

A Comparison of Two Practical Multicast Routing Schemes

Ken Calvert Ramesh Madhavan Ellen W. Zegura

GIT-CC-94/25

February 1994

Abstract

Designing an interdomain multicast routing scheme that makes efficient use of network resources while delivering good performance to applications is a significant challenge. A variety of schemes have been proposed, but little has been done to compare the schemes systematically over a rich set scenarios. In this work we develop a framework to do systematic evaluation of multicast routing schemes, and apply it to two practical schemes: Distance Vector Multicast Routing Protocol and Core Based Trees. We conclude that Core Based Trees has the *potential* to make more efficient use of resources, with modest performance penalty. However, this requires mechanisms to choose good cores. We suggest a heuristic for evaluating the goodness of a core and moving towards a good core.

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280

1 Introduction

There is little question that emerging networking applications will require multicast capability. From video conferencing to replicated database access, the need exists to transmit from one or more sources to multiple destinations. Supporting multicast in local area networks is relatively uncomplicated, particularly with broadcast technologies like Ethernet. Supporting interdomain multicasting is a more significant challenge, requiring a solution which deals with issues such as scaling along many dimensions (e.g., network size, number of groups), incomplete or inaccurate information, and diverse application characteristics.

Routing schemes can be divided into two categories, connection-oriented and connection-less. Connection oriented schemes reserve resources during connection setup, make use of the resources during transmission, and release resources during connection teardown. Connection-less (or datagram) schemes do not reserve resources; rather the network makes a best effort to deliver data.

A variety of schemes have been proposed for connectionless multicast routing, including Distance Vector Multicast Routing Protocol (DVMRP) [6], Link State Multicast Routing Protocol [5], and Core Based Trees (CBT) [1]. The literature describing these schemes emphasizes the ease of implementation within the existing Internet structure, certainly an important issue. However, to date we have not seen a systematic comparison of these schemes for realistic applications. Without such a comparison, it is unclear which scheme, if any, is best suited to the task. The goal of this work is to explore the relative merits of two of these schemes—DVMRP and CBT—by measuring quantities of practical concern under realistic operational scenarios.

Several issues must be addressed. First, what quantities are of practical interest? Multicast approaches can be compared along a number of dimensions, including complexity of implementation and administration, (e.g the amount of state information stored by each router), efficiency of resource usage [3, 10], (e.g. the amount of bandwidth required to deliver a multicast packet), performance delivered to the application, (e.g. the maximum delay from a source to a destination), robustness in case of node or link failure, and scalability.

A second issue concerns the choice of realistic operational scenarios. Key to the success of any Internet multicast routing protocol is the ability to deal with scale along any dimension, including network size, number of sources, number of members, number of groups, group characteristics, etc. We have designed an evaluation framework which constructs realistic, large graphs and allows exploration of a wide range of application characteristics.

We conclude that Core Based Trees has the *potential* to make more efficient use of resources, at a modest cost in terms of performance (delay). Realizing this potential, however, requires mechanisms to choose good cores. Identifying good cores is a nontrivial problem which must take into consideration characteristics of the network and the application. We suggest a heuristic for evaluating the goodness of a core and moving towards a good core.

The remainder of the paper is organized as follows. In the next section we describe the two algorithms, DVMRP and CBT, and briefly discuss some of their qualitative advantages and disadvantages. In Section 3 we outline our approach for evaluating the efficiency and performance of the multicast routing schemes. Section 4 describes the results, with emphasis on performance as various dimensions grow. As Section 4 reveals, the choice of the core in the CBT scheme can significantly affect the performance. In Section 5 we describe a heuristic

for moving the core to a good location. Finally, Section 6 contains conclusions and directions for future work.

2 Algorithms

In this section we briefly describe the operation of the two multicast routing schemes. The reader is referred to the original descriptions [1, 6] for more detail.

2.1 DVMRP

DVMRP [6, 9] is based on *reverse path forwarding*, a method of implementing broadcast in point-to-point networks. DVMRP routers forward multicast packets based on their *source*: upon receiving a packet destined for a multicast address, a router checks whether it was received from the direction of the shortest path to its source. If so, it forwards the packet on outgoing links toward other group members. The “distance vector” part of the name refers to the mechanism used by routers to ensure that only one copy of each packet is forwarded over each network; only the router closest to the source forwards the packet.

The advantages of DVMRP are its low delay characteristics (it performs optimally with respect to this metric, since each packet follows the shortest path from the source to each destination) and its relatively simple implementation. It is the scheme currently used for Internet multicasting over the MBONE [2]. However, DVMRP requires that each router store information for each (source, group) pair; this information determines whether a packet from a particular source is forwarded on a particular link. Moreover, in DVMRP, neighboring routers must periodically take explicit action in order to “turn off” forwarding. Thus, routers *not* involved with a particular multicast group nevertheless incur overhead associated with maintaining the tree for the group.

2.2 CBT

Core Based Trees (CBT) [1] is a recently proposed alternative to DVMRP. The main goals of CBT are scalability and independence of underlying unicast routing mechanism. The key idea behind the scheme is to designate a *core* router for each multicast group. Packets sent to the group travel along the shortest path tree rooted at the core which spans all group members. Sources send to the group by addressing (unicast) packets to the core address of the group. When such a packet encounters a CBT-capable router that is part of the shortest path tree for the group, the packet thereafter follows the tree to the members. Like DVMRP, CBT uses IGMP [4] to deal with hosts joining and leaving groups.

Each CBT-capable router stores information only for multicast groups to which it needs to route packets (i.e., it is in the shortest path tree from the core to some group member). Scalability is enhanced because router storage requirements grow as the number of active groups, rather than the number of active groups times the number of sources. Also, routers that are not involved in a group need not even know of the existence of the group, and extending the tree incurs a low message overhead. The CBT scheme is independent of the underlying unicast routing approach because it requires only a method of forwarding (unicast)

packets along the shortest path toward an individual destination; multicast packets travel between routers the same way as regular unicast packets.

The disadvantages of CBT are its potentially poorer performance (in terms of delay) compared with DVMRP, the potential for congestion in some applications due to concentration of traffic from many sources on the same links, and vulnerability to core failure. The latter two can be ameliorated somewhat by assigning multiple cores per group [1]. Our evaluation considers only single cores, however.

