Video Distribution on Multicast Networks

N. F. Maxemchuk

AT&T Labs - Research
Murray Hill, NJ 07974

1. Introduction

Increasingly, the multicast backbone (Mbone) of the Internet[1,2] is being used to provide real time audio and video coverage of events that are of interest to a limited audience. The audience is frequently sparsely distributed over a wide geographical area, making standard broadcast techniques inappropriate. Initially, the facility was used to distribute selected sessions in technical conferences, such as SigComm or ITS[3], and some events of more general interest, such as the video from inside space shuttle flights. However, as the Internet becomes more widely used by the general public, the use of the Mbone for news, entertainment, and distance learning is increasing.

Applications that transmit continuously for an extended period of time use more resources than most current Internet applications. For instance, a one hour video transmission at 256 kbps requires almost a gigabit. This type of application has a greater demand for network bandwidth than electronic mail or most information retrieval functions on the world wide web. In the Mbone, bandwidth is conserved by constructing a tree of shared channels from the source to the receivers, rather than assigning point-to-point channels from the source to each destination. For video transmission, the bandwidth is also reduced by compressing the transmitted signal.

The acceptance of the Mbone for real time applications is in part due to the ability to use existing computing equipment that is connected to the Internet, rather than requiring special purpose transmitters and receivers connected to a new network. As a result, not every receiver has the same capabilities. The processors in some receivers are able to implement more complex decompression algorithms than others, and not every connection to the network has the same bandwidth. This creates a network design problem that is different from any other video distribution network.

Our objective is to to design multicast tree networks that are tailored for the characteristics of Mbone video. The tree networks can either be used to conserve the resources of the Internet or to lease a low cost set of channels in the telecommunications network. Real time video and audio applications require more stringent quality of service guarantees from the network than the traditional data services of the Internet. Techniques that reserve resources on the Internet in order to provide these guarantees is an active area of research[4]. However, at present the telecommunications network is better able to provide guarantees for continuous transmission and would be better suited for a large scale deployment of the technology being developed for Mbone video. In the telecommunications environment, Mbone video is a candidate for a pay-per-view service in which an entrepreneur leases facilities to provide coverage of a new or sports event.

In section 2 we present our model of video multicasting and relate it to earlier work. In our model, designing multicast trees for video distribution is a variant of the Steiner tree problem[5,6]. The conventional Steiner tree problem constructs a minimum cost tree for a subset of the nodes in a network. The link costs in the network are fixed. In our variant of the problem, the link costs are dependent upon the set of receivers that share the link.

In section 3, we describe a heuristic to solve our problem. Our heuristic is a modification of the Takahashi-Matsuyami heuristic[7,6] (T-M heuristic), which is known to provide good solutions for the conventional Steiner tree problem. We establish a common framework for minimum depth routing, Solin's implementation of the minimum spanning tree algorithm, the T-M heuristic, and our modified heuristic. In section 3.3 we emphasize how closely all four procedures are related by providing the pseudo-code of a common implementation. The four techniques differ only in the nodes that are in a target set, and a single multiplicative factor.

In section 4 we evaluate the new heuristic and determine some operating regions where it provides a significant reduction in cost when compared with the other three techniques. In fairness, in section 4.3.2, we also show a region where the heuristic can be misapplied and result in higher cost than the T-M heuristic. We compare the cost of trees generated by the four techniques on randomly generated networks with a wide range of receiver loads.

In section 4.1 we describe a technique for generating random networks. The networks are made more realistic than previously reported random networks by clustering nodes rather than distributing them uniformly. Different clusterings are used for random networks that are intended to look like national and international networks. In a national network there are more nodes in major population centers and in the international network there are more nodes on land masses. Practical techniques that have been used to design Mbone trees implicitly assume that receivers are clustered, as will be explained.

2. Model

In our model of an Mbone video network, a tree is constructed from the source to every receiver. Receivers can request different bandwidth signals from the source. The source transmits only one signal that is sufficient for the highest bandwidth receiver. As the signal traverses the tree it is filtered so that the bandwidth on a link is only the maximum of the bandwidths of the receivers that are connected to the source through that link. This mode of operation is similar to the receiver initiated reservations and packet filtering used in the RSVP protocol[8].

