EVALUATION OF MULTICAST ROUTING ALGORITHMS FOR MULTIMEDIA STREAMS*

Ciro A. Noronha Jr. Fouad A. Tobagi

Department of Electrical Engineering
Stanford University
Stanford, CA 94305

Abstract

In this paper we present an evaluation of multicast routing algorithms for multimedia streams, under realistic network and traffic scenarios. Multimedia streams have different requirements than traditional data traffic, which should be taken into account by the routing algorithm. We discuss these requirements, describe the multicast routing algorithms, and specify the scenarios used in the evaluation. The evaluation scenarios are composed of different network topologies, either corresponding to deployed networks, or generated at random, and to different kinds of offered traffic. Simulation results, comparing a number of multicast routing algorithms under uniform conditions, are given.

1 Introduction

Multimedia represents the integration of a variety of media, such as data, video, audio and still images. The traffic underlying networked multimedia applications has different requirements than that underlying traditional data applications. These differences are:

Bandwidth - multimedia streams use relatively high bandwidth on a continuous basis for long periods of time, while the average bandwidth used by data applications is low; for example, a high-quality compressed video stream can use anywhere from 1.5 to 8 Mb/s;

Multipoint Communications - it is expected that a significant fraction of the multimedia traffic will be multipoint (generated from video-conferencing applications, for example); and

Low Latency (on the order of 100-200 ms end-to-end), required for interactive communication.

Multicast is the name applied to the unidirectional communication from one source to multiple destinations. The high bandwidths required to transmit real-time audio/video streams suggest that the routing of such streams should be performed as efficiently as possible, to minimize the usage of network resources and maximize the number of multicasts the network is able to carry. The focus of this work is on evaluating multicast routing algorithms under realistic network and traffic scenarios, taking into account the specific requirements in terms of bandwidth, multicasting, and latency underlying multimedia traffic.

In section 2, we present the general formulation of the problem and section 3 contains the routing algorithms considered in this paper. In section 4 we briefly discuss the previous work in the area, and section 5 describes the evaluation scenarios. The evaluation results are presented in section 6, and section 7 has the conclusions.

2 Problem Formulation

A *stream* is a continuous flow of information (i.e., video frames or audio samples) that have to be delivered in a timely fashion; streams can either be constant bit rate or variable bit rate, but they are much less bursty than data. In a computer network, streams are divided into packets for transmission. Several streams can be multiplexed (in time) and sent through a channel, each stream using a fraction of the channel's bandwidth. A *multicast stream* is a one-to-many communication, whereby a single source sends information to a number of destinations. For efficiency reasons, a channel should never carry more than one copy of a stream; the network is responsible for replicating the stream at the appropriate places to reach all the destinations. Given the network and the traffic requirements, the *multicast routing algorithm* is responsible for finding a

*This work was in part supported by NASA under grant NAGW-419, by NSF under grant NCR-9016032 and by Pacific Bell. Ciro Noronha was supported by a graduate scholarship from FAPESP from Sept/89 to Aug/93 under grant 89/1658.

route in the network that satisfies the requirements, while optimizing a certain objective function, which can be the delay, the cost or a combination thereof. If no route can be found, the request can either be dropped, or wait until the necessary resources are free; in this paper, we assume that blocked sessions are cleared (dropped). In the following, we define the network parameters and the traffic requirements.

2.1 The Network

The network is a collection of *nodes*, interconnected by *links* subject to a certain *topology*. In this paper, we consider all links to be point-to-point and directed (i.e., information can flow in only one direction); the reader is referred to [1] for a discussion of how other kinds of topologies (such as shared-medium and WDM networks) can be accommodated. Full-duplex connections between a pair of nodes correspond to two independent links, one in each direction, since multicasts are unidirectional, and the bandwidth is managed separately in each direction. Each link is characterized by the following parameters:

Capacity: Link bandwidth, in bits/second.

Cost: Monetary cost of using the link, in \$/bit. In other words, the cost of routing a stream of r bits/second over a link costing C \$/bit is rC \$/second.