2.3 Summary

DVMRP can be expected to be superior to CBT in terms of delay, since a shortest path tree rooted at each source is optimal with respect to that measure. CBT, on the other hand, is designed to scale better due to reduced storage requirements and control overhead. Compared to DVMRP, CBT should also offer a reduction in the bandwidth required to transmit packets, because multiple shortest path trees do not necessarily perform well with respect to bandwidth consumption. (Optimizing bandwidth requires solving the NP-complete Steiner tree problem [8].) In the remainder of the paper we attempt to quantify these differences in bandwidth and delay, and to characterize, in a general way, their sensitivity to a variety of parameters.

3 Evaluation Methodology

3.1 Network Model

We model the network as an undirected graph in which nodes represent routers and edges represent links connecting routers.¹ A given multicast application is characterized by a subset of the graph nodes designated as group members (more precisely, the nodes represent designated routers for hosts who are the actual group members), and a (possibly disjoint) subset of nodes designated as sources. For a given set of members and sources, the multicast algorithm defines a subgraph containing a path from each source to each member. This subgraph determines both the efficiency and the performance of the algorithm.

Each of the algorithms makes use of trees rooted at one or more nodes. These trees are derived from shortest path trees provided by the underlying internetwork (unicast) routing protocol (whatever it may be). Our model embodies several assumptions about these underlying shortest path trees. First, they are assumed to be stable and globally consistent, that is if m is a descendent of n in the shortest path tree rooted at r , then the path from m to n in the tree rooted at m is the same as in the tree rooted at r . Second, shortest paths are symmetric: the shortest path from m to n is the same as the shortest path from n to m (this assumption is already implicit in the use of an undirected graph). Finally, the triangle equality holds: the shortest path from m to n is no longer than the concatenation of the shortest path from m to r with the shortest path from r to n . Note that the multicast algorithms themselves may not necessarily require all of these properties, only our model, which is intended for evaluation of the *data transfer* characteristics of the algorithms.

Our graphs have a hierarchical structure intended to reflect the structure of the real Internet. They consist of 20 or 30 interconnected “neighborhoods”, with 20 or 30 nodes per neighborhood. The neighborhoods represent routing domains or autonomous systems, in

¹This is a simplification: an edge in the graph actually represents a *network*, which may connect more than two nodes.

which groups of routers are connected via local networks and other links. Graphs with similar structure have been used by others in evaluating multicast routing schemes [7].

The neighborhoods are constructed as “semi-geometric” random graphs. To create a such graph of N nodes, each node is assigned random coordinates in the unit square. An edge is placed between two nodes with probability $p_\alpha(d)$, where d is the Euclidean distance between the nodes, and α is a constant used to vary the degree of connectivity. We used a function p_α defined as follows:

$$p_\alpha(d) = \begin{cases} \alpha & \text{if } d \leq 0.3 \\ \alpha(\sqrt{2} - d)/(\sqrt{2} - 0.3) & \text{if } d > 0.3 \end{cases}$$

Thus nodes have a probability of being connected that is fixed below a certain distance threshold, and then decreases linearly with the distance between them. If the constructed graph is not connected it is discarded and the process iterates.

To construct a hierarchical graph, the following method is used. First a semi-geometric graph of N nodes is constructed; this graph determines the top-level structure of the full graph. Each node of the top-level graph is then replaced with a “neighborhood” graph of N nodes. The edges incident on a node in the top-level graph are connected sequentially to the node in the replacing neighborhood that has the lowest degree greater than one. (This ensures that the connection process preserves leaves). The replacement of nodes in the current graph can be iterated to create graphs with any number of levels. We ran our experiments on two-level graphs with 400 and 900 nodes. An example of such a graph is shown in Figure 1.

3.2 Efficiency and Performance Measures

We measure the efficiency of a multicast algorithm by its bandwidth requirement, defined as the number of packet-hops required to deliver a packet from every source to every group member.² For example, for DVMRP, the bandwidth required for a given set of sources and members is the number of edges in the shortest path tree from each source to all members, summed over all sources. With CBT, this quantity differs slightly from the number of edges in the shortest-path tree rooted at the core, because in general only a subset of nodes are CBT-capable. For example, if a packet is forwarded from r to both m and n via a non-CBT node x , the cost is four packet hops, because the same message must be transmitted twice over the link from r to x : once addressed to m and once to n .

As an approximate measure of the *delay* experienced by individual packets under each multicast scheme, we consider the number of edges in the path between a source and a member. The *average delay* is defined to be the path length from source to group member, averaged over all source-member pairs. The *maximum delay* is defined to be the maximum over the same set of path lengths.

3.3 Experiment Design

We generated 10 graphs for each of four combinations of size and connectivity: 400-node graphs with connectivity parameters of 0.1, 0.2, and 0.3 (average node degree for each combination is 2.48, 3.35, and 4.78, respectively) and 900-node graphs with connectivity parameter 0.16 (average degree 3.92). For each graph, about half the nodes were designated at random as CBT-capable routers.

²While our methods are valid when edges can have differing costs, the results reported here assume that all edges have equal cost. This assumption seems appropriate for modeling an *inter-domain* multicast protocol.

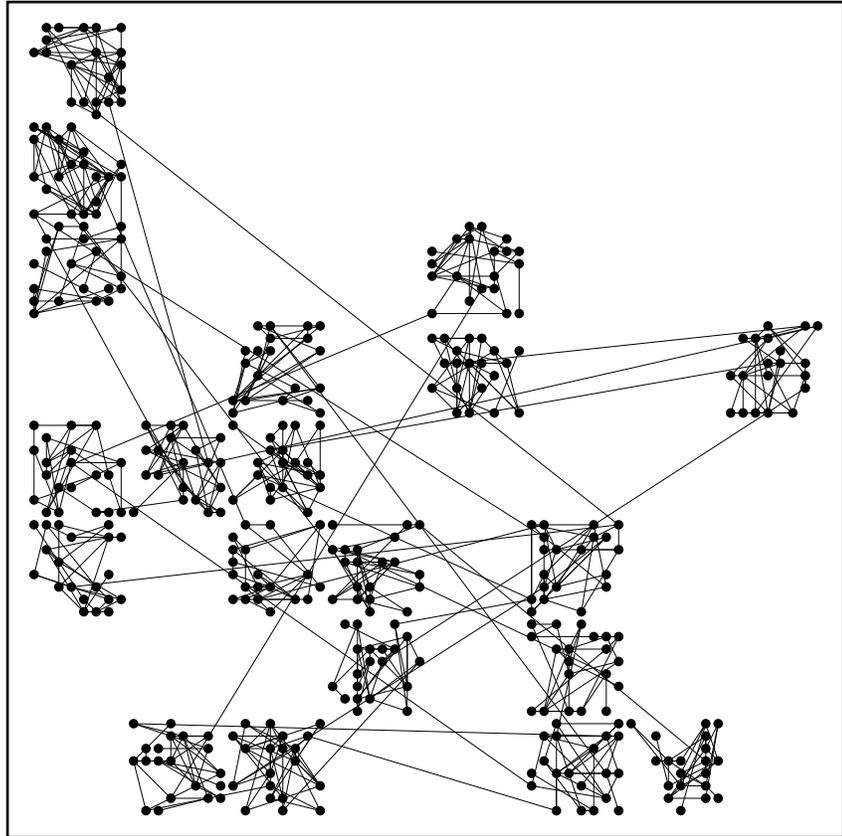


Figure 1: Example 400 node graph

We then selected four scenarios reflecting different numbers of sources and distribution of group membership. (These scenarios are described in detail in the next section.) For each configuration of sources and group members, we tried each CBT-capable node as core, and computed the following quantities:

