

Core Selection Methods for Multicast Routing

Kenneth L. Calvert Ellen W. Zegura
Michael J. Donahoo

GIT-CC-95/15

Abstract

Multicast routing is an important topic of both theoretical and practical interest. Some recently-proposed multicast routing algorithms involve the designation of one or more network nodes as the “center” of the routing tree for each multicast group address. The choice of this designated router (which we refer to as the “core”) influences the shape of the multicast routing tree, and thus influences performance of the routing scheme. In this paper we investigate the relationship between the choice of core and three performance measures. Specifically, we compare various methods of selecting a core with respect to their effect on bandwidth, delay, and traffic concentration. We conclude that simple methods are adequate for widely distributed groups, but that the addition of group information can be leveraged to improve performance especially when the group is small or exhibits a high degree of locality. We also conclude that core choice has a significant impact on traffic concentration, in fact traffic concentration effects can be ameliorated by appropriate core choice policies.

Keywords: Multicast routing, Scalability, Network modeling

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 such as 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.

Shortest-path trees offer several advantages in multicast routing. They minimize the delay to every receiver in a multicast group. (We use the term “source” to designate an originator of multicast information and the term “receiver” to designate a recipient of multicast information.) Further, shortest-path trees can be constructed relatively easily in a distributed fashion, using information that is available for shortest path, unicast routing [4, 6]. Why, then, are other methods for constructing routes being considered in recent multicast routing schemes such as Core-Based Trees [1] and Protocol Independent Multicast [2, 3]? Among the reasons are reduction in router storage and control message overhead; both of these issues are addressed by the use of center (or core)-based trees, in which the choice of a designated router determines the shape of the multicast routing tree. Shortest path trees have an additional disadvantage if bandwidth is of importance. A shortest path tree does not generally minimize bandwidth, and in some cases can lead to poor bandwidth utilization in order to achieve minimum delay. Minimizing bandwidth requires solving the NP-complete Steiner tree problem [5], and thus is probably not a practical alternative. As we will see later, core-based trees can sometimes offer improvement in bandwidth utilization over shortest path trees.

Given the prominence of core-based trees in emerging multicast routing schemes, it is important to better understand the relationship between the choice of core router and the performance of the routing scheme. Specifically, our objective is to determine how complex the core choice method must be to ensure reasonable performance. We begin by assessing the circumstances in which core choice is an important determinant of performance. When there is little variation in performance for different cores, an arbitrary choice is sufficient. When the variance is substantial, more sophisticated methods to choose the core are called for. We propose several more refined core choice methods, grouping them into three categories, namely Random Choice, Topological Choice and Group-Based Choice. We analyze the performance of various methods on multicast instances, and use the results to give recommendations on core choice method.

Previous work considering the relationship between core choice and performance has focused primarily on worst case bounds [7] and performance with an optimal choice of core [8, 9]. The work by Wei and Estrin [8, 9] begins to address the relationship between core choice and performance by also considering cores chosen optimally from amongst the group members. This paper is distinguished from other efforts by the emphasis on a much richer set of core choice methods of varying complexities, and a wider range of application models.

The contributions of this paper are threefold. First, we delineate the circumstances

under which core choice has a significant impact on performance metrics of interest in multicast routing. Second, we propose a range of methods for choosing cores. Third, we have explore the performance of these methods for a variety of multicast instances and derive rules-of-thumb about when to use particular methods.

2 Methodology

This section describes our methods in analyzing the effect of core choice on performance of multicast routing algorithms. We first give a brief overview of our approach; the following subsections describe our methods in greater detail.

Based on a set of randomly-constructed graph models of networks, we define a number of different multicast routing scenarios, or *instances*. For each instance, and for each individual node in the graph, we measure the performance of a simple core-based routing algorithm when that node was chosen as core. We then normalize these measurements to account for differences among graphs and instances, and average the performance obtained for cores chosen using varying amounts of information about the instance.

2.1 Graph Models of Internetworks

We model a network in the traditional manner as an undirected graph, in which nodes represent routers (switches) and edges represent links between routers.¹ Because the principal advantage of a core-based routing method is its scalability, we used relatively large (400 node) graph models to compare core choice algorithms. These graph models were designed to reflect some locality characteristics of actual internetworks; they consist of 20 interconnected “neighborhoods”, with 20 nodes per neighborhood. The neighborhoods represent routing domains or autonomous systems, in which groups of routers under the same administration are connected via local networks and other links.

These graph models are based on “semi-geometric” graphs, which are constructed as follows: first, nodes are assigned random coordinates in the unit square. Then an edge is placed between each pair of 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, two nodes have a probability of being connected that is fixed if they are placed within a certain distance of each other, and decreases linearly with the distance between them. If the constructed semi-geometric graph is not connected, it is discarded and the process iterates.

To create a network graph model, first a semi-geometric graph of 20 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 20 nodes. The edges incident on a node in the top-level graph are connected, one at a time, to the

¹In an actual internetwork a link may connect more than two routers; we model this situation by a clique of nodes.

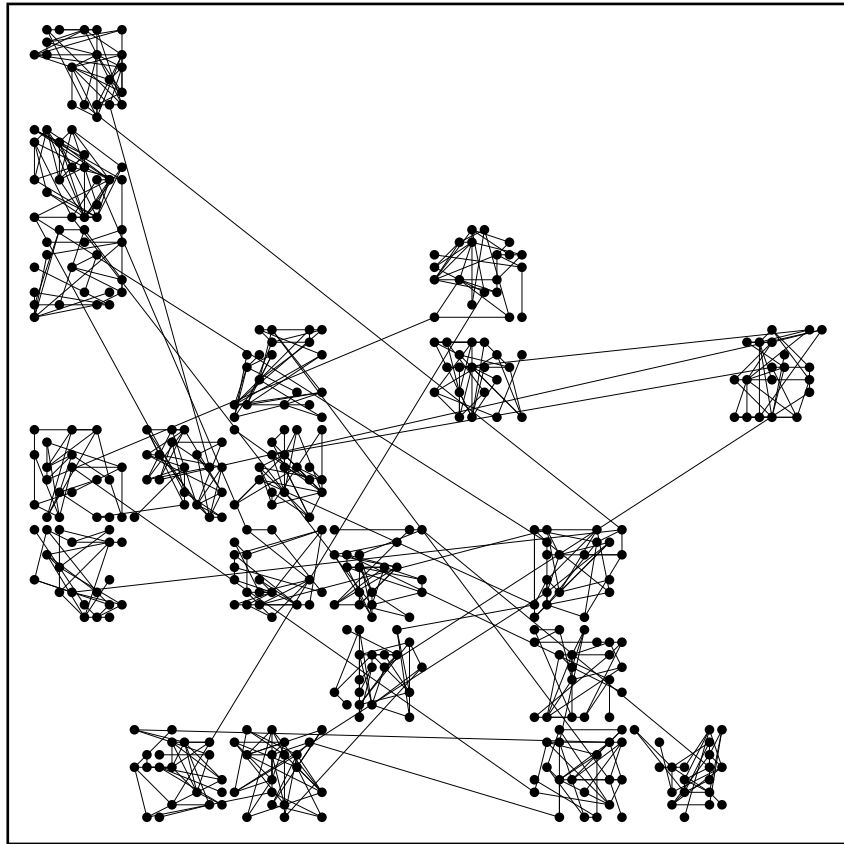


Figure 1: Example network graph

node in the replacing neighborhood that has the lowest degree greater than one. This ensures that the connection process preserves leaves, which is important because real internetworks typically have many leaf nodes. Figure 1 shows one of the graphs used in our analysis. In the graph models used in this paper, all edges have unit weight; our methods are also valid if edge weights vary.

2.2 Multicast Routing Scenarios

We use the term *instance* to refer to the input given to a multicast routing algorithm. An instance represents the multicast distribution requirements of a particular set of users for a particular application. An example of such an application would be a videoconference among a small set of participants, or distribution of a lecture series to a large, widely distributed group of students. An instance is characterized by a subset of the graph nodes designated as the multicast *receivers*, and another subset designated as *sources*;

these two sets may or may not intersect. (More precisely, the graph nodes represent “last-hop” routers, which are directly connected to the hosts that are the actual sources and receivers of multicast data.)