Delay: Each link has associated with it a propagation delay τ, which is proportional to the physical length of the link. The delay experienced by a packet when crossing the link is the propagation delay, plus the packet transmission time and the time spent waiting for the link in a queue (queueing delay). We assume that the streams have priority over other traffic; therefore, the queueing delay is limited and typically small. The total delay of a link, denoted by D, is assumed to be fixed (independent of the load in the link).

2.2 The Traffic

Multicast streams arrive at the network in *sessions*. A session is a group of multicasts which are logically related. One example is a video-conference, where each of the participants can see all other conferees; if there are P participants, a video-conference session would be composed of P multicasts, one from each of the participants to the other $P-1$ conferees.

A multicast session is characterized by the following parameters:

Arrival Process: Characterizes the multicast session arrivals.

Session Duration: If accepted, a multicast session will be established and will remain in the network for a certain amount of time; this time is a random variable.

Session Components: A session is composed of T multicast streams; the number T can be a random variable. Each of the multicasts in the session is characterized by:

Addressing Parameters: Source and set of destinations. The number of destinations can be a random variable. In some cases, the set of destinations can change in time, as nodes join and leave the multicast.

Bandwidth Requirement: The stream data rate.

Latency Constraint: The maximum delay acceptable between the source and any of the destinations in this multicast.

2.3 The Problem Definition

When a session arrives, the routing algorithm is responsible for finding routes for each of its multicasts; the routes should have enough free bandwidth to support the streams, and should satisfy the latency constraints. The algorithm selects the routes to optimize a certain objective function, which could be either delay or cost or a combination thereof. In this paper, we compare multicast routing algorithms under a uniform criterion, the probability that a session arrives to the network and is blocked, i.e., a route satisfying the constraints cannot be found.

3 Multicast Routing Algorithms

We can classify the multicast routing algorithms in three groups:

Shortest Path Algorithms: The routes are computed independently from the source to each destination, using the shortest path; the paths are then merged into a single tree. There are several exact algorithms available to compute the shortest path [2]. If the link labels used in the routing are the link delays, this approach will yield the minimum delay routing from the source to each of the destinations. This algorithm is used in the Multicast OSPF routing protocol [3] (where the link labels can be set arbitrarily). Note that this algorithm computes the routes from a topological point of view; to accommodate for the stream bandwidth requirement, the links which do not have enough free bandwidth to support it must be pruned from the network topology prior to applying the algorithm. Moreover, it can only route one stream at a time; for multiple multicast sessions, it has to be applied sequentially in some order (and the routes are order-dependent). For the evaluation, we will consider the shortest path algorithm with the link delays as labels, which will be denoted by **SP/delay**, and with the link costs as labels, which will be denoted by **SP/cost**.

Minimum Cost Algorithms: The routing is done so as to minimize the sum of the labels of the links used; if the link labels are set to the link costs, this approach will lead to the minimum cost multicast routing. This is the well-known problem of finding minimum-cost Steiner trees in graphs, which is known to be NP-complete. Many exact solutions (with exponential worst-case run times) and heuristics have been proposed to address this problem; see [4] for a comprehensive survey of the field. The most important heuristic solutions are the algorithms by Kou, Markowsky and Berman [5] (which we will refer to as KMB), Rayward-Smith [6] (which we will refer to as RS) and Takahashi and Matsuyama [7] (TM). As with the shortest-path algorithms, these heuristics can only route one stream at a time, and the network topology must be pruned of the links that do not have enough free capacity. An additional problem is that these heuristics were designed for <u>undirected</u> graphs; however, when the links with not enough free bandwidth are pruned from the network, the resulting graph is in general directed, and the original heuristics cannot be applied. We have modified the KMB heuristic to take into account the direction of the links [8]; in the evaluation, we will denote this algorithm by **KMB**.