- The *best* average delay value, the *best* maximum delay value, and the *best* bandwidth value among all possible cores (i.e. all CBT-capable nodes).
- The *worst* average delay value, the *worst* maximum delay value, and the *worst* bandwidth value among all cores.
- The *average* of the average delay, maximum delay, and bandwidth, over all possible cores.
- The average delay, maximum delay, and bandwidth for DVMRP.

Note that for a given graph and scenario, the same core was not necessarily the best with respect to all three measures, i.e., no single core would necessarily produce all three “best” values, and similarly for the “worst” values.

For each graph and scenario, we computed the ratio of the best, worst and average core values to the DVMRP values for each metric. Finally, we took the average of these ratios over the 10 graphs and computed 95% confidence intervals.

4 Quantitative Results

We have used the evaluation approach of the previous section to extensively analyze the performance of DVMRP and CBT. We have designed four scenarios with varying characteristics so as to compare the schemes across the spectrum of possible applications. We begin by presenting results from each of the four scenarios, then consider the impact of graph connectivity and size.

4.1 Scenarios

Scenario 1 consists of members that are all from the same local neighborhood, and a number of distributed sources. We vary the number of sources, and compare the performance of CBT to DVMRP. As an example, this scenario matches the characteristics of a local database which has been distributed over several hosts, where the sources represent remote accesses to the database.

Figure 2 shows the ratio of bandwidth for CBT to bandwidth for DVMRP for the best CBT core, the worst and the average. Figure 3 shows the same type of results for maximum delay. The bandwidth ratio for the best core is about .75 for any number of sources, demonstrating the ability of CBT to offer a savings in bandwidth over DVMRP. In addition, the best core performs almost as well as DVMRP for maximum delay. Even a random choice of core has bandwidth ratio of about 1.25 and maximum delay ratio of about 1.5. The worst possible core has a bandwidth ratio of approximately two and a maximum delay ratio of about 2.5.

Inspection of more detailed output from the experiments indicates that, except when there are a small number of sources (1 or 2), the core that minimizes bandwidth and average delay