Our objective is to construct a minimum cost tree. In this work, the cost of a link is an increasing function of the bandwidth and the length of the link, and not dependent upon the utilization or availability of the link. This is a common pricing structure for leased lines in the telecommunications network. The assumption is that the network is engineered so that capacity is available for most of the connections that are requested. Other work on designing Mbone trees has used constraints based upon the availability of capacity[9] or delay[10]. Including these factors in the cost function is appropriate in the current Internet because the network is not designed for video services, and the available resources are inadequate.

The trees that we design are for a specific group of receivers. If there are changes in the group, reapplying the design procedures may result in a tree that has very little resemblance to the original tree. Much of the work on designing Mbone trees has been concerned with data connections that are maintained for long periods of time for a dynamic user community[5,11]. The trees are designed to support a wide range of users without extensive modifications in the basic tree. The group of receivers for a scheduled, limited duration, video program, particularly in a pay-per-view environment, is likely to be much more static than the group of receivers in an open ended data collaboration. Assuming a relatively static receiver group allows us to design better tailored, less expensive tree networks.

Our problem depends upon receivers requesting and being able to obtain different bandwidth signals. In the current environment, the receivers are typically computers with a wide range of processing capabilities, possibly augmented by special purpose video processing hardware. As a result, some receivers can implement more complex decompression algorithms, at a higher frame rate or resolution than others. In addition, different receivers have different rate connections into the network. Connections to the Internet range from voice band modems of a few tens of kilobits per seconds for homes, up to *OC* 3 rates of 155 megabits per second for several super computer centers. In a pay-per-view system, pricing can also be used to encourage receivers to limit the demands that they place upon the network. At present, most video broadcasts over the Mbone deliver the same signal to all of the receivers and operate conservatively so that all of the intended receivers can receive and decode the signal. In effect, everyone gets the grade of service of the least capable receivers.

In order to provide each receiver only with the bandwidth that it requests, the bandwidth of the signal must be reduced as it passes through the network. This can be implemented by using a progressive coder[12,13] or by converting between encoding formats. An example of a progressive coder is a Fourier transform coder in which the high resolution components and low resolution components are placed in different packets.

The low resolution signal can be transmitted to all of the receivers and the high resolution components only to those that request them. Similarly, progressive intraframe coders can be designed to deliver 30, 15, or 5 frames per second, by marking the frames and not forwarding all of them along all of the branches. An example of a format converter that changes the bandwidth of a compressed video signal is the Internet video gateway[14] which converts a multimegabit per second JPEG signal to a 128 kilobit per second H.261 signal.

An earlier work on Mbone routing that uses progressive coding, routes different components of the signal destined for a specific receiver over different paths[9]. This work is concerned with a resource constrained environment. When the resources are adequate and the cost to send the components on a path is additive, then the least cost solution is separate Steiner trees for each of the components. For instance, if all of the receivers are sent the low frequency components and half of the receivers are sent the high frequency components, then a Steiner tree is designed with all of the receivers and a second tree is designed with the half of the receivers that are sent the high frequency components. The first tree is the least cost way to send the low frequency components to all of the receivers, and the second tree is the least cost way to send the high frequency components to the subset of receivers. Therefore, the superposition of the two trees is the least cost way to send both components. With this approach, receivers must resequence the packets before decoding the signal because the paths on the different trees may have different delays. The processing available in receivers is frequently a limiting factor, therefore, in the remainder of this work we assume the current, simpler mode of operation, in which the entire signal for a specific receiver traverses the same path.

The techniques that have been used to design low cost, multicast trees for the Internet[2] include minimum distance routing, truncated spanning trees and Steiner trees. In minimum distance routing, the least cost path from the source to each receiver is determined, and links are shared when they are used by more than one receiver. In truncated minimum spanning trees, links are removed from the tree if they are not used by active receivers.

