The Trade-offs of Multicast Trees and Algorithms [1]

Liming Wei

Computer Science Department
University of Southern California
Los Angeles, CA 90089 - 0781
lwei@catarina.usc.edu

Deborah Estrin

Computer Science Department/ISI
University of Southern California
Los Angeles, CA 90089 - 0781
estrin@usc.edu

draft-ietf-idmr-mtree-01.ps

March 24, 1995

Abstract

Multicast trees can be shared across sources (shared trees) or may be source-specific (shortest path trees). Inspired by recent interests in using shared trees for interdomain multicasting, we investigate the trade-offs among shared tree types and source specific shortest path trees, by comparing performance over both individual multicast group and the whole network. The performance is evaluated in terms of path length, link cost, and traffic concentration.

We present simulation results over a real network as well as random networks under different circumstances. One practically significant conclusion is that member- or sender-centered trees have good delay and cost properties on average, but they exhibit heavier traffic concentration which makes them inappropriate as the universal form of trees for all types of applications.

Status of This Memo

[1] This work has been supported by a grant from Sun Microsystems inc, Mountain View, California, and by NSF grant CDA-9216321.

1 Introduction

Multimedia communication is often multi-point and has contributed to the demand for multicast support. moreover, multimedia applications are often high bandwidth and delay sensitive. Therefore it is essential to consider the efficiency and quality of multicast distribution trees. The problem of computing the optimal multicast path, in the shape of a tree or a group of trees, has many potential solutions. Traditional IP multicast protocols have solely used source rooted shortest path trees, not because it is the best bandwidth-saving strategy, but based on the fact that today's multicast applications are primarily small scale and local area. Factors such as protocol simplicity and convenience of development have dominated protocol design activities. In the context of large-scale, wide area networking, where resources are not as plentiful as in the local area network, the previously-ignored differences between tree types may lead to significant differences in cost and performance of multicast services. However, to date there have not been systematic comparisons among the different solutions.

In this paper, we judge the quality of a tree according to the following three dimensions:

1. **Low delay**. The delay of a multicast tree is evaluated in terms of the end-to-end delay between a source and receiver, relative to the shortest unicast-path delay between the same source and receiver.

2. **Low cost**. There are two different costs associated with a multicast tree:

   (a) Cost of total bandwidth consumption .

   (b) Cost of tree state information.

   In this paper, we only deal with 2a, and leave 2b to future analysis of protocol overhead.

3. **Light traffic concentration**. When a multicast group establishes its delivery trees across the network, traffic from different sources may share links that are not shared when each source uses its own shortest path tree. We compare the maximum number of flows[2] on a unidirectional link as a simple measure of the degree of traffic concentration.

When using the above evaluation criteria, we bear in mind the restrictions of real world large scale networks, e.g. limited knowledge of global topology at each network node and need for algorithms that work efficiently in a distributed manner.

The types of trees can be roughly divided into source specific shortest path trees and group shared trees. In the former tree type, a shortest path tree rooted at each sending source needs to be established; in the later type only one tree shared by all sources within the multicast group needs to be maintained. In the past, the vast majority of research on tree types have been about shared trees. One common class of shared trees under frequent investigation are the Steiner minimal tree [1], and its suboptimal approximations [2] and variances dealing with constrained delays[3]. The main objective of these algorithms was to achieve optimal cost. The other class of shared trees is *center-based* tree. These were mainly introduced in response to other protocol design requirements, such as reduction of setup state information and the need for protocol rendezvous mechanisms [4, 5]. In this paper we investigate the delay, cost and traffic concentration characteristics of various tree types. Our work encompasses two fields: the theoretical algorithms field where most group shared tree algorithms' researches have been carried out and the network protocol design field where real world protocols have been designed and studied. A review of the related backgrounds is necessary.

[2] We call a stream of packets on a link, originated from a particular source, a flow.

1.1 Background

As background, in this section we discuss shortest path tree, Steiner minimal tree and traffic concentration.

1.1.1 Shortest Path Trees

A Shortest Path Tree (SPT) rooted at the source is composed of the shortest paths between the source and each of the receivers in the multicast group. Multicasting eliminates duplicate data packet copies that would otherwise traverse those links that are common to two or more of the source-to-receiver shortest paths. However, a SPT algorithm does not attempt to minimize the total cost of distribution.

Source-rooted shortest path trees are easy to compute, and can be implemented in a distributed fashion efficiently. In networks consisting of symmetric links or paths, reverse path forwarding (RPF) algorithms can be used to derive shortest path trees from the unicast routing mechanisms[6, 7, 8]. When asymmetric paths exist RPF will provide *reverse* shortest paths[3], or distributed link state protocols such as MOSPF can be used to compute shortest path trees[9].

Although not offering minimal cost paths, protocols based on shortest path trees have been adopted most widely [10, 11]. This is due to the fact that when compared with multiple unicast transmissions, SPT-based multicast already provides substantial savings in link cost, and it helps to avoid fan-out problems at sources. For a (virtual) network such as the MBONE [11] with relatively few globally-active multicast groups, SPT's are satisfactory. This is because the network is not rich in connectivity and therefore different types of trees would be mapped to the same routes anyway. Perhaps most important is that to date the control aspect of protocols, instead of the type of distribution trees, dominates the efficiency. However, to support increasing usage and large scale applications there is a need for protocol designers to explore properties of shared-tree types. The next subsection discusses the minimal cost shared tree type.