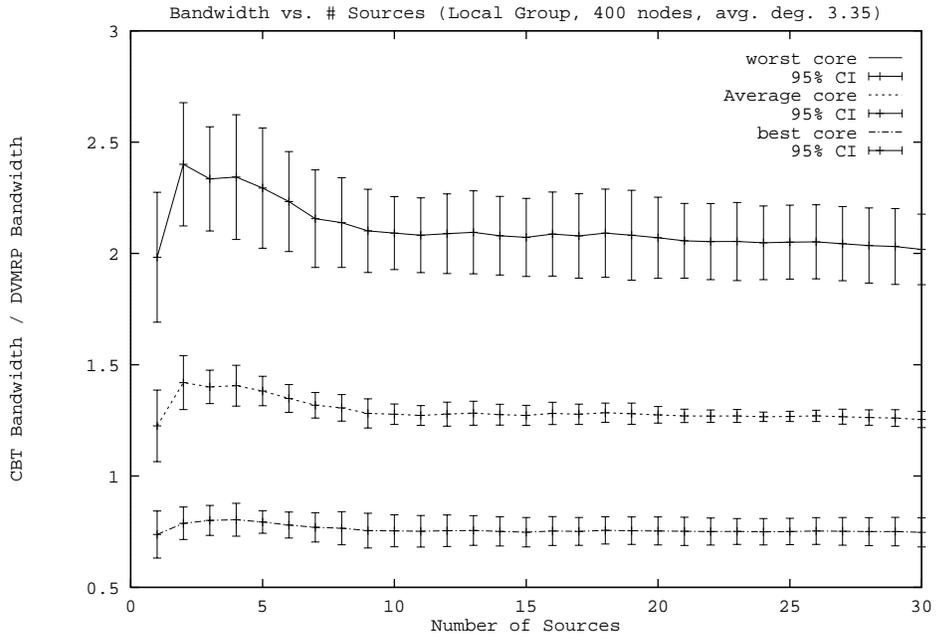


Figure 2: Scenario 1 - Bandwidth

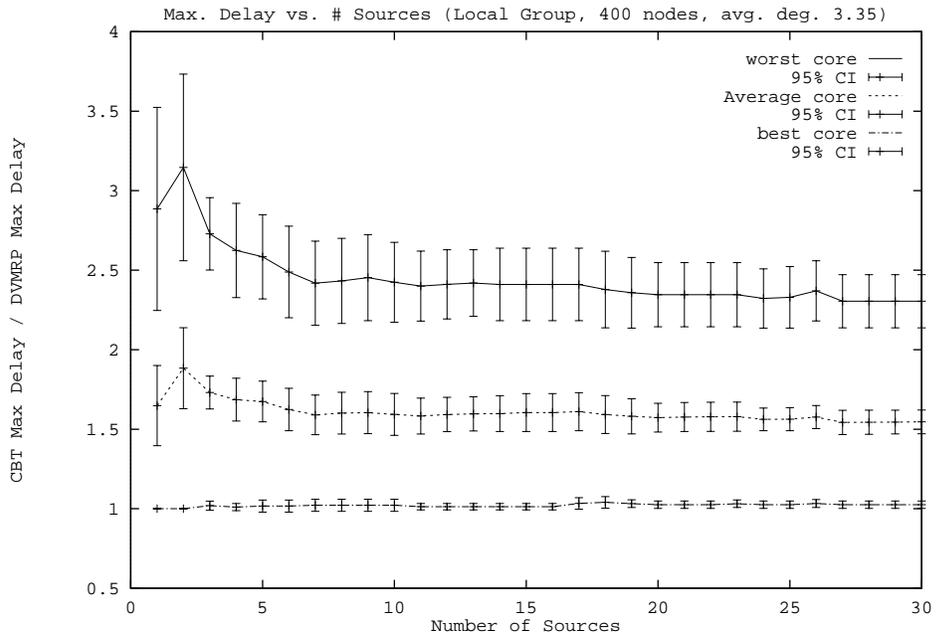


Figure 3: Scenario 1 - Maximum Delay

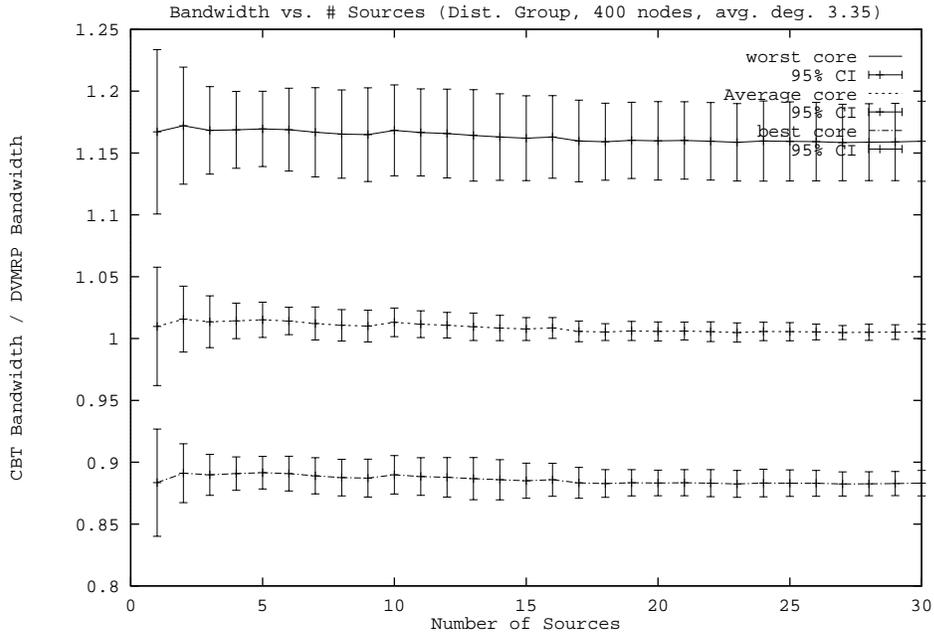


Figure 4: Scenario 2 - Bandwidth

comes from the neighborhood where the members are located. Based on this, we make the following observation:

For a local group with fixed sources, a core chosen from within the group with no knowledge of the location of the sources can provide near-optimal bandwidth and average delay.

We can also reach some conclusions about the robustness of the best core in the presence of volatility in the sources. This reflects a practical scenario, such as distributed database access, in which the members are relatively constant, but the sources may be highly dynamic. It would be nice if a good core would remain good as the identity and number of sources change. The experiments reveal that the identity of the best core (by any of the metrics) tends to remain the same when a moderate number of sources are added. Based on this, we make the following observation:

For a local group, a core chosen to be good for a particular set of sources should not require relocation under moderate changes to the number of sources.

Scenario 2 is similar to Scenario 1 except that the group members are distributed and there are considerably more members (approximately 100 in the 400 node graph). For a small number of sources, this can model a video broadcast of a few channels. For a larger number of sources, this can model accesses to a large, distributed database.

Figures 4 and 5 show the bandwidth and maximum delay ratios, respectively. The best core for this scenario again improves on DVMRP for bandwidth, although not as significantly as in Scenario 1. The average and worst core ratios for bandwidth are better than in Scenario

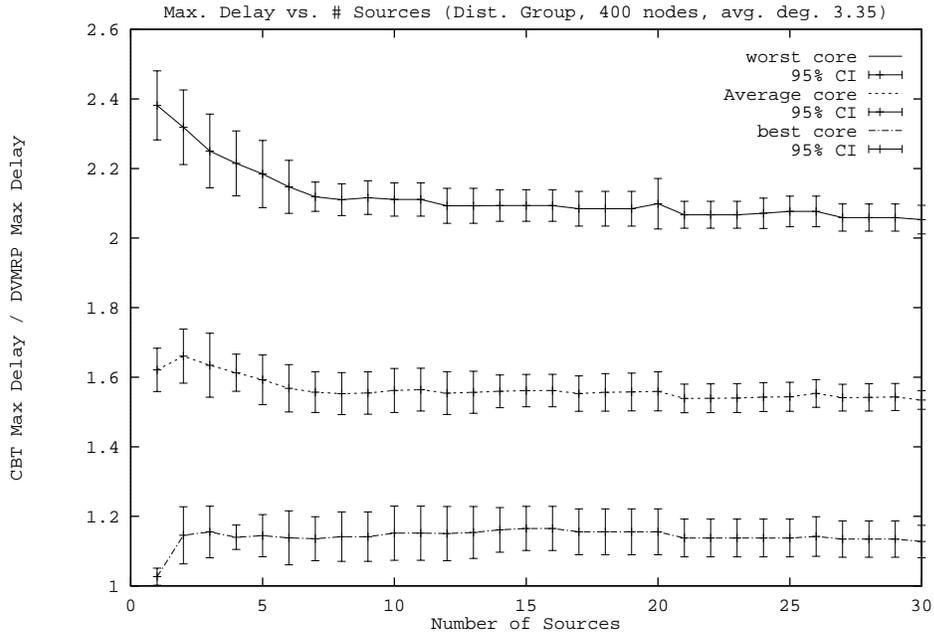


Figure 5: Scenario 2 - Maximum Delay

1. There is a slight increase in maximum delay ratio for the best core, although the worst core ratio is better than in Scenario 1. This behavior is due to the distributed locations of both members and sources; when both are distributed, any core can do reasonably well. Based on this, we make the following observation:

For a large, distributed group, choosing a core at random or from a list of topologically “near-center” routers will give good performance for any number of sources.

As for Scenario 1, we can reach conclusions about robustness of the core in the presence of volatility by looking at more detailed experimental results. For Scenario 2, we found that the best core for bandwidth was almost *independent* of the number of sources. For almost all graphs, a single core had the best bandwidth ratio over the full range of numbers of sources. We can thus make the following observation:

For a large, distributed group, a good core should not require relocation due to changes in the identity or number of sources.