The nodes in the network are switching centers, and can have several receivers. In a shared network, the bandwidth required at a node is the maximum of the bandwidths required by the receivers. If there is at least one receiver at every node in a network, and every node requires the same bandwidth, then the lowest cost tree is a minimum spanning tree. If there is not a receiver at every node, but every node with a receiver requires the same bandwidth, then the minimum cost tree is the Steiner tree. The brute force way to find the Steiner tree is to consider a set of networks, each of which contains all of the nodes that have receivers and a subset of the nodes that do not have receivers. The minimum of the minimum spanning trees for all possible networks of this type is the Steiner tree. This exhaustive search is usually lengthy, particularly when a small fraction of the nodes are receivers. Therefore, heuristics are used that yield low, although not necessarily minimum, cost trees. Reference 6 provides a survey of these heuristics.

The problem that we must solve is related to the Steiner tree problem in that only a subset of the nodes in the network have receivers. However, it is complicated because not every receiver requires the same bandwidth. In the Steiner tree and minimum spanning tree problems, the cost of a link is known and fixed at the beginning of the problem. In our problem, the cost of a link is determined by the requirements of every receiver that uses the path. It is possible that the cost of a link in the tree will change as the tree is designed because a receiver that is added at a leaf has a higher requirement than any of the other receivers that use the link.

3. An Heuristic Algorithm

The heuristic algorithm that we propose to use on our variant of the Steiner tree problem is modification of the T-M heuristic. The T-M heuristic is a combined application of the minimum spanning tree algorithm and minimum distance routing, and can be implemented in a way that is almost identical to Solin's minimum spanning tree algorithm.

In order to explain and provide justification for our heuristic, we will establish a common framework for minimum depth routing, Solin's minimum spanning tree algorithm and the T-M heuristic. In section 3.1.1, 3.1.2 and 3.2.1 we provide the necessary background information the place the existing algorithms and heuristics into the framework, and in section 3.2.2 we present the modified heuristic.

In section 3.3 we provide the pseudo-code for a unified implementation of all four tree generation procedures. The common implementation emphasizes the close relationship of the four procedures. It also shows that the time to execute the modified heuristic is the same order as the T-M heuristic.

3.1 Tree Algorithms

Consider a graph that consists of a set of nodes, links between the nodes, and weights or costs assigned to the links. A class of algorithms have evolved that generates a tree on a graph by starting from a single node and adding one new link and one new node at a time to form a tree. Depending upon the criterion that is used for adding the new node, a minimum depth tree, in which each node is as close as possible to the starting node, or a minimum spanning tree, in which the sum of the link weights is minimized, can be generated.

There are alternative means of generating trees with these characteristics. A well known procedure for generating minimum spanning trees is Kruskal's algorithm (see reference 15 page 67). In Kruskal's algorithm, instead of adding nodes to a single tree, at each step the least weight link that has not been included in a graph is determined. If the link joins two nodes that are in the same tree, it is discarded, otherwise, it is added to the graph. The new link can either join a node to an existing tree, or join two trees together. In the intermediate steps of this algorithm there may be a forest of trees that are later connected.

3.1.1 Minimum Depth Tree In a minimum depth tree, every node is connected to a root node by its shortest path. One way to generate this type of tree, starts by assigning the root node to the tree and setting its depth to zero. At each step in the algorithm the node that is closest to the root, but is not in the tree, is added to the tree. The node that will be added is determined by considering every link, at every node that is in the tree. If a node that is not in the tree is connected to one of the links, then the depth of the node is the depth of the node in the tree plus the cost of the link. If a node that is not in the tree is connected to the tree by several links, its depth is the minimum of the depth calculations. The minimum depth node that is not in the tree is added. The algorithm continues until all nodes that are connected to the original node have been added to the tree.

An example of this algorithm is shown in Figure 1. Node 1 is the root, and the number in brackets is the distance of the node from the origin. In each step, the solid link, and the node at the other end of it are added to the network.