Optimum Multicast Routing Algorithm: In [1, 9], an integer programming formulation for the optimum multicast routing problem and a solution method based on the decomposition principle is presented. This algorithm is able to directly compute the optimum routes for a session composed of multiple multicasts, taking into account the requirements of bandwidth and latency for each individual multicast, and using an arbitrary combination of cost and delay as the objective function. Since the worst-case run time of this algorithm is exponential, it is used essentially as a benchmark in this evaluation. We use three variations of this algorithm in the evaluation: **Optimum/cost**, which minimizes the cost, **Optimum/cost/delay**, which minimizes the cost as the primary objective, and the delay as a secondary objective, and **Optimum/delay/cost**, which minimizes the delay as the primary objective, with the cost as a secondary objective.

4 Previous Work in Evaluation of Multicast Routing Algorithms

The algorithms referenced above have been studied in the context of multicast routing by a number of researchers. Kumar and Jaffe [10] considered the problem of routing single multicasts in a network with undirected links. They compared a number of minimum delay and minimum cost algorithms when the link cost and delay weights are the same and derived analytical bounds in cost/delay under that assumption. Waxman [11] and later Doar and Leslie [12] studied the problem of dynamically adding and removing destinations on a multicast that had been already routed, also in the context of a single multicast in an undirected network. Waxman studied the RS and KMB heuristics, while Doar and Leslie compared shortest path with KMB for re-routing the multicast. Chow [13] studied the case of single multicasts on directed graphs, and proposed a minimum cost heuristic algorithm for the case where not all nodes are capable of multicasting. Ammar et al [14] studied the minimum cost routing under a variety of additional constraints using a non-linear integer programming formulation.

Kompella et al [15] studied the case of minimum-cost routing under maximum delay constraints. They assumed independent weights for costs and delays, undirected links, and give a variation of the KMB heuristic to compute a suboptimal solution taking into account the delay constraints. If the link delays come from a bounded set, their algorithm is polynomial; otherwise, it is exponential on the number of bits used to represent the delays. The results are shown to be close to the optimum, which is found by exhaustive search.

Jiang [16] considered the problem of establishing videoconferences; a video-conference with P participants is established as P multicasts, from each conferee to all other members of the conference. He proposed variations to the KMB and RS algorithms to take into account the link bandwidths. Links are undirected, with the bandwidth available in any direction.

In summary, most of the previous work in this field has focused on proposing and evaluating minimum-cost heuristic algorithms for routing a single multicast, usually on undirected graphs generated at random. Most of these heuristics are not really appropriate for use in a realistic scenario where multiple multicasts coexist in the network; since multicasts are unidirectional, the algorithm *cannot* assume that links are undirected (i.e., that bandwidth is available in both directions). None of the previous work has evaluated the performance of the routing algorithm in a realistic scenario, where streams are established, exist for a period of time, and terminate. In such a scenario, the blocking probability (i.e., the probability that a session arrives and cannot be routed) is the basic performance measure, instead of just cost or delay.

5 Scenarios for the Evaluation

In this section, we define the network and traffic scenarios used in the evaluation, as well as the performance measures. For the *network*, we consider:

- The link capacities are all equal. The link costs are also all equal; this would apply, for example, in a private network with leased lines, where the cost of operating the network is not a function of the traffic on individual links.
- The criterion to select network topologies for the evaluation is that they should be general, but have the properties that one would be likely to find in an existing network. Deployed networks are characterized by: (i) they are usually two-connected[1] for reliability purposes; and (ii) links are more likely to exist between nodes that are close together than between nodes that are far apart. Waxman [11] proposed an algorithm to generate random topologies that takes into account (ii); however, property (i) is more important, as the evaluation in section 6 will show. The following network topologies are considered:
  - A simplified version of the NSFNet T3 backbone. In this case, we set the link delays to the propagation delays between the nodes.
  - Network topologies generated at random, with a fixed number of nodes and links. We restrict ourselves to both strongly-connected[2] and two-connected topologies. The nodes are placed at random in a plane, and the link delay is set to the distance between the two nodes it connects. For the numerical evaluation, we have generated the node positions in a rectangle whose horizontal dimension is 15 ms and vertical dimension is 10 ms; this would correspond to an area roughly equivalent to that of the United States. The algorithms used to generate the random topologies are discussed in detail in [8].

[1] There are at least two independent paths between every pair of nodes.