Scenario 3 fixes a single random source which transmits to a distributed group of members. In this scenario we explore a much wider range of group sizes than in Scenario 2. This models applications such as video broadcast or distance learning. Figures 6 and 7 show the bandwidth and maximum delay respectively. When the number of group members is small, there is significant difference between the best and worst cores for bandwidth, however these differences diminish as the number of members increases. For a small group and one source, the choice of a good core for bandwidth must take into account the location of the members and the source. Since there is only one source, the core which minimizes maximum depth is

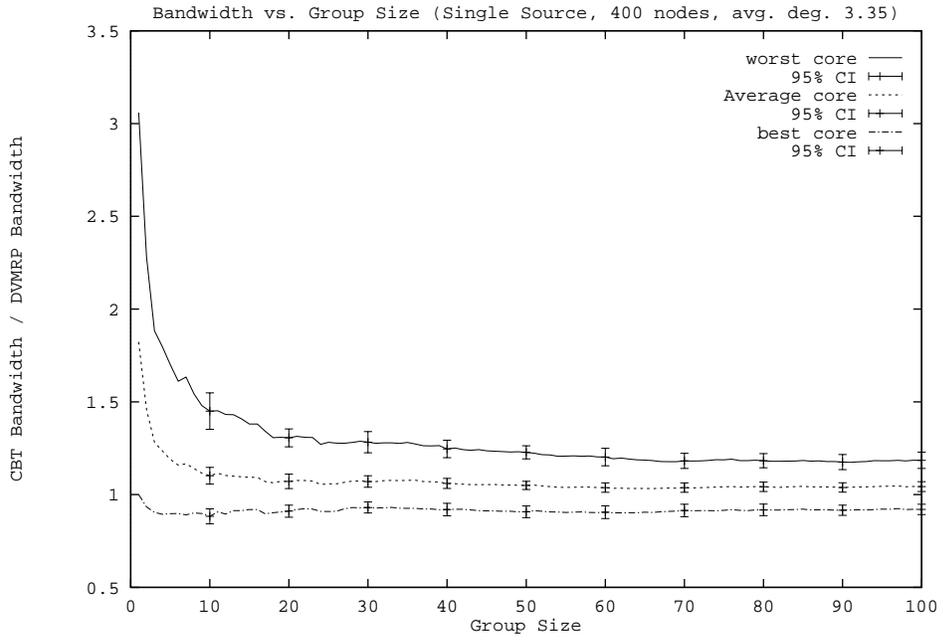


Figure 6: Scenario 3 - Bandwidth

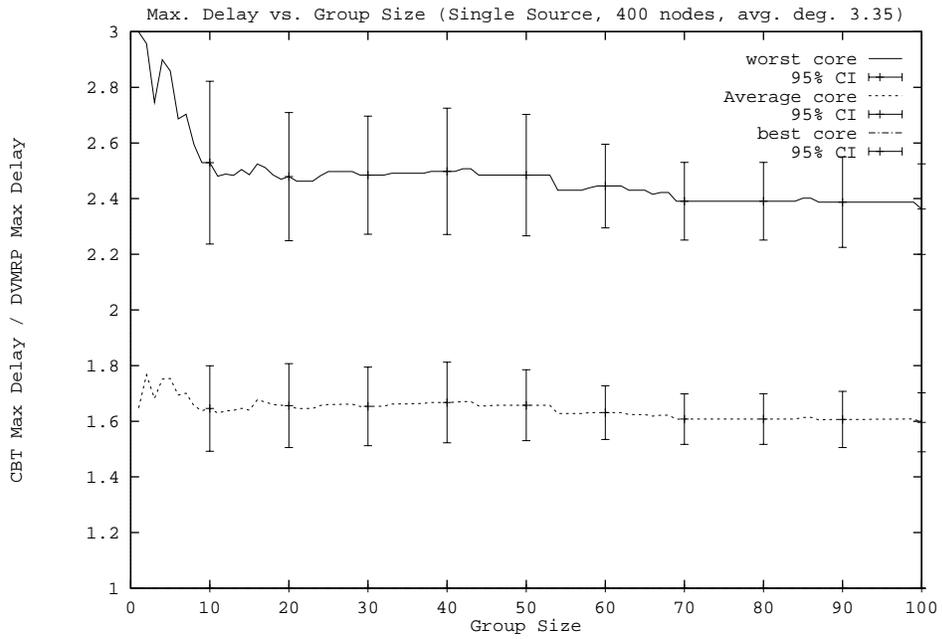


Figure 7: Scenario 3 - Maximum Delay

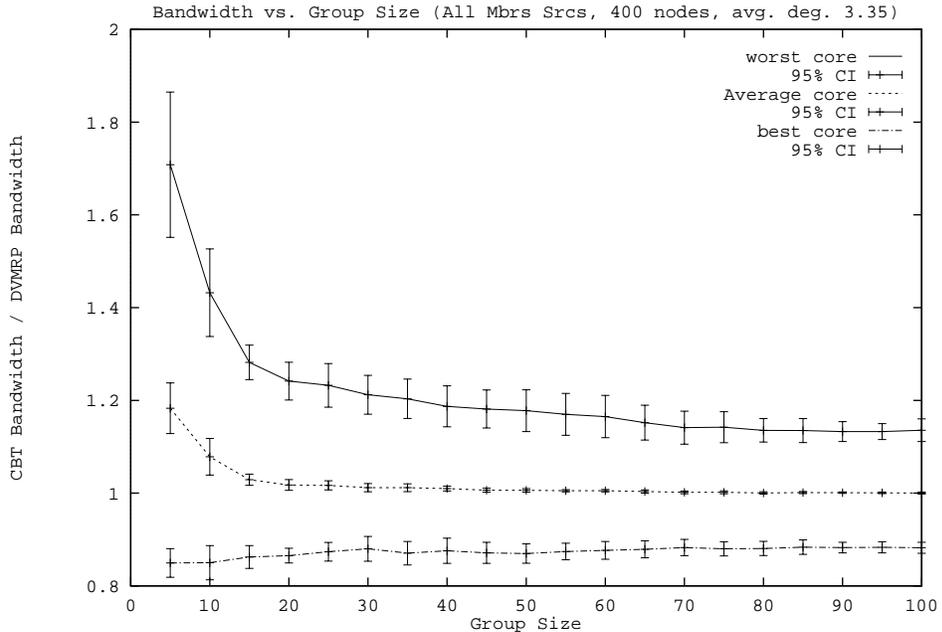


Figure 8: Scenario 4 - Bandwidth

the source itself, matching the performance of DVMRP. Based on this, we make the following observation:

Wide distribution of group members is not enough to make core choice unimportant. For small groups, even if widely distributed, the core must be chosen carefully.