1.1.2 Steiner Minimal Tree

A Steiner Minimal Tree (SMT) is defined to be the minimal cost subgraph (tree) spanning a given subset of nodes in a graph [12, 2]. Since the SMT for all sources within a multicast group is the same, irrespective of the role of sender or receiver, the number of state entries needed to maintain the tree is only 1 per group. Thus it scales well for big multicast groups with large numbers of senders. The Steiner minimal tree problem has been studied intensively in the area of algorithms for the past half century.

It is well known that computation of a SMT is NP-complete, and is not expected to have polynomial time solutions [1]. This computational complexity prohibits on-demand computation over a reasonably-sized graph. Karp has suggested several techniques to reduce the problem size for SMT computations [1]. However these graph reduction techniques are highly graph dependent. For computer networks as complex as the Internet, where the average node degree (of routers) is higher than 3, graph reduction will not be effective enough in reducing the computational demand to a practical range.

Because of the difficulties in obtaining SMT in larger graphs, it is often deemed acceptable to use near optimal trees instead of SMTs. Various near optimal algorithms exist that produce good approximations to SMT [2]. As will be discussed later, no existing SMT algorithms can be easily applied in practical multicast protocols designed for large scale networks, *but* SMT itself does provide the absolute limit on the minimum link cost and serves as a good reference to measure the cost-optimality of other alternative tree types, such as shortest path trees.

In addition to computational overhead, the worst-case maximum end-to-end path length of a SMT can be as long as the longest acyclic path within the graph. Although the worst case scenario is unlikely

[3]Reverse shortest paths will have higher delay than forward shortest paths because the data follows the shortest path *from* the receiver instead of the shortest path *to* the receiver.

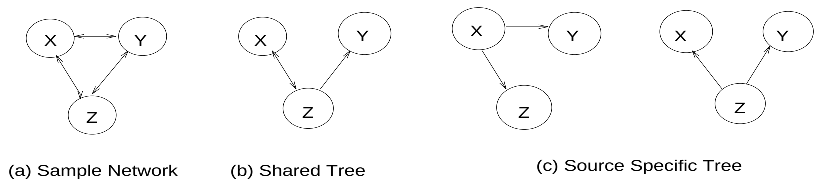


Figure 1: Traffic concentration example

in networks with the rich connectivity typical of today's networks, it is important to know in the *average* case how good or bad a path length along a SMT can be. In section 3.2, we will present simulation results of a near optimal SMT algorithm over random graphs.

1.1.3 Traffic Concentration

Although traffic concentration determines the effective network capacity for multicast applications, the problem has received less attention than the delay-cost tradeoffs in the area of multicast routing[7]. A traffic concentration example is illustrated in figure 1. The network is a simple three-node fully connected graph, where all node pairs are connected by symmetric links in opposite directions. There is a multicast group consisting of 3 receivers on nodes X, Y and Z, and two sources X and Z sending traffic at unit rate 1. Fig 1(b) shows a shared tree used by all senders of the group. Fig 1 (c) shows source-specific shortest path trees. In fig 1(b), link Z—→Y has a load of 2 flows, while in fig 1(c) all links have a maximum load of 1 flow. In Section 3.2 we present simulation studies of traffic concentration in large graphs with many groups.

In summary, minimal cost and minimal delay cannot both be achieved with any single type of tree. With respect to delay and cost, shortest path trees (SPT), which are source rooted, provide minimal delay at the expense of cost; whereas Steiner minimal trees (SMT), shared per group, minimize cost at the expense of delay. Between them are a spectrum of different types of trees offering different trade-offs. In addition, different algorithms use different strategies to place the routes, and may result in different degrees of traffic concentration.

The rest of this paper is organized as follows: Section 2 presents a few typical candidate polynomial time shared tree algorithms; Section 3 describes the network topology and random network model we used in our study; and Section 3.2 presents and analyzes our simulation results.

2 Polynomial Time Algorithms for Group Shared Trees

Group shared trees are used in two proposed mechanisms for scalable multicast routing[4, 13]. Here we briefly discuss related work in polynomial time shared-tree computation algorithms. We will present comparisons based on simulations of the relevant schemes in section 3.2. We divide these algorithms into two major categories: Pseudo Steiner Tree algorithms, and Center Based Tree algorithms.

2.1 Approximations of Steiner Minimal Tree

As mentioned before, the Steiner Minimal tree is a NP-complete problem. As a reference example we choose the algorithm invented by Kou, Markowsky and Berman (referred to as the KMB algorithm) [14] to approximate SMTs. It has been estimated that the cost of a tree generated with the KMB algorithm averages 5% more than the cost of a SMT [15]. However, the KMB algorithm in its original form needs the complete network topology, and therefore is not practical for large wide area internets.

There exist distributed versions of KMB, such as the one proposed in [16] where each node only needs partial knowledge of the network topology. However the required message complexity would significantly complicate protocol design. This issue, in addition to convergence and stability problems, makes it impractical for today's internetwork environment.