[2] There is at least one (directed) path between every pair of nodes.

For the *traffic*, we consider that all multicasts in a session arrive and depart simultaneously. The session arrivals form a Poisson process, with rate λ, and the session duration is exponentially distributed, with rate μ. We assume that sources and destinations are uniformly distributed over the network, and that the set of destinations is fixed for the duration of the session (i.e., no destinations join or leave the multicast while it is in progress). The following kinds of sessions are considered:

- *Single-Multicast Sessions:* Each session is composed of a single multicast, whose number of destinations is chosen at random, uniformly between 1 and n_{max}; the value of n_{max} is chosen depending on the number of nodes in the network under evaluation.
- *Videoconference Sessions:* Each session has P multicasts and corresponds to a videoconference with P participants. P is chosen at random between 2 and 4.

We consider the following two scenarios for the stream bandwidth:

- *Equal Bandwidth:* All streams have the same bandwidth requirement, set at 10% of the link capacity.
- *Discrete Bandwidth Distribution:* The stream bandwidth is chosen at random between the values of 4%, 9%, 18% and 36% of the link capacity, with probabilities of 0.3, 0.3, 0.3 and 0.1.

We assume that blocked sessions are lost, and our primary performance measure is the overall session blocking probability. We are also interested in the blocking probability as a function of the number of destinations of the multicast. Given the traffic characteristics, we define the *network capacity* for a certain blocking probability as the load (λ/μ) for which that blocking probability is reached.

6 Numerical Results

In this section, we present the simulation results. We first investigate the effect of the kind of topology (random/two connected). Next, we present a baseline scenario and characterize its performance. The baseline results are then compared to a number of other scenarios.

6.1 Effect of the Topology

In order to assess the effect of the kind of topology in the evaluation results, we generated at random ten topologies, all with 12 nodes and 15 full-duplex links, which are shown in figure 1. Half of the topologies were completely random (topologies 1 to 5), and the other half was two-connected (topologies 6 to 10). We obtained the blocking probability in each of the networks, for each of the algorithms. We also repeated the same process for a simplified version of the NSFNet T3 backbone (also with 12 nodes and 15 links), shown in figure 2 (the numbers next to the links are the propagation delays in milliseconds). The traffic was composed of single-multicast sessions, with a random number of destinations between 1 and 10. The main observation was that the blocking probability is much higher in the completely random topologies than in the two-connected ones. The results for the Optimum/cost algorithm are shown in figure 3; similar results were observed for the other algorithms.

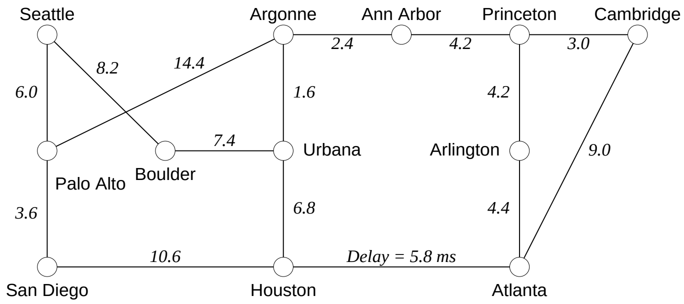

Figure 2: The NSFNet T3 backbone (simplified)