Scenario 4 consists of a distributed group in which all members are also sources. This models applications such as distributed collaboration or video conferencing. Figures 8 and 9 show the bandwidth and maximum delay respectively. These results are consistent with observations for other scenarios with a distributed group. Somewhat surprising is the close agreement between the average core bandwidth and the DVMRP bandwidth. Since all sources are also members, there is no “off-tree” bandwidth, regardless of the location chosen for the core. The CBT bandwidth is simply m times the number of edges in the shortest path tree rooted at the core, where m is the number of members. The DVMRP bandwidth is the sum over all m shortest path trees of the number of edges in each tree. Since the members are distributed, the number of edges in a shortest path tree from a random core is comparable to the average number of edges in a DVMRP shortest path tree.

We have found that the best core for bandwidth is much more sensitive to additions and deletions of group members than was observed in the scenarios that fixed the group and varied the sources. This is consistent with the earlier observation that small groups, even if widely distributed, require careful core choice. We can go further:

Small- to moderate-sized groups with changing members may require frequent changes to the core to optimize bandwidth.

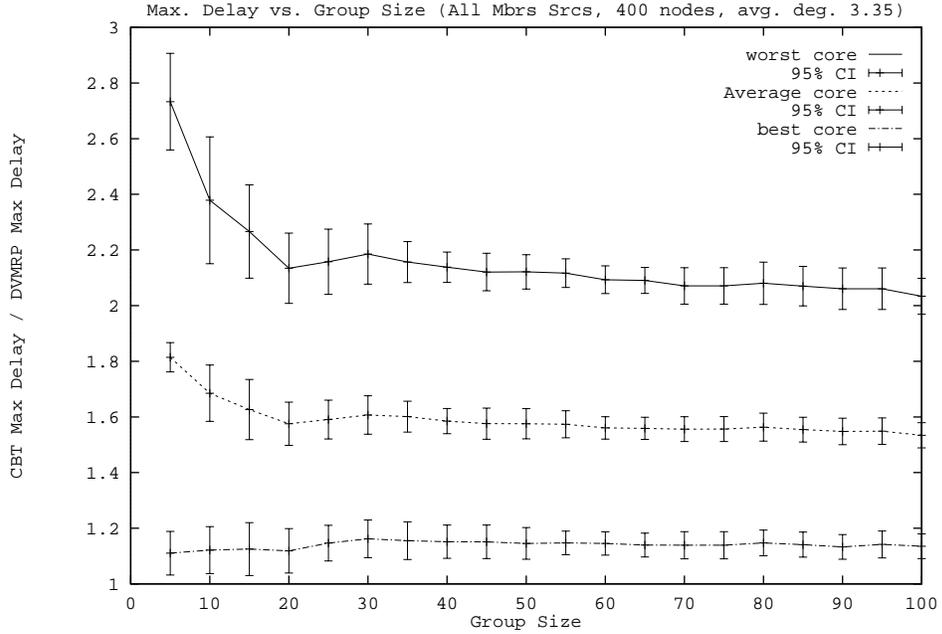


Figure 9: Scenario 4 - Maximum Delay

4.2 Graph Connectivity

We expect the connectivity of the network to have an impact on the absolute performance of the two schemes, and potentially on the ratio of performance. Greater connectivity allows more choices in constructing routes and therefore shorter shortest path trees. Figures 10 and 11 show the *absolute* values of bandwidth and maximum delay for the average CBT core and DVMRP in Scenario 4. Each pair of curves represents a different connectivity of the underlying graph.

As observed earlier, the average core bandwidth and the DVMRP bandwidth are nearly equal, for all three degrees of connectivity. The bandwidth increases nearly linearly with the group size, since each new group member must receive from and send to all other group members.

The maximum delay curves all tend to level off as the group size reaches about 20. At this point, the members are well distributed, and it takes an exceptionally far removed node to increase the maximum delay. As expected, the value of maximum delay is extremely sensitive to the connectivity of the graph. For either scheme, the maximum delay in the least connected graphs is nearly triple that in the most connected graphs.

4.3 Network Size

It is of utmost importance for an interdomain routing scheme to scale with network size. In the final experiment, we explore the performance of the two schemes as network size varies. Figures 12 and 13 show the *absolute* values of bandwidth and maximum delay for the average CBT core and DVMRP in Scenario 1. Two graph sizes are represented, 400 and 900 nodes.

