5 links per call for a 4 x 4 mesh with a 10% multicast traffic workload. Similarly, the 6 x 6 mesh needed 5.4 links per call on average. The 8 x 8 mesh used an average of 7.2 links per call, and the 10 x 10 mesh averaged 9.1 network links per call. Again, however, this single metric provides very little information on routing efficiency. A further breakdown of the results is required.

The plots of link usage with respect to recipient group size are shown in Figure 3. The link usage grows linearly for all mesh sizes. This linearity was expected because in a non-shared link routing model such as N-Unicast, the number of network links used increases in direct proportion to the number of group recipients. Table 2 shows the results of a linear regression fit to each of the lines in Figure 3. All results indicate strong linearity (as expected).

Figure 3 and Table 2 also show that the slope of the line depends on mesh size. This is obvious since larger

**Table 1. Mean and Standard Deviation of the Call Blocking Probability for N-Unicast Model (10% multicast, g = 4)**

| Mesh Size | Call Blocking Probability for N-UNI Model | | | | | |
|---|---|---|---|---|---|---|
| | Unicast | | Multicast | | All Calls | |
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| 4 | .0029 | .0008 | .0746 | .0066 | .0100 | .0012 |
| 6 | .0053 | .0015 | .0983 | .0106 | .0146 | .0020 |
| 8 | .0122 | .0024 | .1541 | .0115 | .0264 | .0028 |
| 10 | .0272 | .0038 | .2351 | .0145 | .0480 | .0044 |

**Table 2. Link Usage Regression Results for N-Unicast Model (10% multicast, g = 4)**

| Mesh Size | Linear Regression | | |
|---|---|---|---|
| | Slope | $y$ intercept | $R^2$ |
| 4 | 2.61 | 0.57 | 0.996 |
| 6 | 3.52 | 3.84 | 0.947 |
| 8 | 5.77 | -0.02 | 0.993 |
| 10 | 8.52 | -5.55 | 0.963 |

meshes have greater average distances from source to recipient than smaller networks.

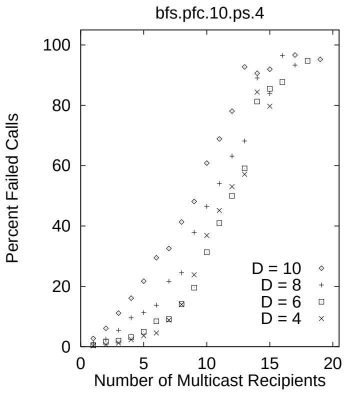


**Figure 2. Call Blocking Probability, N-UNI model, 10% multicast, g = 4.**

## 4.2. Conference Call Model

The Conference Call model uses statically located hubs to route a multicast call. The simulation results presented in this section are for the Conference Call model with 10% multicast traffic and a mean multicast group size of 4 recipients.

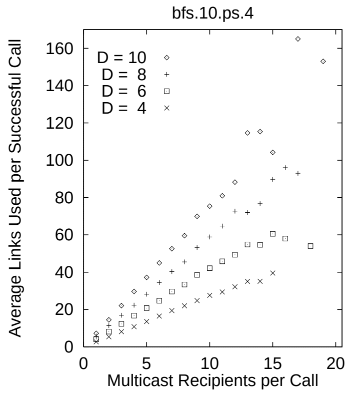


**Figure 3. Avg. links per call, N-UNI model, 10% multicast, g = 4.**

### 4.2.1. Call Blocking Results

The call blocking results for the Conference Call model are summarized in Table 3. As with the N-Unicast model, call blocking probability increased with mesh size, because of the workload scaling effects. A breakdown of call blocking probability as a function of group size is shown in Figure 4. The call blocking probability again shows a strong bias against large groups.

Figure 4 shows that the call blocking values for the Conference Call model were generally higher than those for the N-Unicast model on large meshes ($D = 8$ and $D = 10$). The explanation for the higher overall call blocking probability for the large mesh sizes is that the links surrounding the fixed conference hubs became bottlenecks for routing multicast calls.