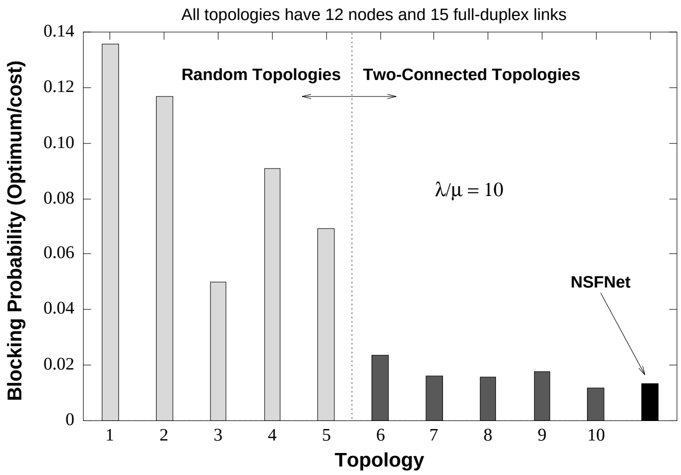

Figure 3: Blocking probability in random topologies and two-connected topologies

These results are explained by the fact that a completely random topology, especially for low node degrees[3], is likely to have a link that, if congested, will "divide" the network into two disconnected parts, leading to a high blocking probability. Further investigation showed that for this kind of topology, there is very little difference in performance between the different algorithms; the blocking probability is essentially determined by the topology. In the sequel, we focus our investigation only into two-connected topologies, both existing and generated at random.

[3]Ratio between the number of links and the number of nodes in a network.

6.2 The Baseline Scenario

In this section, we present a baseline scenario, and characterize its performance. We then discuss how the results change as the scenario changes from the baseline. This baseline scenario corresponds to two-connected topologies, with 12 nodes and 15 links; traffic is composed of single-multicast sessions, with a random number of destinations between 1 and 10. Each stream requires 10% of the link bandwidth and there is no latency constraint.

In figure 4 we plot the overall blocking probability, averaged over a large number of two-connected topologies, as a function of the offered load (λ/μ), for all algorithms. The figure indicates that, as expected, the blocking probability for the *cost-based algorithms* (Optimum/cost, Optimum/cost/delay, KMB) is lower than for the *delay-based algorithms* (Optimum/delay, SP/delay) and the SP/cost algorithm. At 1% blocking, the network capacity for this traffic scenario is about 17 for the cost-based algorithms, and 13 for the delay-based algorithms. At 10% blocking, the values are 25 and 22 respectively. We repeated the same runs for the simplified version of the NSFNet T3 backbone shown in figure 2 and found similar results (with a more marked difference between the two groups of algorithms).

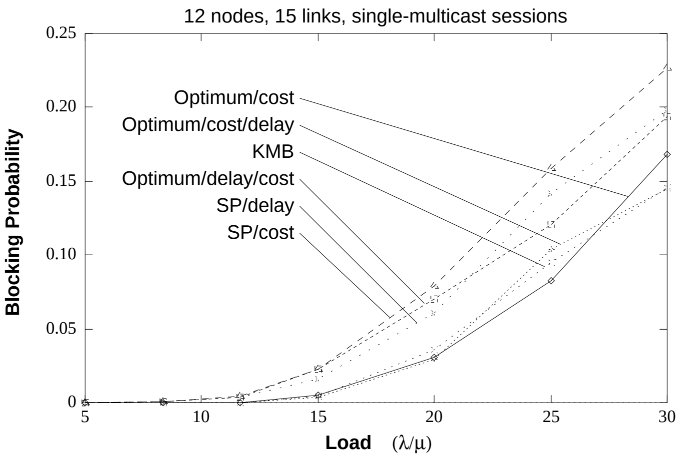

Figure 4: Simulation results for two-connected topologies, single-multicast sessions

When multicasts with different numbers of destinations co-exist in a network, we expect that the blocking probabil-