Doar & Leslie's Naive algorithm for construction of routes for dynamic multicast groups computes the multicast route by combining the shortest paths across initial multicast group members, then joining new members to the nearest attachment point on the existing tree [15]. Their simulation result showed that the cost of the naive trees is within 1.5 times that of the KMB trees, and their maximum path length is around 50% - 60% of the KMB trees'.

Although this particular algorithm requires complete knowledge of the network topology, it may be possible to modify it for use without global knowledge. However, before considering it as a candidate for real networks, one question that needs to be answered is how it compares with the widely used shortest path trees? The answer can be derived from the comparison of KMB trees and shortest path trees which we present in Section 3.2.

2.2 Center-Based Trees

To cope with the unbounded delay problems of the (near) optimal Steiner trees, Wall proposed several center-based tree algorithms. The simplicity of this class of algorithms makes it desirable for the design of practical protocols and was used as the basis for the Core Based Tree interdomain multicast routing protocol [4], as well as for the shared tree mode of another inter-domain multicast proposal called PIM [13].

Center-Based Tree, as the name indicates, uses a shortest path tree rooted at a node "in the center" of the network [16]. Wall proposed two major strategies for locating centers: (1) Choose the optimal network node so that the resulting center tree can have minimal maximum-delay or average-delay among all group members and senders; (2) Optimally choose a member or sender of the group as the center so that the tree has minimal maximum-delay or average-delay among all member/sender centered trees. We denote the former category delay-optimal CBT[4], the later delay-optimal MSPT (for Member-/sender-rooted Shortest Path Tree).

In this paper, we also consider a third center placement strategy: the minimal cost center placement. Such a tree has the minimal sum of tree-link costs. We call this cost-optimal CBT, or if the center is at a member or sender, cost-optimal MSPT.

It is proven in [16] that the maximum delay of an optimal maximum-delay center based tree is bounded at 2 times that of the maximum delay in shortest path trees. It is also shown that for optimal maximum-delay MSPT, the average of the maximum delays over all senders turns out to be less than 3 times the average of maximum delays along the shortest paths.

The fact that there exists delay bounds for these kind of trees is encouraging. However, for practical purposes, what we really want to know is not only the worst case bound, but also the *average* case maximum delay of such a tree, and the distribution of such delays. Another unknown factor is the cost of such trees, are they cheaper than source rooted shortest path trees for packet delivery?

Note that it is difficult to use CBTs in their original form in real multicast protocols, because (1) finding the center for a group is an NP-complete problem, and (2) it requires knowledge of of the whole topology. Alternative practical forms can be based on heuristic center placement strategies [4], or use MSPT instead. The simulation results in section 3.2 will show whether MSPT is a viable alternative.

[4] For formal definitions of these tree types, see appendix.

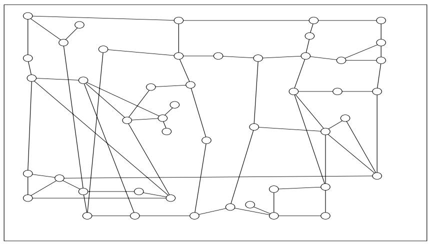


Figure 2: The logical topology of the early ARPAnet

Finally to consider any algorithm for practical application, we need to know how evenly it distributes the routes, i.e. are there serious traffic concentration problems that reduce network utilizations? Again, simulation will be used to answer these questions.

3 Network Topologies

We first simulate the SPT, KMB and different center based trees using the topology of the early ARPAnet, shown in figure 2. This network has 47 nodes. The average node degree is 2.89 — relatively low because of the scarcity of link resources. As the network connectivity improves, the average node degree will be higher.

We then use simulations over different classes of random graphs to capture the comprehensive characteristics of the algorithms and trees. We adopted the random graph model introduced in [17] which can generate a variety of different graphs with classifiable features : connectivity degrees; different edge distributions. One advantage of this model over a purely random model is that it can more easily be correlated to real world networks[5].

In this graph model, graphs are generated in two steps: node coordinates assignments and edge additions. The n vertices are randomly distributed over a rectangular coordinate grid, and are assigned integer coordinates. Edges are introduced according to the edge probability function which takes a pair of nodes (u, v) as its variables:

$$P(u,v) = \beta \cdot e^{\frac{-d(u,v)}{L\alpha}} \tag{1}$$

where $d(u,v)$ is the distance from node u to v, L is the maximum shortest path distance between any pair of nodes in the network (often called network diameter), $1 \geq \alpha > 0$, $\beta > 0$ [6]. A larger value of α increases the ratio of the number of long edges vs. short edges, and a bigger β results in a larger average node degree of the whole graph. Figure 3 illustrates the effect of using different values of α, it shows two 20-node graphs of the same average node degree and under the same node placement.

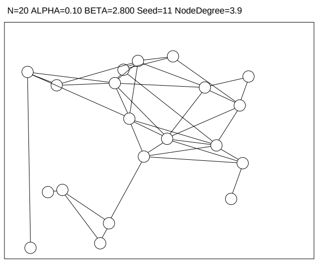
We assign the delay of a link to be the distance between the two end nodes in the simulations.