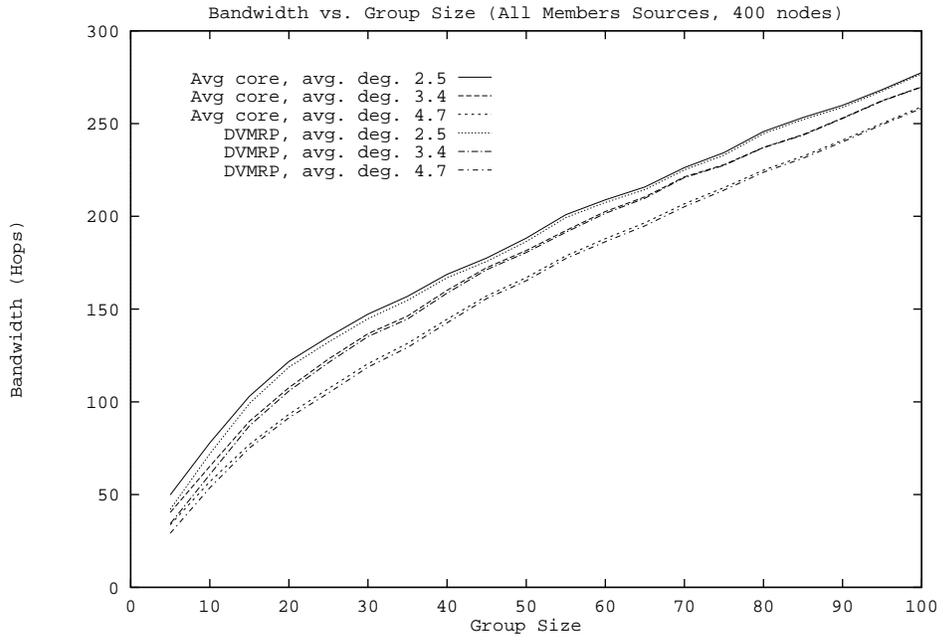


Figure 10: Varying Node Degree - Bandwidth

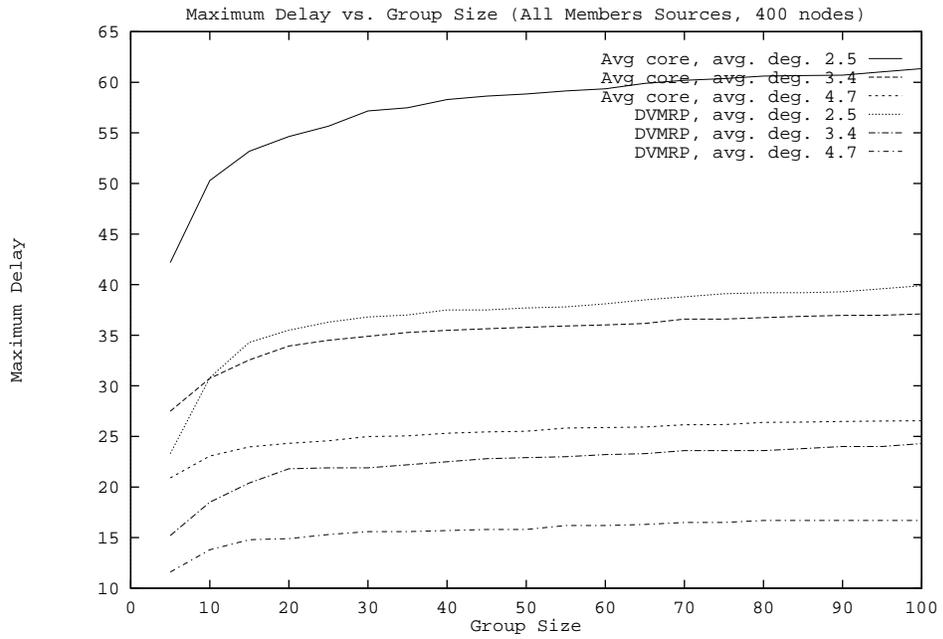


Figure 11: Varying Node Degree - Maximum Delay

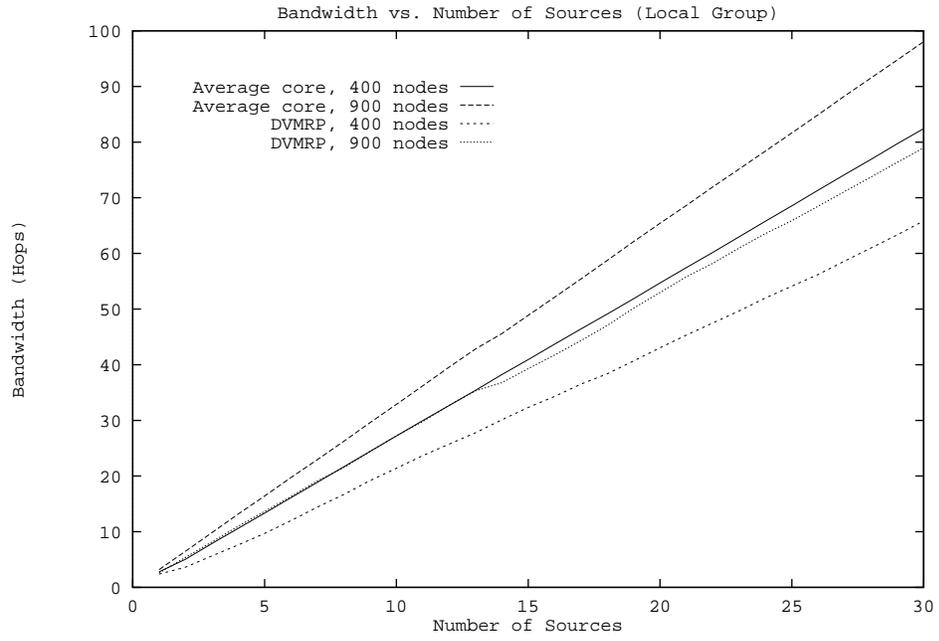


Figure 12: Varying Graph Size - Bandwidth

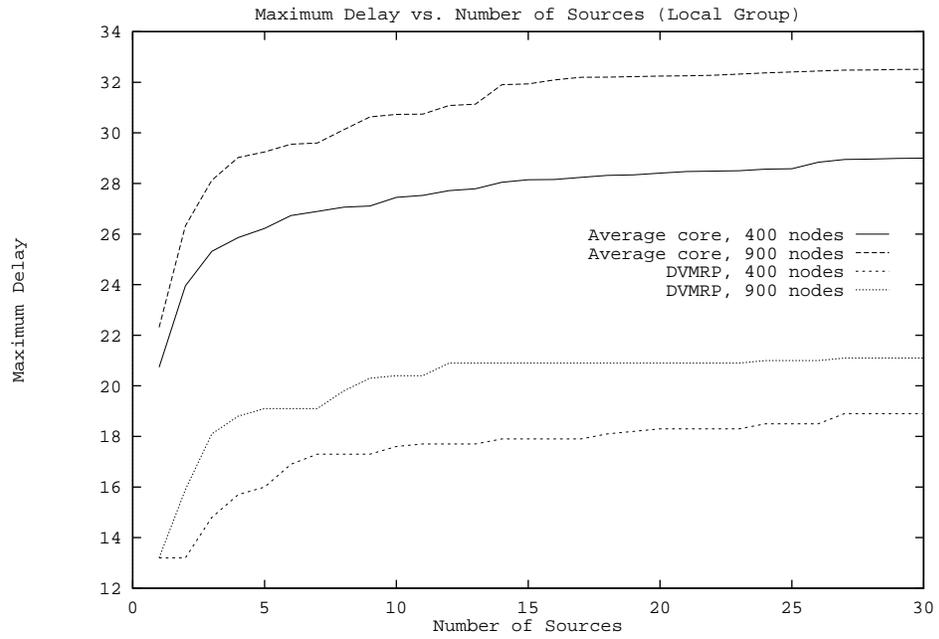


Figure 13: Varying Graph Size - Maximum Delay

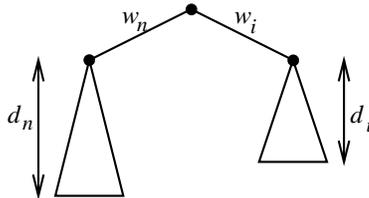


Figure 14: Determining Tree Center

Both schemes exhibit similar behavior as the graph size increases. The bandwidth is linear with the number of sources, and higher for CBT average core than for DVMRP. The difference between the two schemes is relatively insensitive to graph size. The maximum delay is considerably higher in the 900 node graph for both schemes, but this is expected.