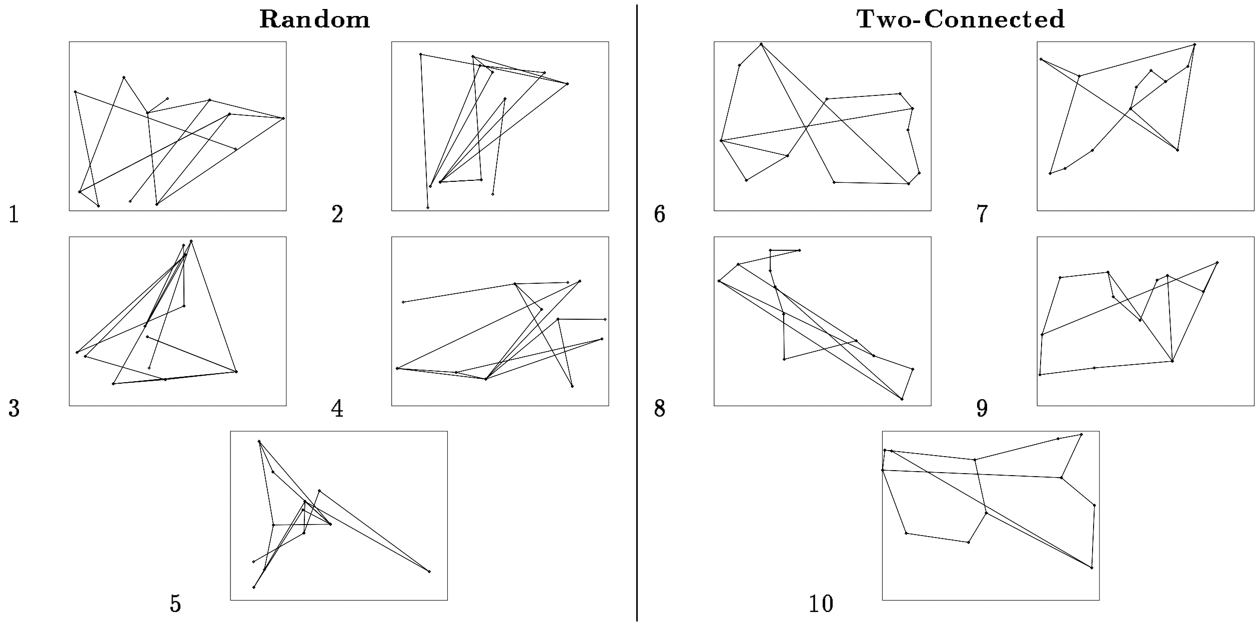

Figure 1: Random network topologies used in the evaluation

ity be higher for the multicasts with higher number of destinations. To investigate this aspect, we plot in figure 5 the blocking probability as a function of the number of destinations, at different load values, for the Optimum/delay/cost algorithm. Figure 5 shows that, for low loads, the blocking probability is a weak function of the number of destinations. Even at high loads, the ratio between the blocking probability for 10-destination multicasts and unicasts (one destination) is in the range of 2-3. The plots for the other algorithms are similar.

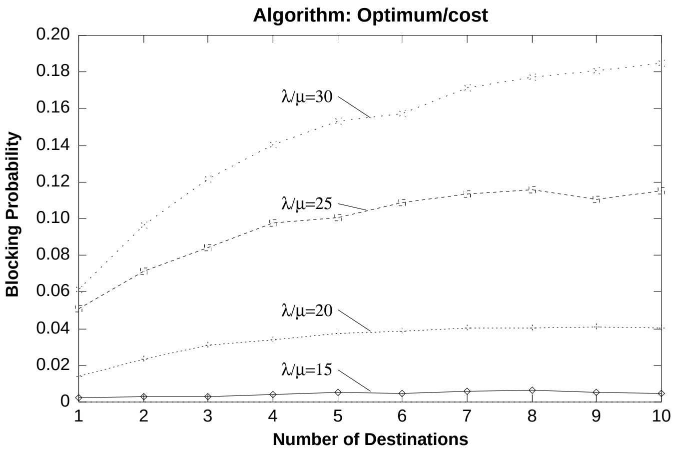

Figure 5: Blocking probability as a function of the number of destinations in the multicast, algorithm Optimum/cost

6.3 Introducing a Delay Constraint

The objective of this section is to determine the effect of a delay constraint on the blocking probability of the various algorithms. We chose the NSFNet T3 backbone, under single multicast sessions, with the stream bandwidth fixed at 10%. The diameter[4] of this network in delay is 28 ms (shortest path from Seattle to Arlington).

A delay constraint comes from the need to support interactive communications; it is a well known fact that human beings can tolerate up to $\approx 200 - 250$ ms of delay in a two-way conversation. In a communications system using compressed video, there are delays both in the encoder and in the decoder that can be as high as 100 ms, as well as delays in the local networks to which the video stations are attached [17]. Therefore, a reasonable constraint for the wide-area component of the delay would be 40 ms. In figure 6 we plot the blocking probability as a function of the load under the 40 ms constraint. Table 1 shows the fraction of the routes in the unconstrained case which would not satisfy the 40 ms latency requirement.