To evaluate the effect of core choice methods, we consider instances of three different types or *scenarios*, reflecting different distributions and numbers of sources and receivers.

All Receivers Sources. Each receiver is also a source; the number of receivers ranges from five to 15 by ones. Receivers are distributed randomly throughout the network. This scenario models, for example, a videoconference application.

Single Source, Distributed Receivers. Receivers and a single source are distributed randomly in the network, with the number of receivers ranging from ten to 100, by tens. This scenario can model a video broadcast of a lecture or meeting.

Localized Receivers. In this scenario there are 20 receivers, all constrained to be within the same neighborhood. The sources are distributed randomly throughout the network; the number of sources ranges from five to 50 by fives. An example of an application with these characteristics might be a distributed database: database clients (sources) submit transaction requests to servers (receivers) via multicast, with each response being unicast from a server back to its originator.

2.3 Multicast Routing

For a given graph and instance, a multicast routing algorithm defines a *distribution tree*, which determines the path followed by each packet sent by a source on its way to all the receivers. Each distribution tree is derived from the shortest-path trees provided by the underlying internetwork (unicast) routing protocol.

In the core-based routing algorithm we consider, the tree for each source consists of two parts. One part is the same for all sources and forms a tree, namely the shortest-path tree from the core to all the receivers. The other part, which may differ from source to source, is that portion of the shortest path from the source to the core that is distinct from the core’s shortest-path tree. In cases where a source node is on the shortest path from the core to some receiver, this part is empty.

Figure 2 shows an example multicast routing tree. In this instance there are two sources (marked “S1” and “S2”) and six receivers (marked “R”); the core node is marked “C”. The shortest-path tree from the core to all receivers is indicated by the darker edges. The links traversed by packets sent by S1 are marked by dotted arrows; the path followed by packets sent from S2 are marked by solid arrows. A packet need not travel all the way to the core before being routed toward receivers; as soon as the packet encounters a node in the core’s shortest-path tree, thereafter it follows the shortest route *in the tree* to each receiver. Note that in general, a packet does not follow the shortest path from its source to each receiver; this is evident in the figure for the path followed by packets from S1 to the lower-left-most receiver.

The *routing subgraph* comprises the distribution trees for all sources. The form of the routing subgraph depends not only on the instance, but also on the location of the node designated as core; this is the motivation for our investigation.

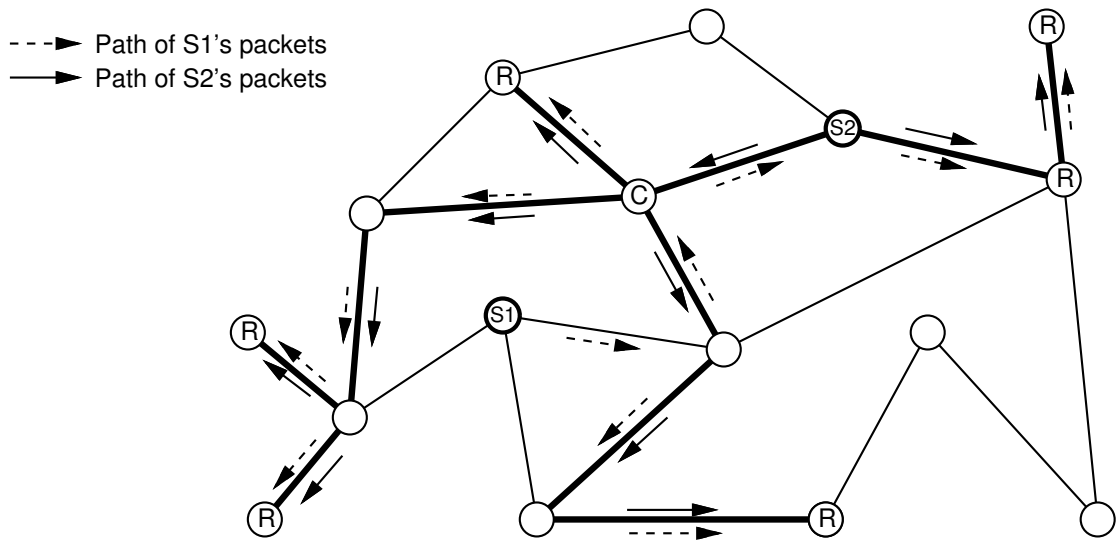


Figure 2: Multicast routing subgraph for two source nodes, S1 and S2. The node marked “C” is the core, nodes marked “R” are receivers.

2.4 Performance Measures

The routing subgraph (computed as described above) determines three quantities of interest in characterizing the algorithms’ performance:

- The maximum number of links traversed by any packet in traveling from a source to a receiver. This gives an upper bound on the *delay* experienced by multicast packets in an *unloaded* network (no queueing in switches). For a given routing subgraph, we measure the length (in hops) of the longest shortest path between a source and a receiver; throughout this paper the term “delay” refers to this fixed component of the actual delay. (We consider maximum source-receiver path length, as opposed to average source-receiver path length, because many multicast applications have a real-time flavor, and are therefore interested in *bounds* on delay.)
- The total number of packet transmissions required to deliver a packet from each source node to all receiver nodes. This gives an indication of the resources used by the algorithm, in particular *bandwidth*. For a given routing subgraph, we compute the sum, over all sources, of the total number of edges traversed by a packet sent from that source to all receivers, and call that sum the “bandwidth” measure of the algorithm.
- The number of packets transmitted across each link in the network when each source sends to all receivers. This gives an indication of the *traffic concentration* effect of the multicast routing algorithm. Traffic concentration is an indicator of

potential delays due to queueing; thus we consider the two directions of transmission of an edge separately. For a given instance and core, we compute, for each edge in the subgraph and each direction, the number of source distribution trees that include that edge-direction.

For each graph and instance, we perform the following computation: for each node in the graph, compute the routing tree with that node as core, and compute (and record) bandwidth and delay as described above. This is repeated for each combination of instance and number of receivers, for each of 10 different graphs of 400 nodes each. Analysis of this collected data gives an idea of the magnitude of the effect of the core location on performance. Figure 2.4 shows the distribution of bandwidth and delay measures across all possible cores for one example instance.