3.1 Experimental design

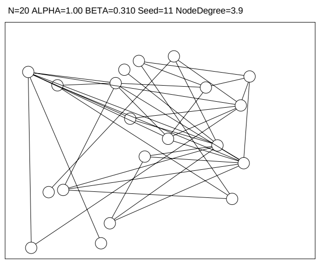
Because SPTs have already achieved much success in practical protocol designs, we use the ratio of each measurement on a certain tree vs the measurement over corresponding SPTs, as the metric for

[5]According to results from [18] when the size of random graphs is big enough, most graph properties will hold when the graph size grows even bigger. We repeated our experiments with two different graph sizes and found the measured results are consistent.

[6]The original model has restriction of $\beta \leq 1$. We found that larger values of β beyond 1, when combined with appropriate small α values, also generate graphs that, subjectively at least, appeared to be of practical significance. This observation was made through a X-window based Random Topology Generator/Previewer that we developed to help visualize different graphs.



(a) $\alpha = 0.1$



(b) $\alpha = 1.0$

Figure 3: Random graphs of different α's

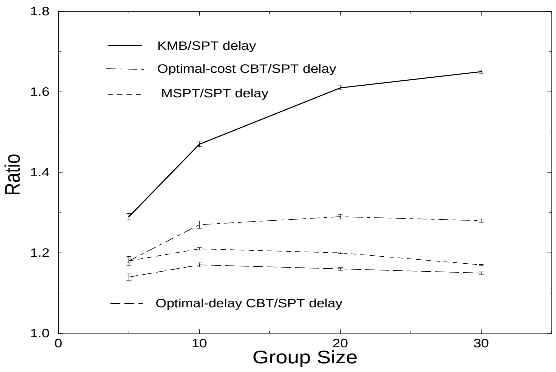


Figure 4: Delay ratios of KMB, CBT and MSPT in ARPAnet topology

comparisons among different algorithms. In the following presentation, we use MaxD to denote the maximum delay experienced from any source to any receiver along a specific tree, and R_MaxD to denote the ratio between MaxD of that particular tree and the MaxD of corresponding SPTs; AveD denotes the average of the maximum delays of all sources, and R_AveD is the respective AveD ratio to SPTs; Cost_ratio denotes the ratio between the sum of link cost of a particular tree and the sum of link cost of the corresponding SPT. The appendix gives formal definitions of these terms.

We constructed 2 sets of experiments, one to to measure the costs and delays of trees, the other to measure the traffic concentrations with different tree types.

In the first set of experiments, we compare KMB trees with SPTs, then center based trees with SPTs. Comparison of SPT and KMB trees can reveal how much more room for link cost savings we may have beyond SPTs, and how much more link delays we should expect if such cost savings are achieved using KMB trees. Comparisons between SPT and center based trees illustrate how much more delay will be incurred by center based trees, on average; and whether they use more or less bandwidth resource than the shortest path trees.

Since center based trees are the only practical candidate for constructing group shared trees in real world protocols [16, 5], we only compare traffic concentrations of center based trees with SPTs.

The parameters that could affect the performance of different distribution tree types are: (1) Reasonableness of graphs, i.e. the proportion of short links vs. long links, which was suspected to have influence over the delay and cost of trees;[7] 2) graph node degree; (3) multicast group size (number of receiver members in current group); (4) number of sources sending to the group (5) Distribution of sources and receiver members; and (6) graph size.

[7] We say a graph is more reasonable if there is a higher probability for a node to be connected to a near neighbor than to a distant neighbor

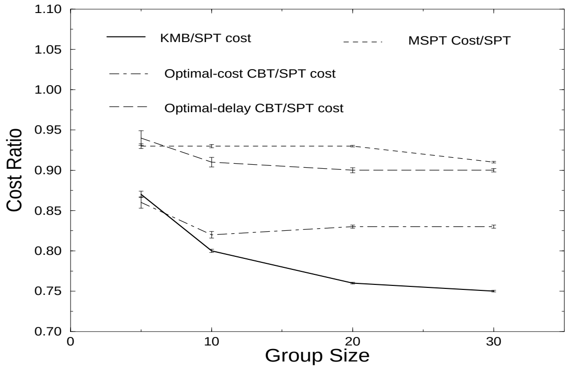


Figure 5: Cost ratios of KMB, CBT and MSPT in ARPAnet topology

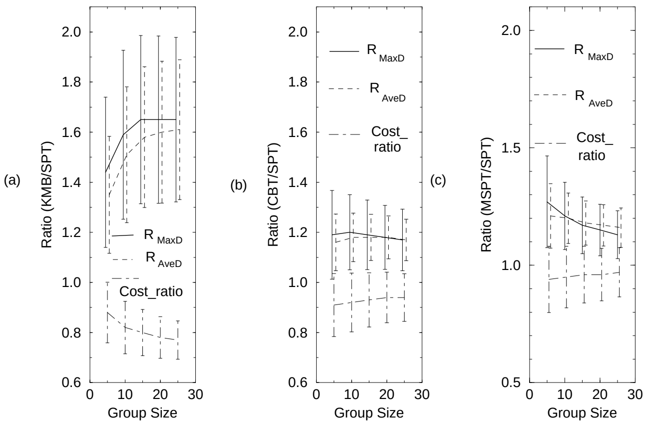


Figure 6: Effect of group size on delays and costs, in 50-node random graphs: (a) KMB; (b) CBT; (c) MSPT