For smaller meshes ($D = 4$ and $D = 6$), however, the behaviour is more interesting. For small group sizes, the Conference Call model had higher call blocking than the N-Unicast algorithm, and for larger group sizes, the Conference Call model had lower call blocking than the N-Unicast algorithm. This "cross-over" between the two algorithms takes place at a group size of 10 recipients.

This cross-over effect is due to "hot spot" effects and "edge and corner node" effects in the mesh topologies simulated. For small multicast groups, the Conference Call model experiences higher call blocking due to congestion at the conference hub(s), which *must* be used for routing multicast (but not unicast) calls. The N-Unicast algorithm is not subject to this hot spot problem, and thus performs better for small multicast groups. For large multicast groups, the N-Unicast algorithm blocks more calls because of edge and corner node effects. For example, a corner node can *never* succeed in routing a multicast call with more than 10 recipients, because of the assumptions in our network model (*i.e.,* two outgoing links, each of capacity 5, for a total of 10 units of capacity). The Conference Call model is not subject to this limitation, since all multicast calls are routed to the conference hub before branching out for delivery, and conference hubs are always *interior* nodes of the mesh (with four outgoing links). These two opposing effects account for the cross-over between the two algorithms.

#### 4.2.2. Link Usage Results

Three trends regarding link usage in the N-Unicast model were observed in the Conference Call model. First, the average link usage per call increases with the number of recipients. Second, the relationship of average link usage per call to recipient group size is linear. Finally, the slope of the line increases with mesh size. A summary of linear regression results for the Conference Call model appears in Table 4.

The link usage values for the Conference Call model were less than or equal to those for the N-Unicast model[2], with the difference more pronounced on large meshes. A small savings in link usage was expected for the Conference Call model, since the conference hub is usually closer to the recipients than the source node (particularly in large meshes with regional hubs).

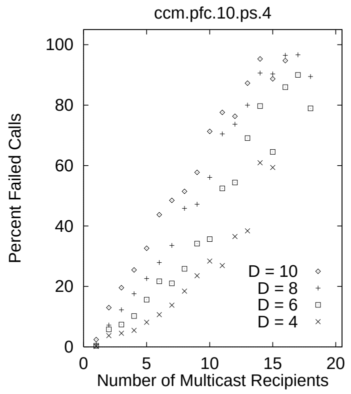


**Figure 4. Call Blocking Probability, CC model, 10% multicast, g = 4.**

**Table 4. Link Usage Regression Results for Conference Call Model (10% multicast, g = 4)**

| Mesh Size | Linear Regression | | |
|---|---|---|---|
| | Slope | $y$ intercept | $R^2$ |
| 4 | 2.25 | 1.35 | 0.999 |
| 6 | 3.75 | 1.66 | 0.994 |
| 8 | 5.09 | 1.14 | 0.982 |
| 10 | 6.99 | 2.56 | 0.991 |

**Table 3. Mean and Standard Deviation of the Call Blocking Probability for Conference Call Model (10% multicast, g = 4)**

| Mesh Size | Call Blocking Probability for the CC Model | | | | | |
|---|---|---|---|---|---|---|
| | Unicast | | Multicast | | All Calls | |
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| 4 | .0015 | .0006 | .0957 | .0052 | .0109 | .0007 |
| 6 | .0034 | .0011 | .1702 | .0141 | .0201 | .0020 |
| 8 | .0076 | .0012 | .2409 | .0123 | .0309 | .0022 |
| 10 | .0239 | .0027 | .3247 | .0203 | .0540 | .0042 |

### 4.3. Core-Based Trees Model

The Core-Based Trees (CBT) model is similar to the Conference Call model in that a hub (core) is used to replicate packets of the call and dispatch them to the recipients. In the CBT model, the core location for multicast calls is dynamically chosen based on the locations of the source and recipients. This model was expected to achieve lower call blocking probabilities and link usage than the Conference Call model since the core node would always be placed relatively central to the group of nodes containing the source and the recipients. In this section, the simulation results presented are for the Core-Based Trees model with a mean multicast group size of 4, and 10% multicast traffic.