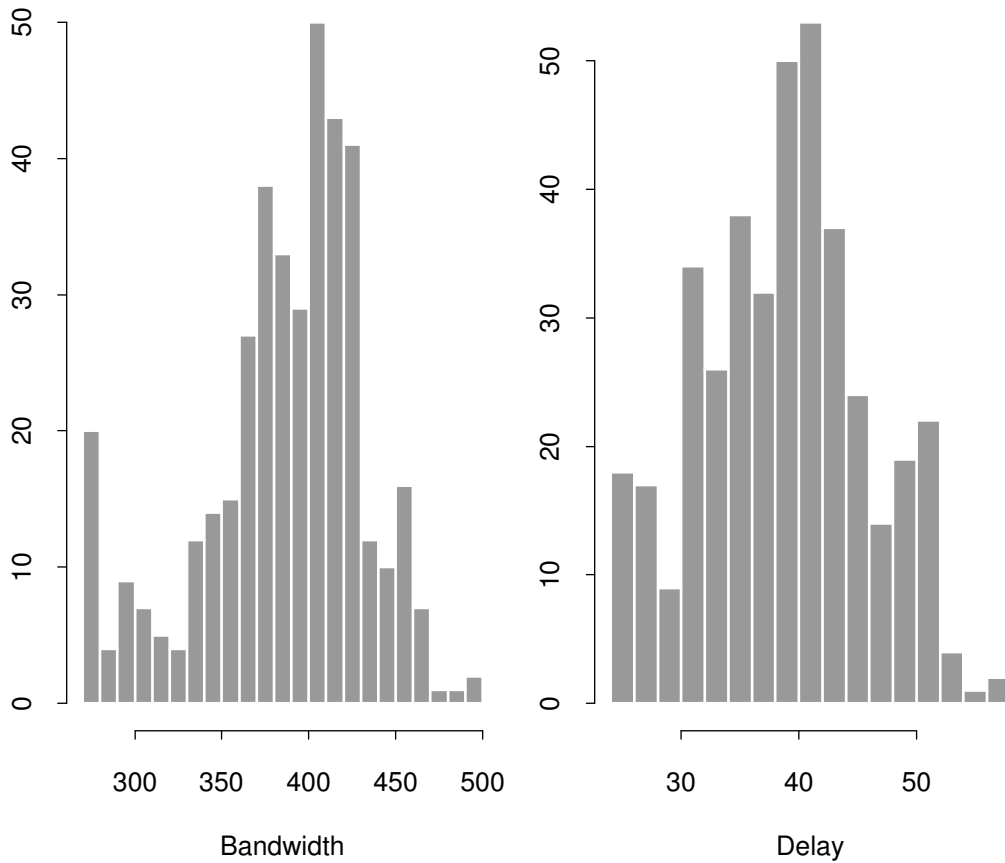


Figure 3: Distribution of bandwidth and delay performance across all possible cores.

In order to compare performance *across* graphs and instances, we also we also computed bandwidth and delay measures for a different routing approach, namely shortest-path routing, in which each packet travels along the shortest path from its source to

each receiver. The results presented later are given in terms of the average *ratio* (over different sets of cores) between performance achieved with core-based routing and with shortest-path routing. Among other advantages, this allows direct comparison of the effect of core choice on the different performance measures.

2.5 Core Choice Methods

The information collected as described above can be used to evaluate the effects that various methods of choosing a core have on multicast routing performance. The next several sections present our analysis for five different classes of core selection methods, which span a range of complexity. They are:

Arbitrary. We investigate the limits of the negative effects of core choice on performance. In other words, this section answers the question, “If the worst possible core is chosen, how bad can it be?” This gives a bound on performance when a core is chosen in the complete absence of other information.

Random. With this method, a core is selected randomly from among all the nodes of the graph. We examine the performance obtained by averaging over all possible cores. We show that this method, which requires very little information about the network or instance, also has advantages from the standpoint of traffic concentration.

Topology-Based. These methods make use of information about the topology of the network. In particular, we analyze the relationship between performance and nearness of the core to a topological center of the network.

Group-Based. These methods make use of information about *both* the network topology and the location of the nodes (sources and receivers) that make up the group. The availability of such information can potentially enhance the attainable performance.

3 Arbitrary Core Choice

We begin by examining the performance of a core chosen arbitrarily. This method of choosing a core is the least complex of the methods we consider, requiring no knowledge of the group, the network topology, or previously chosen cores. This method will be appropriate only when core choice does not significantly affect the performance of the core-based tree. To determine the practicality of an arbitrary core choice, we consider the performance of the *worst* core. The rationale for looking at the worst core is that an arbitrary choice *may* select this node as core. Since the selection is arbitrary, this method offers no guarantee of choosing a core that is better than the worst possible.

Figure 4 shows the bandwidth (delay) performance of the core with the highest bandwidth (delay). As described in Section 2, the performance is shown as a ratio to the performance in a shortest-path routing. The x-axis is the percentage of receivers from the range specified for the scenario (e.g. for All Receivers Sources the number of receivers ranges from 5 to 15 so 50% on the x-axis represents 10 group members for

that scenario). Observe that for delay, all three scenarios performed approximately 2.5 times worse than the shortest-path routing, demonstrating that an arbitrary core choice can result in significantly higher maximum delays. For bandwidth, the performance is sensitive to scenario, but for the All Receivers Sources and Localized Receivers scenarios, the performance is significantly worse than shortest-path routing (Note that while a shortest-path routing does optimize delay, it does not, in general, optimize bandwidth. Thus, ratios lower than 1.0 for bandwidth are possible). The Single Source scenario performs better than the other scenarios for bandwidth because it has a large number of well distributed receivers so the placement of the core does not affect performance as much as it does for the other scenarios.

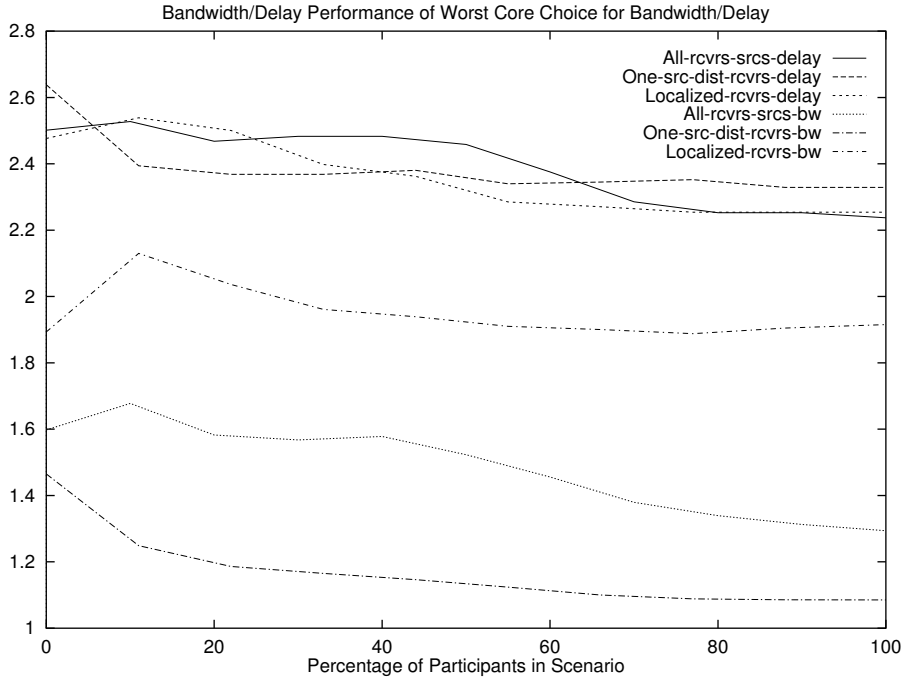


Figure 4: Bandwidth and Delay for Worst Core Choice

While arbitrary core choice is easy to perform, it should be clear from the preceding discussion that there are many scenarios in which an arbitrary core does not perform well. Indeed, it is easy to construct pathological cases in which an arbitrary core is even worse than the performance measured in these scenarios. (e.g. a group with highly localized sources and receivers, and a core that is far removed from the rest of the group). Further, an arbitrary core choice may repeatedly select a node with poor performance. The next method uses a small amount of information about the network to ensure that bad cores are not chosen repeatedly.

4 Random Core Choice

The random core choice method requires knowledge of the nodes in the network, in order to select a core at random from all nodes. This method allows some hedging

against outliers that perform poorly; if only a few nodes make poor cores, the average performance may still be reasonable. Random core choice requires no knowledge of the scenario for which the core is selected.

To determine the feasibility of random core choice, we average the core performance in each scenario. Figure 5 shows the average bandwidth and delay ratios of a random core choice to the shortest-path routing. Observe that the figure looks very similar to Figure 4 except that the range of y-axis is reduced by almost 40%; as noted earlier, the delay performance is similar across the three scenarios, while the bandwidth performance varies somewhat more across the scenarios. In all cases, the random core choice method performs better than the arbitrary core choice method. To further explore random core choice, consider the variance of core performance (See Figure 6 - The Single Source Scenario cannot be seen due to the range of the y-axis). The high variability of the random core choice method for the Localized Receivers and particularly the All Receivers Sources scenarios implies that some applications may experience a wide range of bandwidth performance across instances, possibly making a random core choice unacceptable (Users may find it unacceptable to be given guarantees that apply only to the average performance across multiple instances, rather than a guarantee about performance of their particular instance).