Figure 6 indicates that, as hinted by the numbers in table 1, the delay-based algorithms are essentially unaffected by this constraint, because delay is their primary objective. The optimum algorithms take the delay constraint into account when computing the routes; they are also not affected because they can usually identify alternate routes satisfying the constraint. In the case of the KMB algo-

[4] Maximum shortest path over all pairs of nodes

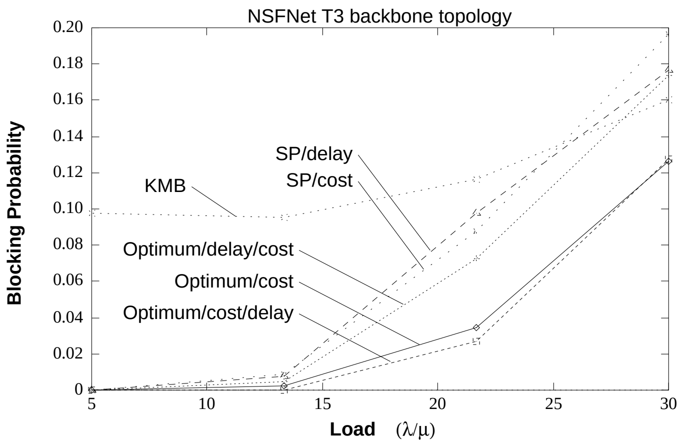

Figure 6: Simulation results for the NSFNet T3 backbone, under a 40 ms delay constraint

Table 1: Fraction of routes that do not satisfy the 40 ms latency in the unconstrained case

Algorithm	Fraction
Optimum/cost	16%
KMB	9%
Optimum/cost/delay	2.6%
SP/cost	≈ 0%
Optimum/delay/cost	≈ 0%
SP/delay	≈ 0%

rithm, however, there is a large effect since its objective is cost, not delay. Even at low loads, the blocking is high because the algorithm is forced to reject calls whose routes would exceed the latency constraint. The bigger the number of destinations in the multicast, the larger the effect: while at low load the KMB algorithm is capable of accommodating almost all single-destination multicasts under the 40 ms constraint, the blocking probability is over 20% for 10-destination multicasts. As the constraint is decreased from 40 ms, the blocking probability "plateau" observed at low loads in the KMB algorithm will significantly increase (for example, a constraint of 30 ms would move it to about 30%). As the constraint is made tighter, the optimum algorithms will tend to use the shortest paths in delay, regardless of their objective functions. Of course, if the constraint is set to a value lower than the network diameter, the blocking will be high for all algorithms. More detailed results can be found in [8].

6.4 Other Scenarios

In this section, we consider other variations to the baseline scenario and comment on the results. The variations pertain to: (i) the stream bandwidth distribution; (ii) the network size; and (iii) multiple-multicast sessions.

Instead of the equal bandwidth streams of the baseline scenario, we considered streams with a discrete bandwidth distribution, as described in section 5. We observed similar results as the baseline case, but the overall performance is now lower.

We considered two-connected random topologies with 6 nodes and 8 links (approximately the same node degree as the baseline case), and with 6 nodes and 12 links (almost completely-connected). The number of destinations was chosen at random between 1 and 4. For the topologies with 6 nodes and 8 links, the results were similar to the baseline case, with a distinct difference in blocking probability between the cost-based and the delay-based algorithms. For the topologies with 6 nodes and 12 links, however, there was little difference between the performance of the cost-based and delay-based algorithms.