[2] A precise comparison of link usage results for the two algorithms is somewhat dangerous, since the call blocking behaviours are different. Nevertheless, the link usage differences reported here are statistically significant.

#### 4.3.1. Call Blocking Results

Table 5 summarizes the call blocking results for the CBT routing model. As with the other two non-shared link models, the call blocking probability increased with mesh size and recipient group size. The performance results are shown graphically in Figure 5.

The call blocking probability of the CBT model is sometimes lower and sometimes higher than that for the N-Unicast and Conference Call models, depending on the mesh size. It appears that the non-shared link CBT model routes unicast calls at the expense of multicast calls in large meshes.

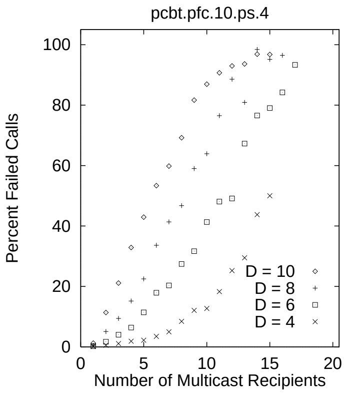


**Figure 5. Call Blocking Probability, CBT model, 10% multicast, g = 4.**

### 4.3.2. Link Usage Results

It was again observed that the average links used per call increased linearly with group size, and for a given group size, link usage increased with mesh size. Linear regression results are summarized in Table 6. As expected, the CBT model consistently had the lowest average link usage of the three non-shared link routing models. Link usage savings with the CBT model also improved with mesh size.

## 4.4. Effect of Level of Multicast Traffic

As expected, call blocking and link usage increased for all three routing models when the level of multicast traffic increased from 10% to 20%. Due to the strong similarity between the results for 10% and 20% multicast workload, the results for 20% multicast workload are not shown in this paper, for space reasons. Complete results are available in [6].

**Table 5. Mean and Standard Deviation of the Call Blocking Probability for Core-Based Trees Model (10% multicast, g = 4)**

| Mesh Size | Call Blocking Probability for the CBT Model | | | | | |
|---|---|---|---|---|---|---|
| | Unicast | | Multicasts | | All Calls | |
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| 4 | .0009 | .0005 | .0407 | .0040 | .0049 | .0008 |
| 6 | .0023 | .0007 | .1409 | .0104 | .0162 | .0014 |
| 8 | .0050 | .0006 | .2461 | .0136 | .0291 | .0013 |
| 10 | .0124 | .0019 | .3844 | .0173 | .0496 | .0030 |

**Table 6. Link Usage Regression Results for Core-Based Trees Model (10% multicast, g = 4)**

| Mesh Size | Linear Regression | | |
|---|---|---|---|
| | Slope | $y$ intercept | $R^2$ |
| 4 | 2.26 | -0.33 | 0.996 |
| 6 | 3.17 | 0.66 | 0.997 |
| 8 | 4.30 | 0.93 | 0.985 |
| 10 | 5.97 | -1.54 | 0.971 |

## 4.5. Effect of Multicast Group Size

Increasing the mean multicast group size from 4 to 8 increased call blockage for both the 10% and 20% multicast traffic workloads. The effect was most pronounced in small mesh sizes. Unlike the situation with call blocking, however, the increase in mean multicast group size had minimal effect on average link usage.

## 4.6. Summary

The experiments with the three non-shared link routing models revealed several trends:

- Large multicast groups had a much higher call blocking probability than small groups when all other factors were held constant.
- The relationship between average number of links used per successful call and group size is strongly linear, with a slope depending primarily on mesh size, but also on the routing algorithm.
- Increasing the mean multicast group size in the workload increased call blockage, but had minimal effect on link usage.

# 5. Results: Link Sharing

This section presents the simulation results for shared link multicast routing models. The effect of the link sharing heuristic is easily discernible since it is the only modification made to the routing models mentioned. By its definition, the N-Unicast algorithm does not permit link sharing. Thus it is not included in the experiments in this section.