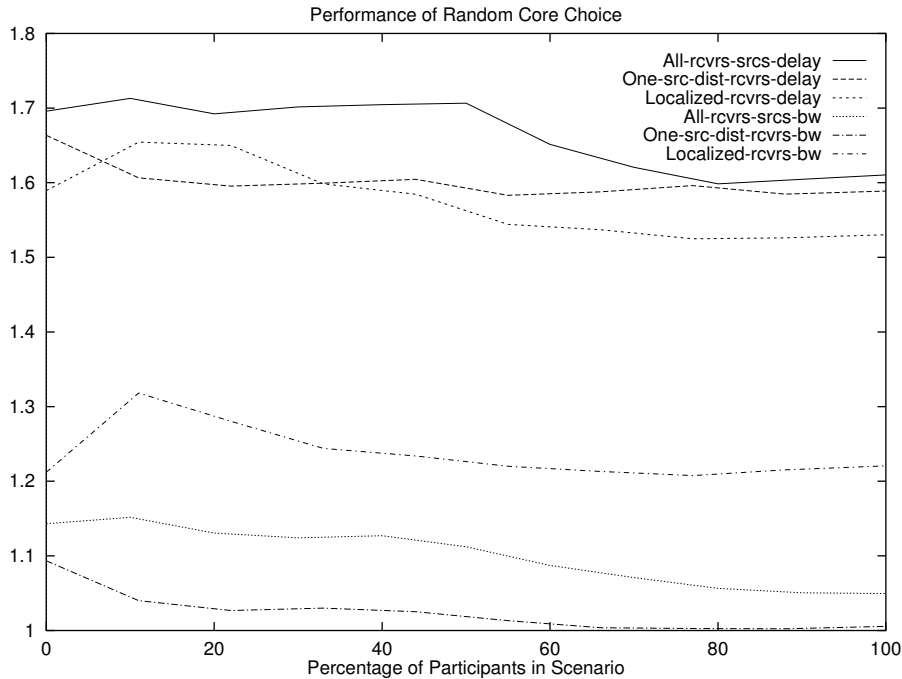


Figure 5: Bandwidth and Delay for Random Core Choice

5 Topology-Based Core Choice

Despite the improvement we get from random over arbitrary, a random core still results in average delay that is 60-70% worse than the shortest-path routing delay. While the

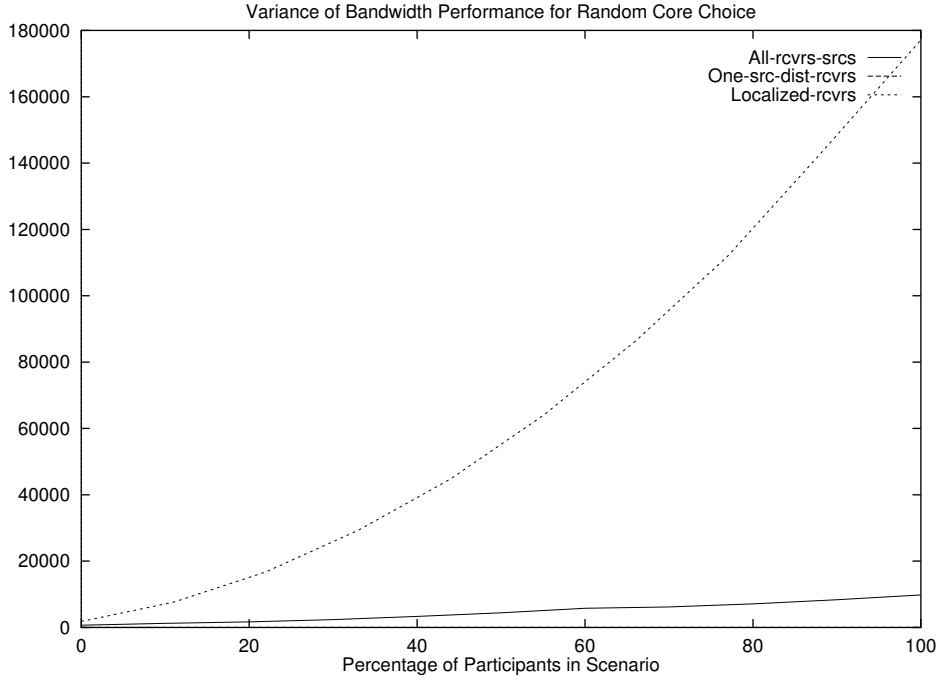


Figure 6: Variance for Bandwidth with Random Core Choice

bandwidth approaches the shortest-path routing bandwidth, the variance is quite high for some scenarios. We now move to methods that use additional information about the topology of the network to make better core choices. These methods will require more knowledge than a random choice, but this is knowledge of the gross topological characteristics that do not change as quickly as the finer details of internetwork structure. Consequently, the time scale for recalculation of the topology measure used in the topology-base choice method is very coarse (e.g. once a week) relative to the time scale for a group-based choice. In addition, choosing a core based only on topology is attractive, as it requires no knowledge of the group members or sources. A core that performs well irrespective of group location will be robust to changes in the group.

We define a single method that chooses a core according to the depth of the shortest-path routing from each node. Formally, let $d.u$ denote the depth of the shortest-path route rooted at u (The depth of a tree rooted at u is the maximum length from u to any leaf in the tree). A *center* of a network is any node, c , that minimizes $d.c$. Our method takes a parameter t and selects a core, u , at random from those satisfying $d.u \leq d.c + t$. In effect t specifies a tolerance (in excess of the minimum achieved by a center) on the depth of the shortest-path routing. By varying the parameter t , we get a family of methods between the extreme of choosing a center ($t = 0$) and choosing a node at random. We will let Tol- t denote this method with parameter t . The intuition is that nodes at or near the center should have, on average, better performance than nodes chosen at random.

We first consider the performance for Tol-0 (network centers). Figure 7 shows the bandwidth and delay ratios of the core-based tree to the shortest-path routing for the

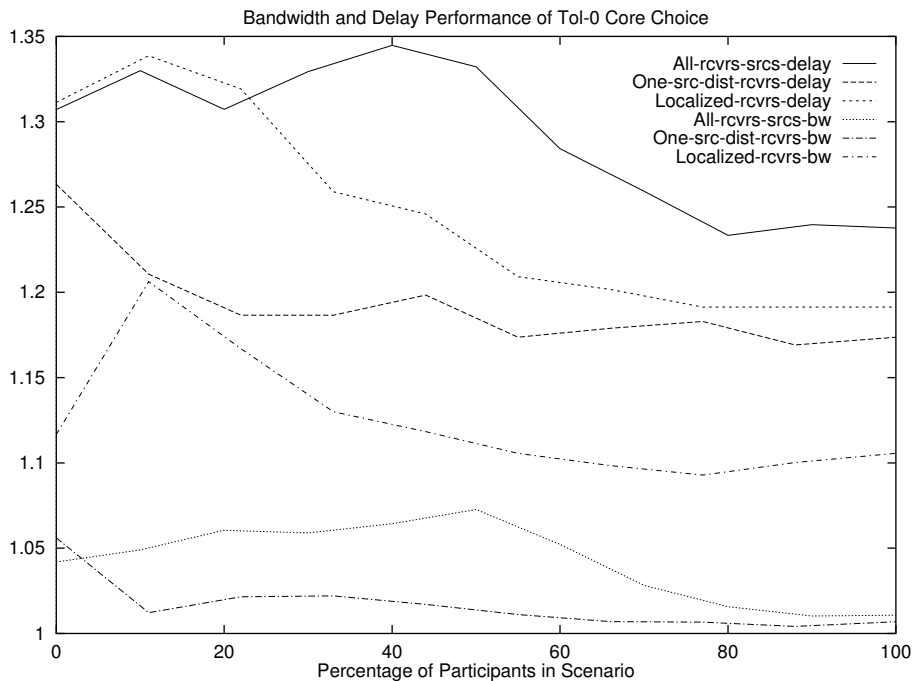


Figure 7: Bandwidth and Delay Performance Ratios for Tol-0

three scenarios. Again, the graph looks similar to the graphs for arbitrary and random choice, but with a reduction in the range of the y-axis by 25% from the random choice. Note that the effectiveness of this core choice method is more sensitive to scenario. For both bandwidth and delay, the Single Source scenario has the best performance relative to the shortest-path routing. With just one source, it is easier for a core-based tree to be competitive with a shortest-path routing.