Finally, we considered multiple-multicast (videoconferencing) sessions, under a number of network scenarios (6-node networks and 12-node networks), with the stream bandwidth fixed at 10% of the link bandwidth, as in the baseline scenario. The main observation, valid for all scenarios, is that there is very little difference in blocking probability between all the algorithms, although the cost-based algorithms still have a small advantage. This is due to the fact that a session is blocked if any of its components is blocked; therefore, the blocking probability is a much "coarser" measure of performance in multiple-multicast sessions than for single-multicast sessions. Moreover, since in the cases evaluated the number of multicasts in the session is small and each stream requests a small fraction of the link bandwidth (i.e., 10%), the problem is in most cases naturally decomposable (i.e., there is no coupling between the routes of the streams in the session) and there should be little difference between the optimum solution (which takes all streams into account simultaneously when computing routes) and the solution found by the heuristic algorithms (which considers each stream in isolation). Note that if the stream bandwidth is a significant fraction of the link bandwidth, this is not true anymore, as shown in [9], and there will be a large difference between the optimum and the heuristics. Another important observation is that, except at very low loads, the blocking probability is now very dependent on the number of participants in the conference for all the algorithms; for example, in the NSFNet under the Optimum/cost/delay algorithm, the blocking probability at $\lambda/\mu = 10$ is about 5% for conferences with 2 participants, while it reaches 22% for conferences with 4 participants. Additional results can be found in [8].

7 Conclusions

In this paper, we have compared a number of existing multicast routing algorithms under uniform conditions in realistic scenarios; we determined that cost-based algorithms perform better than delay-based algorithms, as long as there are no latency constraints. The difference in terms of network capacity between the two groups can be anywhere from 1.2 to 2.0 times, depending on the scenario. However, if there is a latency constraint, cost-based algorithms that do not explicitly take it into account have very poor performance.

References

[1] C. Noronha and F. Tobagi, "Optimum Routing of Multicast Audio and Video Streams in Communications Networks," *Technical Report CSL-TR-94-618*, Computer Systems Laboratory, Stanford University, April 1994.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, New Jersey, 1987.

[3] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Proteon, Inc., March 1994.

[4] F.K. Hwang and D.S. Richards, "Steiner Tree Problems," *Networks*, Jan. 1992, vol.22, no.1, p. 55-89.

[5] L. Kou, G. Markowsky and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica* 15, pp 141-5, 1981.

[6] V.J. Rayward-Smith and A. Clare, "On Finding Steiner Vertices," *Networks*, Fall 1986, vol. 16, no.3, pp. 283-94.

[7] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math. Japonica* 6, 1980, pp. 573-7.

[8] C. Noronha and F. Tobagi, "Evaluation of Multicast Routing Algorithms for Multimedia Streams," *Technical Report CSL-TR-94-619*, Computer Systems Laboratory, Stanford University, April 1994.

[9] C. Noronha and F. Tobagi, "Optimum Routing of Multicast Streams," *IEEE INFOCOM 94*, Toronto, Canada, June 1994, pp 865-73.

[10] K. Bharath-Kumar and J.M. Jaffe, "Routing to Multiple Destinations in Computer Networks," *IEEE Trans. on Comm.*, vol. COM-31, no. 3, March 1983, pp 343-51.

[11] B.M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, (Dec. 1988) vol.6, no.9, p. 1617-22.

[12] M. Doar and I. Leslie, "How Bad is Naïve Multicast Routing?" *IEEE INFOCOM '93*, San Francisco, CA, USA, pp 82-9.

[13] C.-H. Chow, "On multicast path finding algorithms," *IEEE INFOCOM '91*, Bal Harbour, FL, USA, 7-11 April 1991, pp. 1274-83.

[14] M.H. Ammar et al, "Routing Multipoint Connections Using Virtual Paths in an ATM Network," *IEEE INFOCOM '93*, San Francisco, CA, USA, pp 98-105.

[15] V.P. Kompella et al. "Multicast Routing for Multimedia Communication," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, June 1993, pp 286-92.

[16] X. Jiang, "Routing broadband multicast streams," *Computer Communications* (Jan.-Feb. 1992) vol.15, no.1, p. 45-51.

[17] İ. Dalgıç and F. Tobagi, "Evaluation of 10Base-T and 100Base-T Ethernets Carrying Video, Audio and Data Traffic," *IEEE INFOCOM 94*, Toronto, Canada, June 1994, pp 1094-1102.