## 5.1. Conference Call Model with Link Sharing

The shared link Conference Call model attempts to route multicast calls using common link channels whenever possible. That is, if two or more recipients lie in the same general direction from the Conference Call hub, then a common path (requiring only one unit of channel capacity) is used to route toward those recipients. Routes diverge from this common path only as necessary to reach the individual recipients. The use of the link sharing heuristic should reduce link usage and call blocking. The simulation results discussed in this section are for the Conference Call model with link sharing, for 10% multicast traffic and a mean multicast group size of 4 recipients (g = 4).

#### 5.1.1. Call Blocking Results

Call blocking results for the shared link Conference Call model appear in Table 7. A breakdown of call blocking probability as a function of group size is shown in Figure 6.

Link sharing reduces call blocking and reduces the bias against large multicast groups. The average values of call blocking probability in Figure 6 are much lower than those in Figure 4 for the non-shared link version of this model.

The shared link model also allows larger multicast groups to be connected, since calls with more than 25 recipients are being successfully routed, as seen in Figure 6. In contrast, the largest successful group in the non-shared link Conference Call model with the same workload had only 18 recipients (Figure 4). Link sharing obviously increases the maximum size of groups that can be routed, since the constraint that the number of recipients is at most four times the link channel capacity no longer applies.

The improved call blocking performance for multicast calls in the shared link Conference Call model comes at the expense of unicast calls. For example, the call blocking probability for unicast calls on the 10 x 10 mesh was 3.1%, compared with 2.4% for the non-shared link model. It appears that once a certain mesh size is reached (10 x 10 in this case), multicast calls are less likely to be blocked, but at the expense of *increased* unicast call failures. This effect was even more pronounced in the workloads with 20% multicast traffic (*e.g.,* 7.3% unicast call blocking with link sharing, versus 3.6% call blocking without link sharing).

**Table 7. Mean and Standard Deviation of the Call Blocking Probability for the Shared Link Conference Call Model (10% multicast, g = 4)**

| Mesh Size | Call Blocking Probability for CC-LS Model | | | | | |
|---|---|---|---|---|---|---|
| | Unicast | | Multicast | | All Calls | |
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| 4 | .0002 | .0001 | .0003 | .0005 | .0002 | .0001 |
| 6 | .0012 | .0006 | .0045 | .0025 | .0015 | .0006 |
| 8 | .0071 | .0014 | .0203 | .0050 | .0084 | .0014 |
| 10 | .0306 | .0045 | .0759 | .0100 | .0351 | .0046 |

**Table 8. Link Usage Regression Results for Shared Link Conference Call Model (10% multicast, g = 4)**

| Mesh Size | Linear Regression | | |
|---|---|---|---|
| | Slope | $y$ intercept | $R^2$ |
| 4 | 2.05 | 1.68 | 0.999 |
| 6 | 3.47 | 1.96 | 0.996 |
| 8 | 4.73 | 3.75 | 0.926 |
| 10 | 7.02 | 2.59 | 0.947 |

#### 5.1.2. Link Usage Results

As with the non-shared link routing models, the relationship between average link usage per call and multicast group size is linear. Average link usage per call increases with group size, and link usage increases with mesh size. Regression results appear in Table 8. The slopes in Table 8 are generally lower than those in Table 4 for the non-shared link Conference Call model.

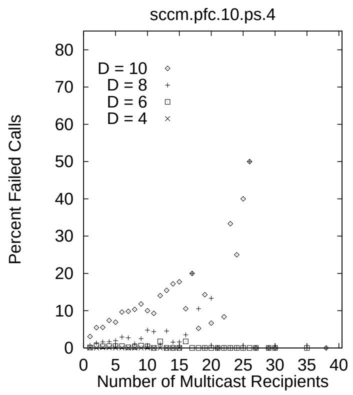


**Figure 6. Call Blocking Probability, CC-LS model, 10% multicast, g = 4.**

### 5.2. Core-Based Trees Model with Link Sharing

The shared link Core-Based Trees (CBT) model differed from its non-shared link counterpart by sharing common links in a multicast call whenever possible. The results discussed in this section are for the CBT model with link sharing, for 10% multicast traffic and a mean multicast group size of 4 recipients.