There are at least two reasons to consider the performance of a topological core choice as the tolerance t is increased. First, since the graph center would likely be determined on a coarse time scale, we need to examine the effects of imperfect center information on core performance. Second, to distribute the network load, we might want to select cores from a larger set than the set of network centers (This particular issue of network load is examined further in Section 7).

Consider the performance when t is increased. Intuitively, we would expect the delay to increase as the tolerance increases. Figures 8 and 9 show the ratio of delay for core-based trees to delay for shortest-path routing for the All Receivers Sources and Single Source Scenarios, respectively (The graph for Localized Receivers is very similar but is excluded for brevity). An increase in delay as the tolerance increases is consistent across all group sizes, with a significant difference between choosing from a center (Tol-0) and choosing at random. It is interesting that the shapes of the curves for a particular scenario remain similar across all tolerances.

The effect of increasing the tolerance is less pronounced for bandwidth performance, particularly for the All Receivers Sources and Single Source Scenarios. As shown in Figure 10 increasing the tolerance by a moderate amount actually improves bandwidth

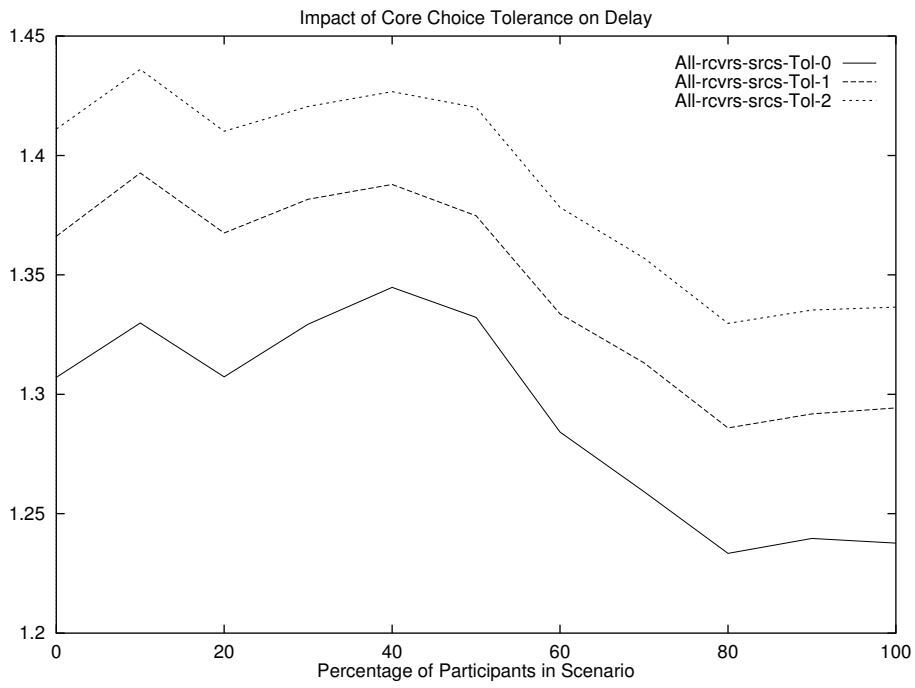


Figure 8: Impact of Core Choice Tolerance on Delay for the All Receivers Sources Scenario

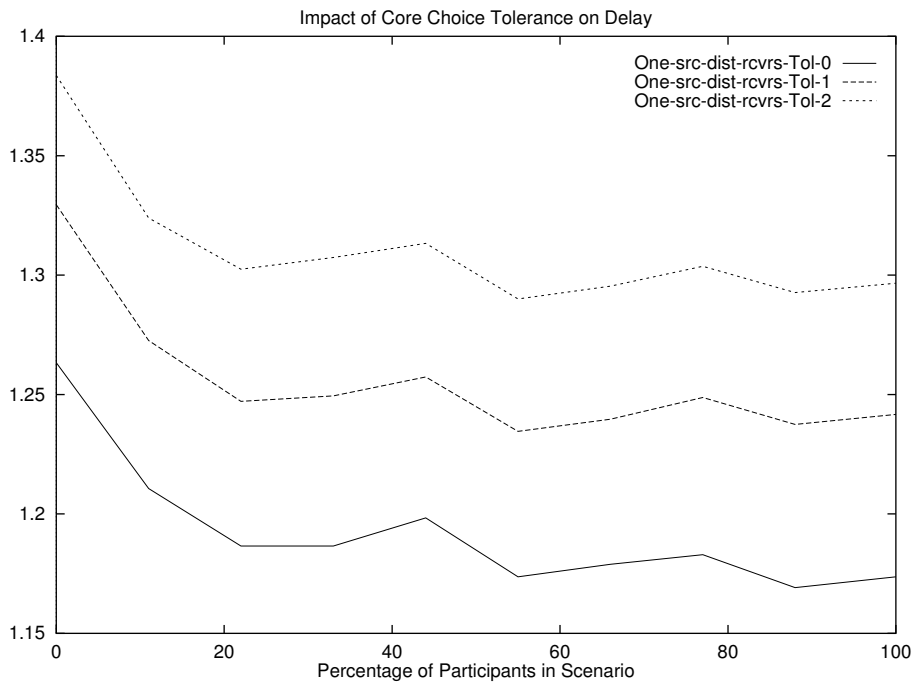


Figure 9: Impact of Core Choice Tolerance on Delay for the Single Source Scenario

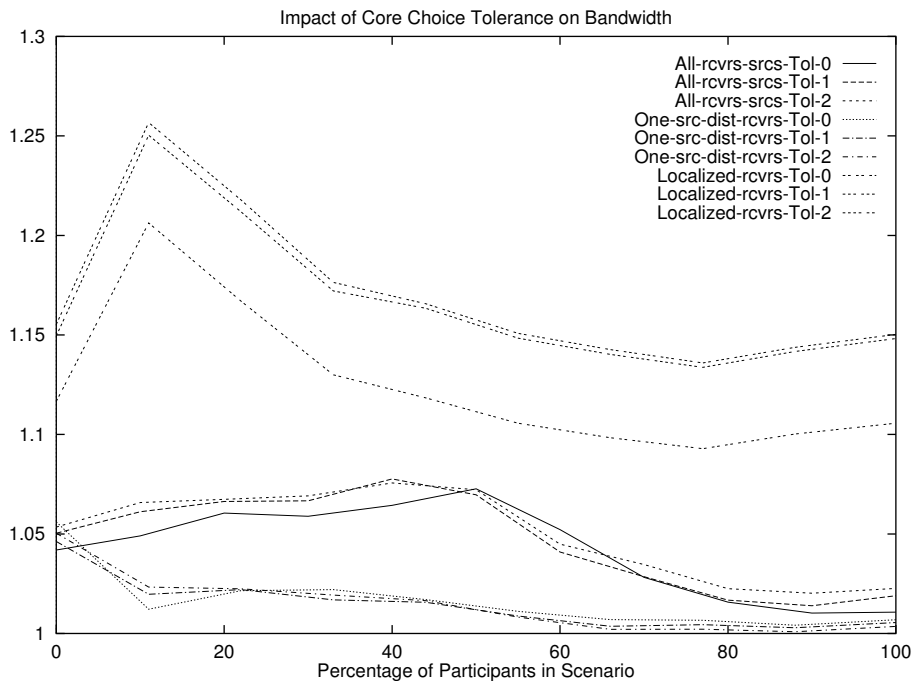


Figure 10: Impact of Core Choice Tolerance on Bandwidth

performance for smaller groups in these scenarios. For the Localized Receiver Scenario, bandwidth increases with the tolerance, but for tolerances above one the increase is negligible. Note that since the position of the core determines the root of a shortest-path routing, the relationship between core choice and performance is far less direct when considering bandwidth than delay. In other words, we can directly control delay by core placement, but we cannot directly control bandwidth. These results reflect that indirect relationship.