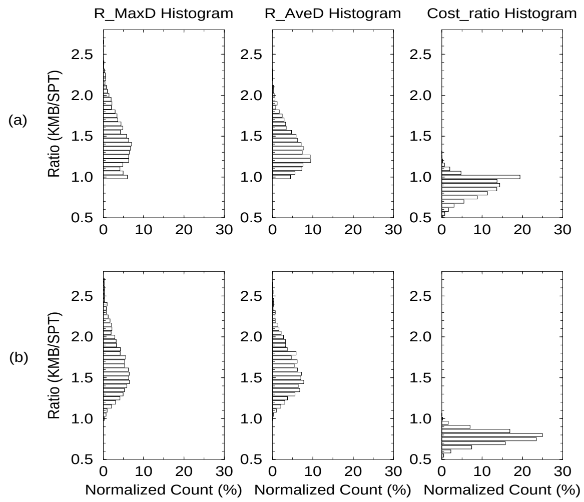


Figure 7: Histograms of delays and costs for KMB trees: (a) Histograms for group size of 5; (b) Histograms for group size of 25

3.2 Simulation Results

Most experiments involving random graphs were run over graphs of two different sizes: 50 nodes and 200 nodes. In the delay and cost comparisons, for each random graph generated, a randomly selected multicast group was put in the graph, then the shortest path trees, the center based trees (CBT and MSPT) and the KMB trees were computed. We assume each multicast group member can take the role of both sender and receiver. Figure 4 through fig 8 are the results for the delay and cost comparisons. In these pictures, there are 500 runs at each data point. Figure 9 and fig 10 are the results for traffic concentration experiments. There we also investigate situations where the numbers of senders and receivers are not equal.

3.2.1 Delay and Cost: SPT, KMB, CBT and MSPT

We first vary sizes of the groups and see how the performance changes. Figures 4 and 5 show the sample means of delay ratios and cost ratios of KMB/SPT, CBT/SPT and optimal-delay MSPT/SPT, in the ARPAnet topology[8]. The group size is varied from 5 to 30. For each group size, we randomly generate 500 groups and compute the different types of trees. The error bars represent the 99% confidence intervals of the means. In figure 4, the average delays of KMB is significantly larger than the other types of trees, especially when group size gets larger. The average delays of optimal cost CBT are above the optimal-delay MSPT, which in turn, are above the optimal-delay CBT. In figure 5, KMB has the minimal average cost, the optimal-cost CBT is the second lowest. Optimal-delay CBT and optimal-delay MSPT being close to each other, have the highest average cost in this comparison. Since it is difficult to adapt optimal-cost center tree algorithms to an efficient distributed algorithm, real world protocols would choose a form closer to optimal-delay CBT or MSPT. Therefore, we will omit results about optimal-cost center trees, and only show results about optimal-delay center trees. Unless otherwise specified, CBT and MSPT refer to optimal-delay CBT and MSPT respectively.

We then repeated the same experiment over a large number of random graphs. Figure 6 shows the effect of group population changes on the KMB and center tree delays and costs. The results were taken

[8]We omitted the average delays of all receivers here. The relationship among different trees' average delays are similar to that of the plotted average-maximum delays.

from 50-node graphs where average node degree is restricted to 4 plus or minus 0.5. In this figure and all subsequent figures, the error bars represent the standard deviations of the data set, to give an indication how wide the measured data is distributed.

Histograms are plotted in figure 7 to show the distributions of the three parameters $MaxD$, $AveD$, and $Cost\_ratio$ at group sizes of 5 (figure 7 (a)) and 25 (figure 7 (b)). For space reasons, we only include the KMB histograms here. The bin size is fixed at 0.05. Note that the horizontal axis of the histograms are normalized. E.g. a bar of length 20 with ratio = 1.1 means 20% of the measured data values are at ratio of 1.1 (to $1.1 + 0.05$).

It can be observed from figure 6 that the delays of KMB trees tend to grow larger with larger groups, while their costs in comparison with shortest path trees tend to be lower. The delay and cost curves of center trees, however, are rather flat. With larger groups, center trees tend to have shorter delays than KMB trees, but the peaks of their cost_ratio histograms are higher than that of the KMB trees (not shown here).

KMB trees in general have bigger variations in delay than center trees. The tails of R_{MaxD}, R_{AveD} in fig 7 (a) and (b) extend to about 2.4, while in corresponding histograms for CBTs and MSPTs the tails are below 1.7, and larger groups have slightly shorter tails. The above observation exhibits an interesting aspect of the fate-sharing nature of center trees: They are not optimal for everyone, and overall, they are also not very bad for everyone either.

We repeated the above experiment in 200-node random graphs, with group sizes ranging from 20 to 80. The results have similar trends. KMB delay curves are higher, the average ratio of delays is around 2. The KMB to SPT cost ratio curve is lower (by about 0.2). CBT and MSPT delay curves, though slightly higher, are not significantly different than in 50-node graphs.