5 A Core Evaluation Heuristic

As the above results show, the primary cost of using CBT versus DVMRP is a potential increase in maximum delay between a source and a destination. Because this cost is rather sensitive to core selection, methods of evaluating the “goodness” of a core are needed. This section describes an heuristic approach that can be used to assess a core with respect to maximum delay and also to identify a better core with respect to that metric, if one exists.

The approach is based on the notion of the *center* of a tree. For a given tree, the center is defined to be the node that, as root, minimizes the maximum depth of the tree. Thus, to minimize the maximum delay in CBT for a given group of members and sources, *the core should be the center of the portion of its shortest path tree that spans the group members and sources.*

It is straightforward to determine whether the root of a given tree is its center, given a way to determine the maximum depth of the subtree rooted at each node in the tree. Let the children of the root be numbered, and let w_i be the length of the edge between the root and child i , and let d_i be the length of the longest path from child i to a leaf (refer to Figure 14). The root is *not* the center of the tree if and only if there exists a child n such that, for every other child i , the following inequality holds:

$$d_n > w_n + w_i + d_i \tag{1}$$

Thus $d_n + w_n$ is the depth of the tree. If edge lengths are measured in hops that pass through non-CBT routers, this definition implies that a core is as close as possible to the center of the *current* tree if there is no CBT-capable router between the root and the actual center of the tree.

In any implementation of CBT, each node in the tree keeps track of its parent and children, and communicates periodically with them in order to detect loss of connectivity. It is therefore straightforward to compute depth recursively: The core asks each of its children to compute its depth; each child then asks its children, and takes the maximum of the results they return

| α | Graph # | Average Improvement | # Centered Nodes |
|----------|---------|---------------------|------------------|
| 1 | 1 | 14% | 43 |
| 1 | 2 | 30% | 10 |
| 1 | 3 | 35% | 3 |
| 1 | 4 | 22% | 20 |
| 2 | 1 | 13% | 88 |
| 2 | 2 | 4% | 223 |
| 2 | 3 | 16% | 83 |
| 2 | 4 | 16% | 66 |
| 3 | 1 | 5% | 252 |
| 3 | 2 | 8% | 159 |
| 3 | 3 | 2% | 311 |
| 3 | 4 | 5% | 236 |

Table 1: Results of Hill-Climbing Heuristic for 12 Sample Graphs

(leaf nodes return zero immediately). Using this information, the core can determine how far it is from the center of the *current* tree, and which of its children is closer to the center, if any.

Moving the core to child n , i.e. toward the center, in general will result in a *different* multicast routing tree—viz., the new core’s shortest-path tree. However, under reasonable assumptions about the shortest path trees provided by the underlying unicast routing algorithms (see Section 3), *moving the core toward the center will decrease the maximum depth of the tree*. To see this, consider that the subtree rooted at node n will not change, (More precisely, the distances to any nodes in the subtree, and thus d_n , will not change.) Moreover, the distance from n to any node m in the subtree rooted at another child i is (by the triangle inequality) at most $w_n + w_i + d_i$, which by (1) is less than d_n . Thus the maximum depth of the new tree, rooted at n , is d_n . Because the maximum depth is bounded from below, this process of moving the core can be iterated, and will eventually terminate with the core located at the center of its shortest path tree.

We tried this “hill-climbing” heuristic on some of the graphs used for evaluation of CBT and DVMRP; the results are shown in Table 1. Beginning with each node, we computed the shortest path tree, and then moved the core toward the center of the tree, recording the amount by which the maximum delay was decreased. As the table shows, the amount of improvement obtained using depends strongly on the number of nodes in the graph that are at the center of their shortest path trees; this number increases rather sharply with connectivity. (The parameter α indicates connectivity; the graphs with higher α values are more connected.) Thus, this core adjustment heuristic should be especially useful in sparsely-connected internetworks. As a diagnostic tool, however, it can be useful in any network.

6 Conclusions

The recent proposal of Core Based Trees as an alternative to source-based routing schemes like DVMRP raises the interesting question of how the two schemes compare based on the issue of ultimate interest to users—performance. We have addressed this question by examining a wide range of realistic networks and application scenarios.

Our experiments confirm intuition: DVMRP is generally superior to CBT in terms of maximum delay, while CBT can be superior in terms of bandwidth requirements. The choice of a good core in CBT is important, but it is often not necessary to have the best core: a poor choice can yield poor bandwidth and delay, but an average choice does surprisingly well for most scenarios.

Our results also indicate that different scenarios require different policies for selecting a good core. While large, widely distributed groups can get good performance from a core chosen based only on the topology of the network, small to moderate groups, even if widely distributed, require that a core choice account for the locations of the group members.

While not presented in this paper, we also examined correlation between the bandwidth and delay measures for a given core. Scatter plots revealed the existence of cores that were good/bad in both dimensions at the same time. Moreover, we found that the best core in either dimension was surprisingly insensitive to changes, especially for large groups.

In future work, we plan to compare the performance of the two algorithms in much larger graphs (thousands of nodes), investigate the effect of the core-movement heuristic on *both* bandwidth and delay, and further explore algorithms for dynamic core selection and movement.

Acknowledgements

The authors would like to thank Buddy Waxman for helpful conversations, and Jim Burns for providing the graph-drawing program.

References

- [1] Tony Ballardie, Paul Francis, and Jon Crowcroft. Core based trees (cbt) an architecture for scalable multicast routing. In *Proceedings of ACM SIGCOMM '93*, pages 85–95, 1993.
- [2] Steve Casner. Second ietf internet audiocast. In *Internet Society News*, 1992.
- [3] Ching-Hua Chow. On multicast path finding algorithms. In *Proceedings of IEEE INFOCOM '91*, pages 1274–1283, 1991.
- [4] S. E. Deering. Host extensions for ip multicasting. Internet Request for Comments 1112, August 1988.
- [5] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, California, USA, 1991.
- [6] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.

- [7] Matthew Doar and Ian Leslie. How bad is naive multicast routing? In *Proceedings of IEEE INFOCOM '93*, pages 82–89, 1993.
- [8] R. M. Karp. Reducibility among combinatorial problems. In R. A. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- [9] C. Partridge, D. Waitzman, and S. Deering. Distance vector multicast routing protocol. Internet Request for Comments 1075, November 1988.
- [10] Bernard M. Waxman. Performance evaluation of multipoint routing algorithms. In *Proceedings of IEEE INFOCOM '93*, pages 980–986, 1993.