6 Group-Based Core Choice Methods

For arbitrary, random, and topologically based core choice methods, the localized receiver scenario is consistently the worst with respect to bandwidth. This leads us to consider several core choice methods that make use of information about the locations of the receivers and/or sources. Note that the group-based core choice requires the most complex knowledge of the methods we consider; the characteristics must be determined per group rather than per network. The practical application of these methods depends not only on the availability of information about the receivers (group members) and their locations, but may also require efficient and reliable methods of changing the core when the participant distribution changes.

For comparison we also consider *performance-based* core choice methods: these select from a set of nodes (receivers, sources, or all nodes) the node with the *best* value of some weighted combination of bandwidth and delay. Such a method requires even more information than group-based methods, because the performance of all nodes in the

subset must be known (or estimated) to make the choice.

We consider the following methods of choosing a core:

Random Receiver One of the receivers is selected as core. Results reported represent the ratio of average bandwidth (delay) over all receivers to the shortest-path routing bandwidth (delay).

Center of Receivers A core is chosen from among the topological centers of the subgraph induced by the receivers. This is only possible when the induced subgraph is connected, which is only guaranteed in the Localized Receivers scenario. Reported results represent the ratio of the average over all centers of the induced subgraph to the shortest-path routing.

Best Receiver The bandwidth and delay for the given configuration is measured with each receiver as core, and then the best is chosen according to a weighted combination of bandwidth and delay. Results reported for this metric are either strongly weighted towards delay or bandwidth, and represent the ratio of that core’s measures to the shortest-path routing.

We compare the group-based methods, along with the Tol-0 topological choice, for each of the three scenarios. As a point of reference, we also include the performance of optimal core choices, with curves labeled Best-BW and Best-DL. These curves indicate the bandwidth (delay) required by the best core for a heavily weighted towards bandwidth (delay) combination of bandwidth and delay. Best-BW-Rcvr (Best-DL-Rcvr) indicates the bandwidth (delay) of the core among the *receivers* with the best value of the same metric.

Intuitively, group based core choice methods should improve performance over topologically-based methods at the least for applications with localized participants, because the core will be placed near the participants. Consider the Localized Receivers scenario where the knowledge of localization can be used to improve both bandwidth and delay (see Figure 11). Choosing the topological center of the subgraph induced by the receivers or just choosing a receiver at random results in bandwidth and delay performance which is better than the performance of a graph center. In fact, the bandwidths of the receiver center and the random receiver are almost as good as the bandwidth performance of the optimum core for bandwidth. Observe that while the receiver center provides the best performance for localized participants, if finding the center is computationally unacceptable, selection of a random participant effectively gives a comparable bandwidth performance and less than 10% increase in delay.

Group information does not necessarily improve performance. In fact, use of group-based methods may actually hurt performance relative to topology-only methods, if the receivers and sources are widely distributed. Figures 12 and 13 show this for the All Receivers Sources and Single Source scenarios. In both scenarios, little change in bandwidth results in changing from a random graph center to a random receiver, while the change results in a significant increase in delay.

To highlight the differences in core performance along the two performance dimensions (bandwidth and delay), we present in Figure 14 a representative scatter plot showing the absolute (bandwidth, delay) measure for each of the 400 nodes in one