Now we try to vary the *node degree* of a graph and measure the performance. Figure 8 shows the effects of different node degrees on the delay and cost properties. All graphs have 50 nodes, all multicast groups have 10 members. When processing simulation data, the average node degrees are rounded to the nearest integer. The vertical axe are the averages of R_{MaxD}, R_{AveD} and $Cost\_ratio$ respectively. Solid lines represent graphs with $\alpha = 0.2$, dotted lines represent graphs with $\alpha = 0.6$. A few observations that can be made from this picture are:

1. All algorithms are relatively insensitive to the reasonableness of graphs;

2. The maximum delay within a KMB tree tends to be larger than that of a CBT tree. The maximum delay of a KMB tree increases faster when the average node degree increases. Note that MSPT R_{MaxD} and R_{AveD} curves are quite close to those of CBT's;

3. The KMB $Cost\_ratio$ curve decreases faster. MSPT $Cost\_ratio$ is close to CBT $Cost\_ratio$.

Observations 2 and 3 above suggest that, in the range of graphs experimented, MSPT will on the average be almost as good as the optimal CBT.

The above experiments were rerun over 200-node graphs, also with 10 member groups. The R_{MaxD}, R_{AveD} and $Cost\_ratio$ all have similar trends; KMB R_{MaxD} and R_{AveD} are slightly higher (by 0.2), and KMB $Cost\_ratio$ curve is slightly flatter. The changes in MSPT and CBT curves are very small. Therefore at higher node degrees, the differences between KMB R_{MaxD} and CBT R_{MaxD}, and the differences between KMB R_{AveD} and CBT R_{AveD}, are slightly smaller than in 50-node graphs.

3.2.2 Traffic Concentration: SPT, CBT and MSPT

In the traffic concentration experiments we use a number of fixed size multicast groups randomly placed in a graph and part of the group members are assigned the roles of senders randomly. Assuming each source of the group generates traffic at constant unit rate, the total number of unit traffic flows that

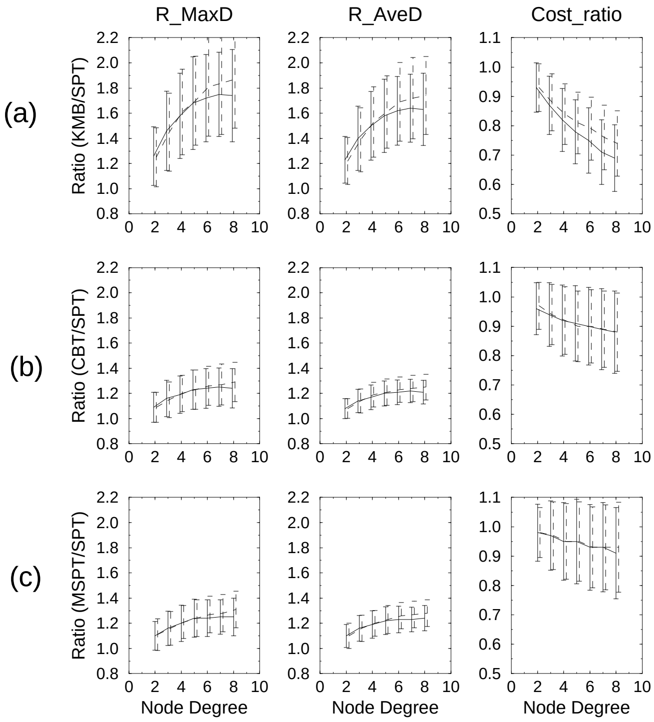


Figure 8: Comparisons of delay and cost in 50-node graphs: (a) KMB tree vs SPT, (b) center tree vs SPT, (c) MSPT vs SPT

traverse a link is counted[9]. [10] For the same graph and groups, center based trees and shortest path trees are computed and link loads counted.

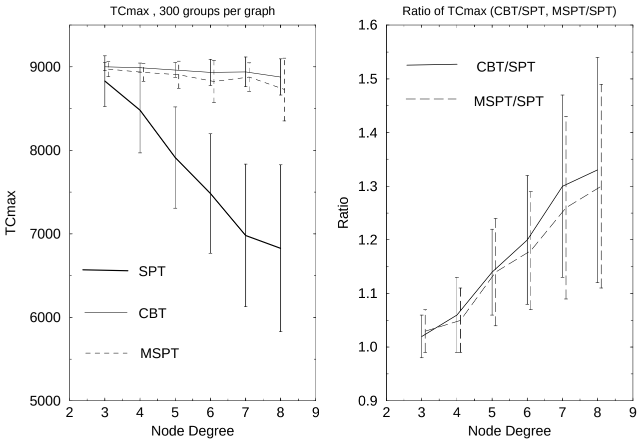
The maximum link loads for SPT, CBT and MSPT are shown in fig 9 (a). Figure 9 (a) shows that as the average node degree of the random graphs grows from 3 to 8, the SPT's maximum link load decreases. This is because there are more redundant links when average node degree is higher, and there exist alternate paths between many pairs of nodes. However, center based trees' maximum link loads hardly change when the average node degree increases, which is not surprising considering the fact they are center or single-source-rooted shortest path trees. Fig 9 (b) shows the ratio of maximum link loads of CBT and MSPT vs. that of SPTs. We have run the same experiment under identical configurations but with 2 senders per group, the results are comparable.

This experiment suggests that in networks with low connectivity degree, the link utilization pattern of center-based trees will be very close to that of shortest path trees. However, when the average node degree increases, center-based trees maintain almost flat maximum link loads, whereas the maximum link load of shortest path trees drop significantly. This indicates that added alternative paths are not being sufficiently utilized by center based trees.