### 5.2.1. Call Blocking Results

A summary of the call blocking results for the shared link CBT routing algorithm appears in Table 9. These average values were lower than those for the shared link Conference Call model. As with all the other models, the call blocking probability increased with mesh size, and for a given workload, call blocking probability increased with the number of recipients in a group. A breakdown of call blocking probability as a function of group size appears in Figure 7. As with the shared link Conference Call model, larger successful multicast groups were possible in the shared link CBT model compared with the non-shared link version.

**Table 9. Mean and Standard Deviation of the Call Blocking Probability for Shared Link CBT Model (10% multicast, g = 4)**

| Mesh Size | Call Blocking Probability for the CBT Model | | | | | |
|---|---|---|---|---|---|---|
| | Unicast | | Multicast | | All Calls | |
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| 4 | .0000 | .0001 | .0004 | .0007 | .0001 | .0001 |
| 6 | .0012 | .0005 | .0036 | .0025 | .0014 | .0006 |
| 8 | .0060 | .0010 | .0217 | .0037 | .0075 | .0010 |
| 10 | .0231 | .0029 | .0856 | .0130 | .0293 | .0034 |

In the shared link CBT model, unicast calls were blocked more frequently than in the non-shared link CBT model, on large meshes. For example, the call blocking probability for unicast calls in the shared link CBT model was 2.3% on the 10 x 10 mesh, while in the non-shared link CBT model unicast calls were blocked only 1.2% of the time. Again, more multicast calls are being routed, but at the expense of unicast calls.

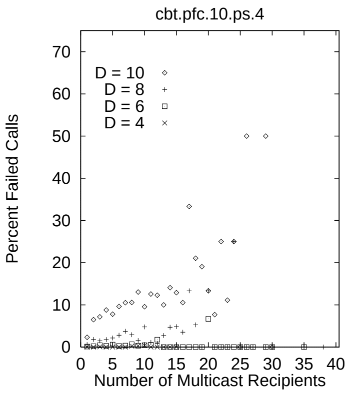


**Figure 7. Call Blocking Probability, CBT-LS model, 10% multicast, g = 4.**

**Table 10. Link Usage Regression Results for Shared Link CBT Model (10% multicast, g = 4)**

| Mesh Size | Linear Regression | | |
|---|---|---|---|
| | Slope | $y$ intercept | $R^2$ |
| 4 | 2.07 | 0.27 | 0.999 |
| 6 | 3.14 | 0.46 | 0.995 |
| 8 | 4.16 | 3.17 | 0.974 |
| 10 | 5.71 | 2.58 | 0.966 |

### 5.2.2. Link Usage Results

Once again, the relationship between average link usage per call and multicast group size was linear (refer to Table 10). The average link usage per call increased with group size, and link usage increased with mesh size. The slopes were marginally lower than those for the non-shared link CBT, and significantly lower than those for the shared link Conference Call model.

## 5.3. Effect of Multicast Group Size

This section discusses the effect of mean multicast group size on call blockage and link usage for the shared link routing models.

The general trends described in the previous sections continued to be observed when the mean multicast group size was raised from 4 to 8. However, increasing the mean multicast group size increased average link usage significantly, with minimal impact on call blocking probability. This behaviour is in direct contrast to that of the non-shared link routing models. That is, the non-shared link models tend to block large multicast calls and increase call blocking (with no effect on link usage), while the shared link models tend to route large multicast calls and increase link usage (with no effect on call blocking).

## 5.4. Summary

The experimental results for the shared link models revealed the following:

- The shared link models resulted in fewer call failures, as expected, and had much less bias against large multicast groups.
- The shared link models allowed larger multicast group sizes than their non-shared link counterparts, but this was at the expense of more links being required to route these calls.
- The shared link models blocked unicast calls more frequently than did the non-shared link models for

large mesh sizes. That is, the reduced blocking of multicast calls in the shared link models came at the expense of unicast calls in large meshes.