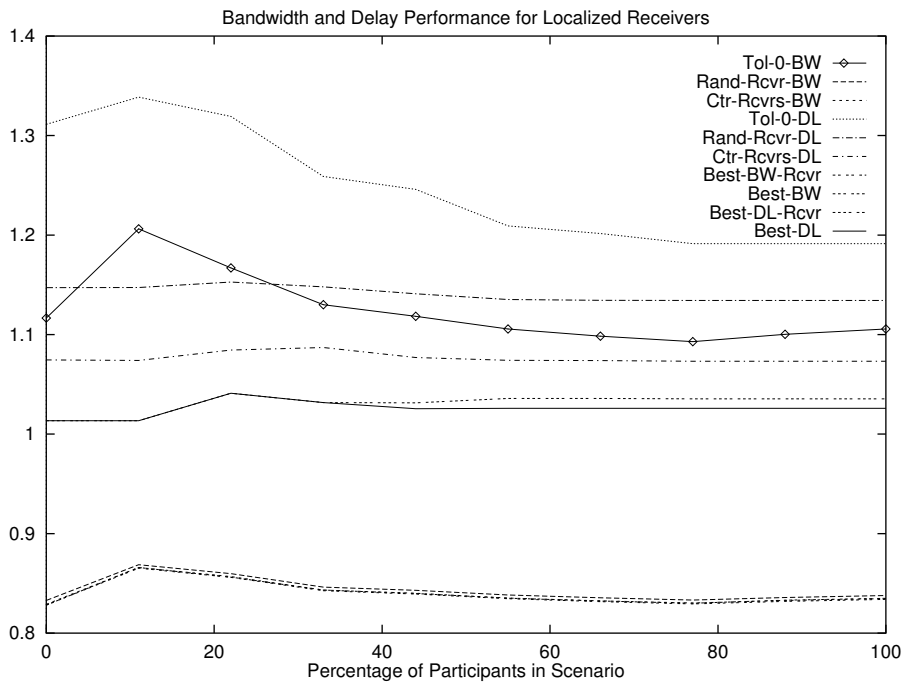


Figure 11: Effect of Core Choice Method for Localized Receivers

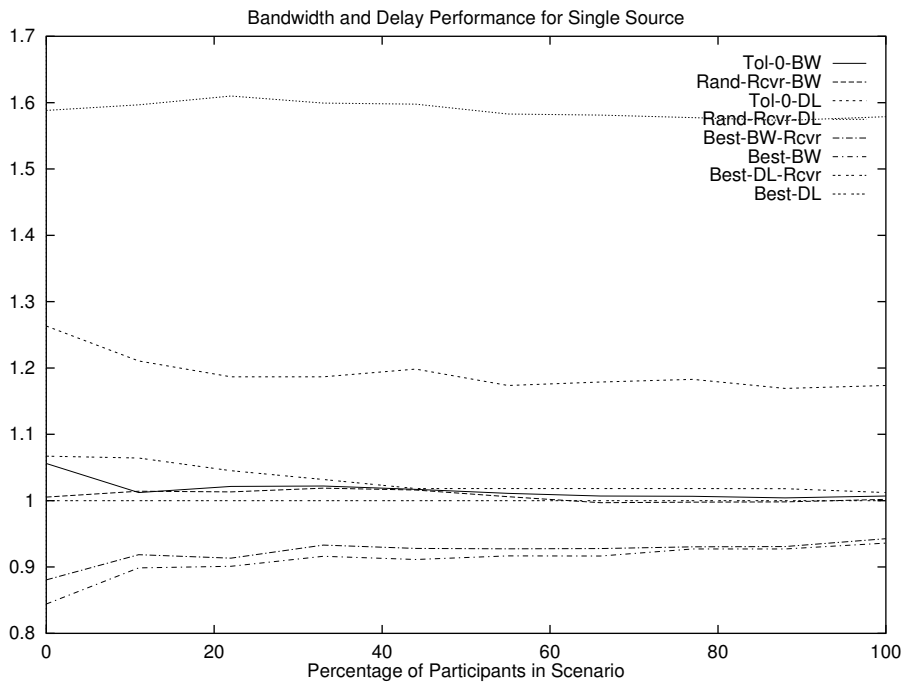


Figure 12: Effect of Core Choice Method for Single Source

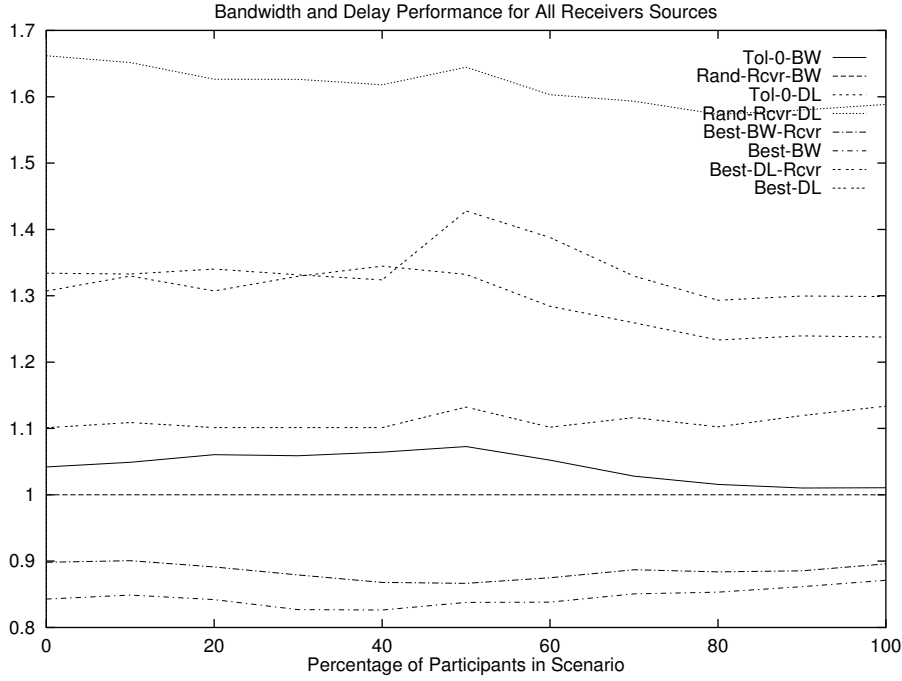


Figure 13: Effect of Core Choice Method for All Receivers Sources

graph under the All Receivers Sources scenario. Receivers (“Group Members”) and the three centers of the graph are marked differently; the shortest-path routing value is shown with a plus. There are fewer than 400 points in the scatter plot since multiple nodes can have the same (bandwidth, delay) measure. Nodes closest to the origin represent good performance along both dimensions; note that several of these nodes are neither centers nor receiver/sources indicating the capability for further improvement.

7 Traffic Concentration

One of the purported drawbacks of core-based trees is their tendency to concentrate traffic on certain links of the network, due to the sharing of the core-based routing tree by all sources in a group [2, 9]. We demonstrate that traffic concentration is *not* inherent in the core-based tree approach; rather it is sensitive to core choice and can be virtually eliminated (if desired) by choosing cores at random.

To measure traffic concentration, we model a large number of connections in the same network. For each connection, we determine the routes for shortest-path and core-based trees, and determine the traffic concentration on each link. The results presented here are for a 49 node graph with 300 groups and 10 members per group, with each group an All Receivers Sources scenario. Figure 15 shows the histogram of link load for shortest-path trees. Most links carry a load of under 1000 flows; the average load per link is 440. At the extremes, there is one link which carries over 2000 flows; the minimum number of flows carried by any link is eight.