To see what proportions of links are highly loaded under different algorithms, we counted the number of links under different loads. Figure 10 shows the distribution of link loads within one specific 50-node graph. The average node degree is 4.8, there are 300 active groups all having 40 members. There are 20 senders within each group. When the number of senders increases from 2 to 20, with all three types of

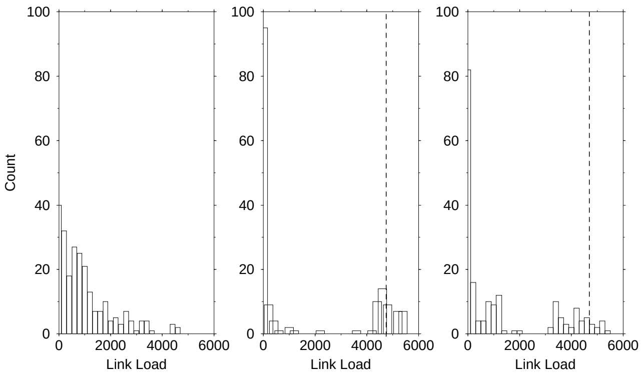
[9] Often, senders within the same multicast group have similar sending rates, but senders for different groups may have different sending rates. The purpose here is to show tendencies of route distributions, thus such uniform sending rate assumption would suffice.

[10] Each physical link is treated as two unidirectional links connecting the same pair of nodes in opposite directions.



(a) (b)

Figure 9: Traffic Concentration, in 50-node graphs with 300 40-member groups, 32 senders per group



(a) SPT (b) CBT (c) MSPT

Figure 10: Distribution of link loads in the same graph

trees, the distributions of maximum link load roughly retain the same shape except that the lengths of the tails extend to about ten times those in 2-sender cases. Only results from 20-sender experiments are shown in figure 10.

In figure 10 the SPT histogram profile drops smoothly with larger link loads — more links are lightly loaded and smaller number of links have high loads. The CBT and MSPT histograms both have a narrow high peak at near 0, representing links under-utilized. The peak is followed by a long tail with a rather significant portion at the end of the tail. If we draw a vertical line in the two right pictures in fig 10 at the position equal to the maximum link load of the corresponding SPT link load histogram (as shown in fig 10 (b) (c) in dotted lines), the area to the right of the line represents the number of links that needs higher link capacities to service the same configuration of multicast groups as in their SPT counterpart. The link load distribution is closely related to the distribution of member locations of the multicast groups. Changing the set of the multicast groups may change some details of the histograms, but the profiles shown in fig 10 remain relatively constant under random distribution of multicast groups.

This suggests that center based trees may not be ideal for high bandwidth applications, which after the multiplying effect would create hot spots and reduce the effective number of traffic flows that can be admitted into the network.

4 Conclusions

SPT and shared tree (SMT & center based) approaches are complimentary in terms of algorithm complexity, link costs, delays and traffic concentration densities. SPT offers minimal delay and is the simplest to compute. SMT offers minimal cost and is the most difficult to compute. Center based tree falls in between these two extreme cases.

SMT delays are not bounded in theory, simulation of KMB trees resulted in widely distributed values. It is not likely that algorithms derived from pseudo-optimal SMT algorithms such as KMB, can be used in real multicast protocols.

Center based tree delays are not adequately bounded, but are (mostly) favorably distributed. MSPT delays and costs are about the same as center based trees with delay-optimal center placement. MSPT and delay-optimal CBT also have similar traffic concentration characteristics. If the use of delay-optimal center based trees is justified in a practical protocol, it will be sufficient to use MSPT which is significantly easier to compute than optimal center based trees.

We have shown that the performance of these different trees is sensitive to group population, average node-degree and locations of the group members. It is relatively insensitive to the "reasonableness" of the edge distributions of a graph.

Our simulations showed that different algorithms indeed lead to different degrees of traffic concentrations. Hence selection of algorithms should not be based purely on performance for individual groups. When there exists heavy traffic concentration, the heavily loaded links become bottlenecks. Although source specific SPTs consume more link bandwidth for each individual multicast group, their demands on bandwidth are more evenly distributed than the center based trees, especially in networks with high connectivity degree. Hence a network may support more high bandwidth multicast groups if SPTs are used instead of center based trees (or MSPTs).

5 Acknowledgments

We would like to thank Robert Braden, Stephen Casner, Robert Felderman, Yakov Rekter, Anthony Li, the people on the idmr mailing list and many other ISI division 7 members for comments and suggestions on various subjects of this paper.

References

[1] R. M. Karp. *Reducibility among combinatorial problems.* Plennum Press, New York, 1972.

[2] Pawei Winter. Steiner problem in networks: A survey. *Networks*, 17(2):129–167, 1987.

[3] G. Polyzos V. Kompella, J. Pasquale. Multicasting for multimedia applications. In *Proceedings of the IEEE Infocom'92*, 1992.

[4] A. J. Ballardie, P. F. Francis, and J. Crowcroft. Core based trees. In *Proceedings of the ACM SIGCOMM*, San Francisco, 1993.

[5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. An architecture for wide-area multicast routing. In *Proceedings of the ACM SIGCOMM 94*, London, September 1994.