## 6. Conclusions

There are three general conclusions that can be drawn from this research.

First, the choice of a multicast routing algorithm in the networks studied has a major impact on the call blocking performance and link usage observed. These performance differences are quite evident even when only 10% of the calls in the network workload are multicast calls. The experiments reported in this paper have shown that simplistic approaches to multicast call routing, such as N-Unicast or the Conference Call model, do not scale well with network size, with multicast load, or with mean multicast group size. More sophisticated routing models, such as Core-Based Trees, offer much more potential for scalability in multicast routing.

Second, among the routing heuristics considered for multicast traffic, the link sharing heuristic is the most important. Experiments have shown that dynamic core placement in the CBT model can somewhat alleviate the hot spots inherent in the Conference Call and N-Unicast models, but it is only with the addition of the link sharing feature that call blocking is dramatically reduced. Link sharing reduces (but does not totally eliminate) the bias against large multicast groups, and permits the routing of larger multicast calls than is possible in the non-shared link routing models.

Third, there is a tradeoff to be made between the routing of multicast calls and unicast calls in the network. Indeed, in several experiments, the overall reduction in call blocking observed for shared link models came from reduced multicast call blocking, at the expense of increased unicast call blocking. For the network manager, balancing this tradeoff becomes an important issue. Under some circumstances it may be better to block a large multicast call to allow several smaller multicast or unicast calls, and vice versa. A multicast call pricing schedule that is linear in the recipient group size may be one way for a network provider to balance this tradeoff.

This paper presents a starting point for the evaluation of multicast routing algorithms for ATM networks, but much future work remains. Future work will consider more realistic network topologies, more realistic traffic flows, dynamic multicast groups, and more sophisticated routing algorithms.

**Acknowledgements**

Financial support for this research was provided by the Natural Sciences and Engineering Research Council (NSERC) through research grants OGP0121969 and IOR152668, and by Telecommunications Research Laboratories (TR*Labs*) in Saskatoon.

## References

[1] T. Ballardie, P. Francis, and J. Crowcroft, "Core-Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing", *Proceedings of ACM SIGCOMM'93*, San Francisco, CA, pp. 85-95, September 1993.

[2] S. Casner and S. Deering, "First IETF Internet Audiocast", *ACM Computer Communications Review*, Vol. 22, No. 3, pp. 92-97, July 1992.

[3] S. Deering, *Multicast Routing in a Datagram Internetwork*, Ph.D. thesis, Stanford University, 1991.

[4] S. Deering, D. Estrin, D. Farinicci, V. Jacobson, C. Liu, and L. Wei, "An Architecture for Wide-Area Multicast Routing", *Proceedings of ACM SIGCOMM'94*, London, UK, pp. 126-135, August 1994.

[5] H. Eriksson, "MBONE: The Multicast Backbone", *Communications of the ACM*, Vol. 37, No. 8, pp. 54-60, August 1994.

[6] T. Harrison, *Performance Evaluation of Routing Algorithms for Multicast Traffic*, M.Sc. thesis, Department of Computer Science, University of Saskatchewan, August 1995. (Available by anonymous ftp to ftp.cs.usask.ca as pub/discus/thesis-harrison.ps.Z)

[7] F. Hwang and D. Richards, "Steiner Tree Problems", Technical Report, Bell Laboratories, *c.* 1987.

[8] R. Jain, *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, Incorporated, New York, 1991.

[9] V. Kompella, J. Pasquale, and G. Polyzos, "Multicast Routing for Multimedia Communications", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 286-292, June 1993.

[10] S. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications*, Vol. 27, No. 9, pp. 17-24, September 1989.

[11] D. Verma and P. Gopal, "Routing Reserved Bandwidth Multi-point Connections", *Proceedings of ACM SIGCOMM'94*, San Francisco, CA, pp. 96-105, September 1993.

[12] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988.