We now consider core-based trees with the core for each group chosen from one

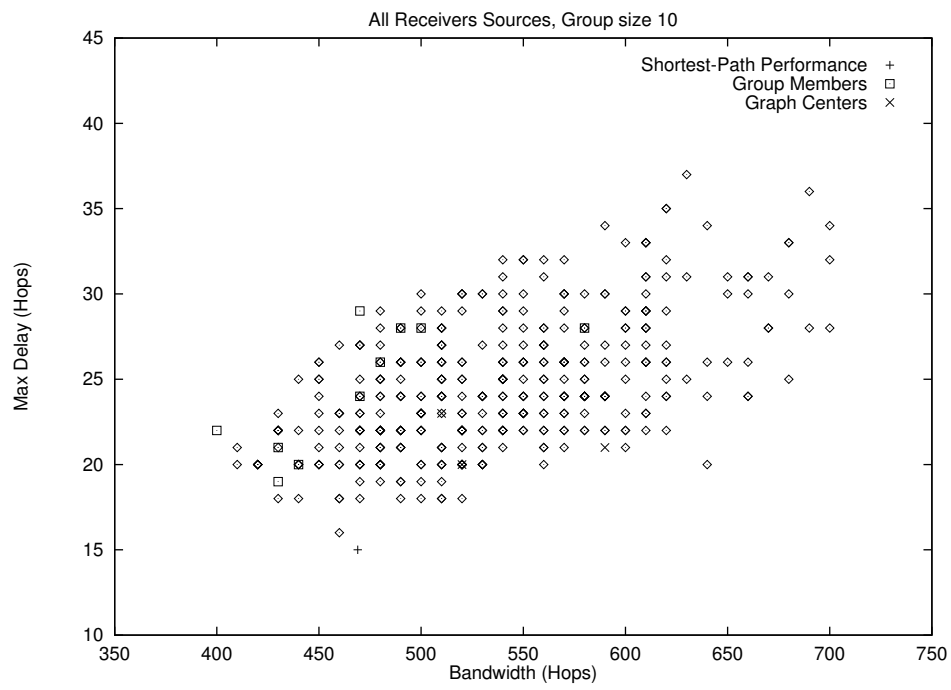


Figure 14: Representative bandwidth-delay plot

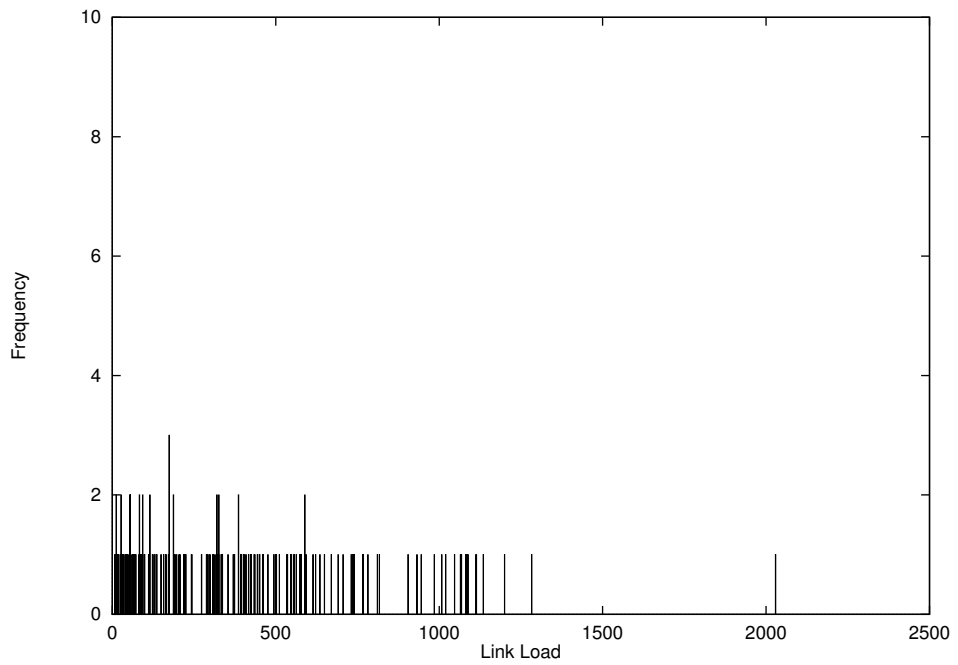


Figure 15: Traffic Concentration in Shortest-path Trees

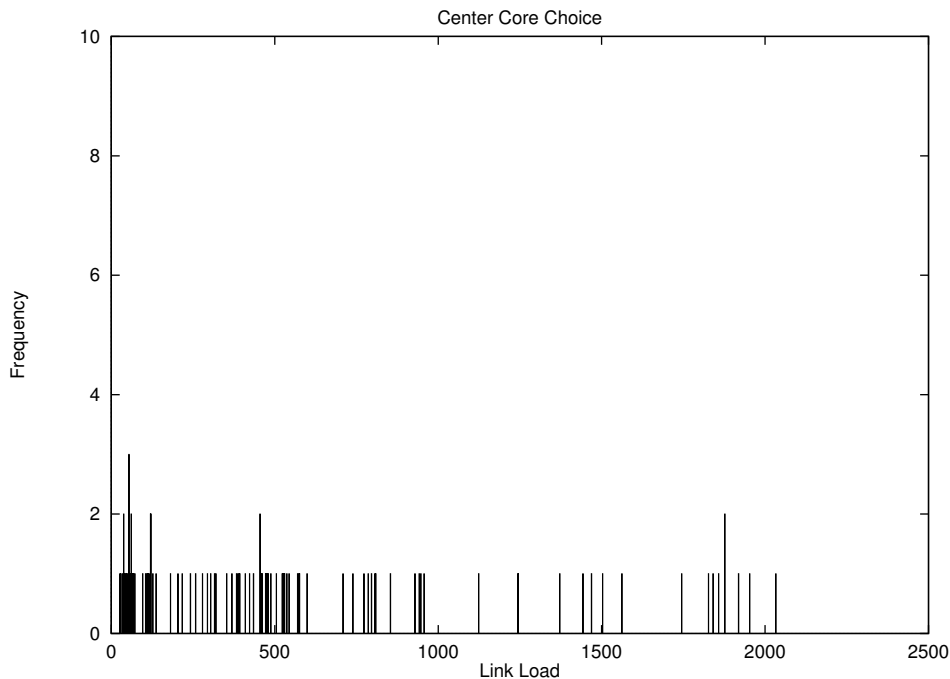


Figure 16: Traffic Concentration in Core-based Trees - Center Core Choice

of the two topological centers of this particular graph. Figure 16 shows the link load histogram for this method of core choice. Considerably more links carry load over 1000 flows; the average load per link is 580. The most significant traffic concentration occurs on a cluster of links with load around 2000 flows, due to the use of only two cores by all of the groups. There are 50 links which carry no traffic whatsoever and are not included on this plot. These results are qualitatively similar to the traffic concentration reported by other authors [9].

To demonstrate that traffic concentration is influenced (and can be alleviated) by core choice, we also compute the link load when the core for each group is chosen at random from all nodes in the graph. Figure 17 shows this result. In marked contrast to the center core choice, this method of core choice does an excellent job of distributing the traffic. In fact, the average load per link is 420 and the maximum is just under 2000 — both improvements on the shortest-path tree measures!

Just as delay and bandwidth are affected by core choice, so too is traffic concentration. The key to avoiding overloaded links in core-based trees is to choose cores from a reasonably large set of routers.

8 Summary and Conclusion

Our results yield no single best core choice method; instead, they suggest points of consideration when making a core choice and demonstrate the potential consequences. To select from our core choice methods, an administrator must consider the tradeoffs between performance and required information. At one extreme, no information about

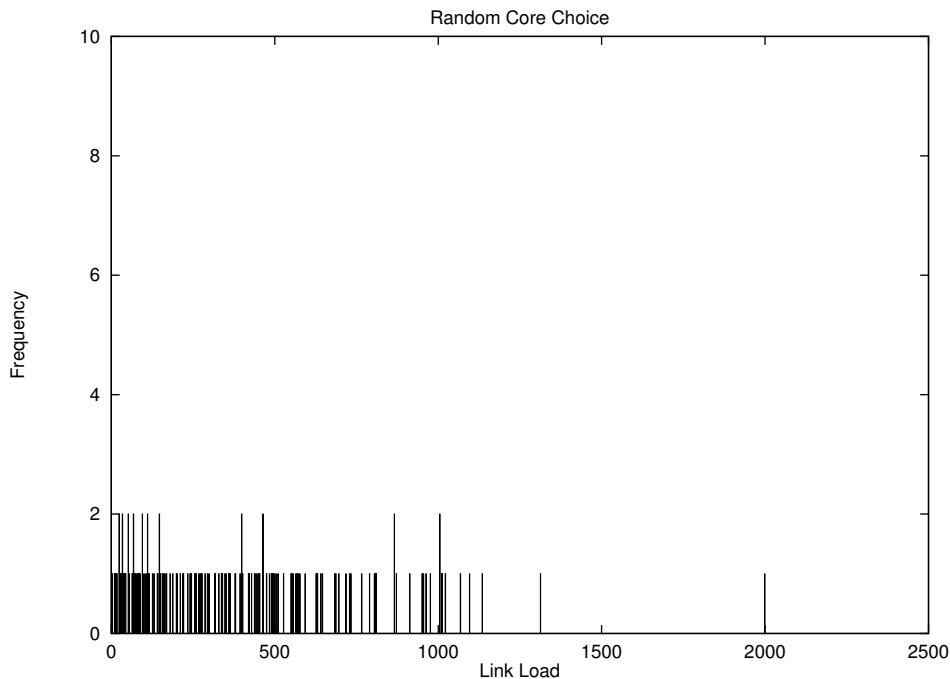


Figure 17: Traffic Concentration in Core-based Trees - Random Core Choice

network topology is required for the arbitrary core choice. In fact, an administrator may designate one core for use by all multicast groups. The advantage is simplicity, but bad core performance and increased traffic concentration may result. At the other extreme, the administrator acquires information about network topology and group distribution (e.g. whether receivers are localized) to designate a core that offers good performance for that particular instance. In this case, the core choice method uses the application semantics to decide between a topological or group based core choice. These tradeoffs are important because in general, information about the network and group topology is not trivial to come by.

For each scenario and each of the four core choice methods, Tables 1 and 2 show bandwidth and delay performance measures, respectively, for the smallest group instance of each scenario. The improvement of bandwidth and delay performance for random core choice over arbitrary is approximately 30% for both bandwidth and delay. In addition, moving from arbitrary to random core choice also reduces the chance for core clustering which results in traffic concentration. In choosing between arbitrary and random, the administrator must decide if insuring random core placement is worth avoiding the performance extremes of the arbitrary core choice.

Observe the improvement in using the topology-based core choice instead of the random core choice. While bandwidth performance has almost negligible improvement for the Localized Receivers and All Receivers Sources Scenarios, the delay decreases by approximately 20%. If delay is important and several groups will use the topological center of the network before it is recalculated, choosing the topology-based core choice makes sense. Finally, for group-based choice, note that the administrator needs to

know the application semantics because the performance of the group-based choice is scenario dependent. If the scenario has highly localized participants, then the group-based choice offers a 22% improvement in bandwidth over the topology-based choice and loses nothing in delay performance. However, if the participants are not localized, bandwidth performance improves nominally but delay increases by approximately 25% from the topology-based choice. In this case, the administrator should use a method other than group-based to place the core.

	Localized Receivers	Single Source	All Receivers Sources
Arbitrary	1.9	1.5	1.6
Random	1.2	1.1	1.15
Center	1.1	1.05	1.05
Group	0.85	1.0	1.0

Table 1: Comparison of Bandwidth Performance for Core Choice Methods

	Localized Receivers	Single Source	All Receivers Sources
Arbitrary	2.5	2.6	2.5
Random	1.6	1.65	1.7
Center	1.3	1.25	1.3
Group	1.15	1.6	1.65

Table 2: Comparison of Delay Performance for Core Choice Methods

We have considered scenarios representative of a few multicast applications, using a relatively simple graph model. Future work will make use of improved application and network models and will consider the effects of multiple cores and dynamic core movement.

Acknowledgements

The authors would like to acknowledge contributions to this work by Megan Thomas and Ramesh Madhavan.

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] S. E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (PIM): Motivation and architecture. Working Draft, March 1994.
- [3] S. E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (PIM), sparse mode protocol specification. Working Draft, March 1994.
- [4] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internet-networks and extended lans. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.
- [5] 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.
- [6] C. Partridge, D. Waitzman, and S. Deering. Distance vector multicast routing protocol. Internet Request for Comments 1075, November 1988.
- [7] D. W. Wall. *Mechanisms for Broadcast and Selective Broadcast*. PhD thesis, Stanford University, California, USA, 1980.
- [8] L. Wei and D. Estrin. A comparison of multicast trees and algorithms. Technical Report USC-CS-93-560, University of Southern California, 1993.
- [9] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *To appear in Proceedings of ICCCN '94*, 1994.