[6] Y. K. Dalal and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21(12):1040–1048, 1978.

[7] Steve Deering. *Scalable Multicast Routing Protocol.* PhD thesis, Stanford University, 1989.

[8] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, pages 85–111, May 1990.

[9] J. Moy. Multicast extensions to ospf. *RFC 1584*, March 1994.

[10] Ron Frederick. Ietf audio & videocast. *Internet Society News*, 1(4):19, 1993.

[11] Steve Casner. Second ietf internet audiocast. *Internet Society News*, 1(3):23, 1992.

[12] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, January 1968.

[13] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (pim), sparse mode protocol: Specification. *Working Draft*, March 1994.

[14] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.

[15] Matthew Doar and Ian Leslie. How bad is naive multicast routing. In *Proceedings of the IEEE Infocom'93*, 1993.

[16] David Wall. *Mechanisms for Broadcast and Selective Broadcast.* PhD thesis, Stanford University, June 1980. Technical Report N0. 190.

[17] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9), December 1988.

[18] Bela Bollobas. *Random Graphs.* Academic Press, Inc, Orlanndo, Florida, 1985.

Formal Definitions

We use graph theory notations to define the performance measures of a multicast tree. Let $G = (V, E, C)$ be a directed graph, where V is a set of nodes, $E = \{(u, v)|u, v \in V\}$ is a set of edges, $C = \{c(u, v)|(u, v) \in E\}$ is a set of edge costs. Let $M \subset V$ be the set of multicast members, $S \subset V$ be the set of senders for M, $T_M(u) = (V_M, E_M, C_M)$ such that $T_M(u) \subset G$ be a multicast tree for M which source u uses for packet delivery, T_M be the set of a specific kind of multicast trees for all sources in S for M, and $d(u,v,T_M(u))$ be the path length from u to v via tree $T_M(u)$. We define the performance measures as follows:

1. Maximum and average delay measures[11]. First, the maximum delay for source u along tree $T_M(u)$ is,

$$\begin{aligned} maxD(T_M(u)) &= MAX\{d(u,v,T_M(u)) \mid \\ &\quad \text{for all } v \in M\} \end{aligned} \tag{2}$$

   the maximum delay for all sources in S for group M,

$$\begin{aligned} MaxD(T_M) &= MAX\{maxD(T_M(u)) \mid \\ &\quad \text{for all } u \in S\} \end{aligned} \tag{3}$$

   and the average of the maximum delay for multicast group M,

$$AveD(T_M) = \frac{1}{n}\sum_{u \in S} maxD(T_M(u)),\ n = |S| \tag{4}$$

   For convenience of comparison, we normalize the delays across different graphs, and use R_{MaxD}, the ratio of maximum delays and R_{AveD}, the ratio of average maximum delays,[12]

$$R_{MaxD}(T_M) = \frac{MaxD(T_M)}{MaxD(SPT_M)} \tag{5}$$

   where SPT_M is the set of shortest path trees for group M. And,

$$R_{AveD}(T_M) = \frac{AveD(T_M)}{AveD(SPT_M)} \tag{6}$$

   An optimal maximum-delay CBT can be defined as a center tree whose MaxD(T_M) is minimal among all center trees.

2. Link cost for traffic from source u along tree T_M,

$$Cost(T_M, u) = \sum_{(i,j) \in E_M(u)} c(i,j) \tag{7}$$

   For shared group trees, if the links are symmetric (c(i,j) = c(j, i)) and all sources are receivers themselves, $Cost(T_M, u)$ will be the same for all sources u.

[11] Despite the form, this definition is the same as those used by Wall in [16]

[12] Note that we could have defined this ratio as,

$$\begin{aligned} R_{MaxD}(T_M)' &= \frac{MaxD(T_M)}{maxD(SPT_M(u))}, \quad \text{where } u \in S, and \\ &\quad maxD(T_M(u)) = MaxD(T_M) \end{aligned}$$

But this definition may not capture the change of delays in a meaningful way. A source that has the largest maximum delay in T_M may not have the largest maximum delay in a SPT. The alternative definition $R_{MaxD}(T_M)'$ gives the ratio of the maximum path length for an individual source. Definition (5) presents the change in maximum delay for a whole group. The down side of definition (5) is that it does not show the change of fate among the group members in different types of trees.

An optimal cost CBT can be defined as a center tree whose $Cost(T_M, u)$ is minimal among all center trees.

For non-shortest-path-tree T_M, the ratio of link cost for traffic from source u is,

$$Cost\_ratio(T_M, u) = \frac{Cost(T_M, u)}{Cost(SPT_M, u)} \tag{8}$$

where $Cost(SPT_M, u)$ is the link cost of a shortest path tree rooted at u extending to all members of group M. When there are multiple shortest path trees, we pick one at random.

3. Traffic Concentration. Let $num\_flow(i, j)$ be the number of flows passing link (i, j). The maximum link load in a graph G when there are n active groups is,

$$\begin{aligned} TCmax(G, n) = \quad & MAX\{num\_flow(i, j)| \\ & \text{for all } (i, j) \in E\} \end{aligned} \tag{9}$$

The distribution function of all link loads of graph G under n active groups is,

$$Dist(G, i) = \text{number of links with } i \text{ flows} \tag{10}$$