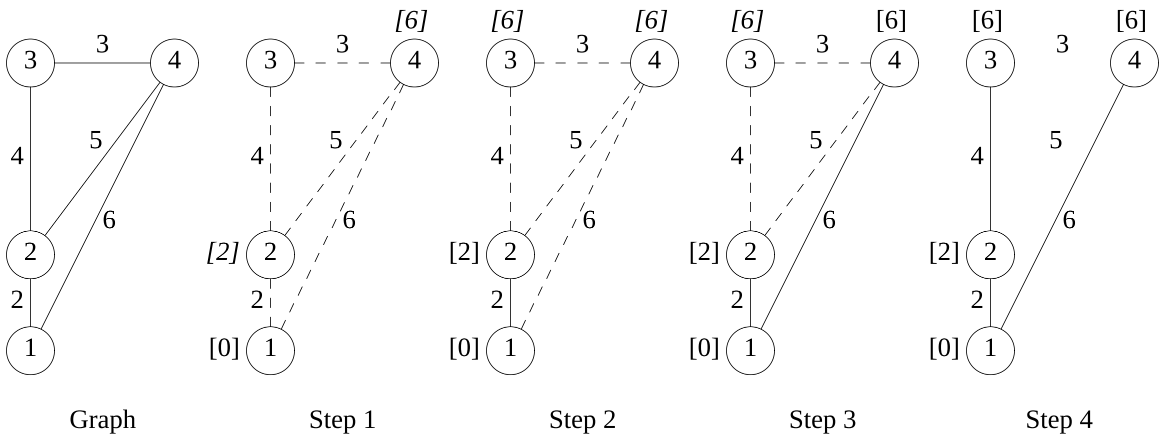


Figure 1. Example of Minimum Depth Tree Algorithm

3.1.2 Minimum Spanning Tree Solin's algorithm for minimum spanning trees operates in a manner that is similar to the minimum depth tree algorithm [see reference 15 page 71]. At each step a single node is added to the tree. The node that is added to the tree is selected by finding the least weight link from a node that is connected to the tree, to a node that is not. Solin's algorithm can be implemented by executing the minimum depth algorithm, but setting the depth of all nodes in the tree to zero, instead of the distance to the origin. An example of Solin's algorithm is shown in figure 2.

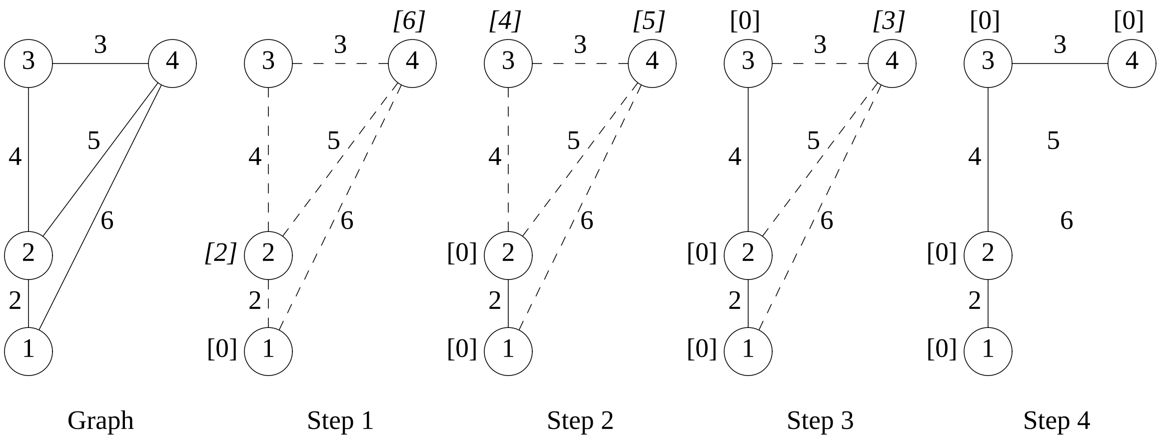


Figure 2. Example of Solin's Minimum Spanning Tree Algorithm

3.2 Steiner Tree Heuristic

The minimum depth tree and the minimum spanning tree do not provide the best solutions to the Steiner tree problem. Consider the network in figures 1 and 2 with node 1 as the source and nodes 2 and 4 as the receivers. The sum of the link weights resulting from the minimum depth algorithm is 8, and from the minimum spanning tree algorithm is 9. The optimum solution to the Steiner tree problem can be found by applying the minimum spanning tree algorithm to all possible networks with different combinations of nodes that are not transmitters or receivers. In this particular example, there are only two such networks, one with node 3, and one without that node. When the minimum spanning tree algorithm is applied to the network without node 3, node 1 is connected to node 2, and node 2 to node 4. The resulting link weights is 7, which is the minimum possible set of link weights.

3.2.1 T-M Heuristic The T-M heuristic operates in a manner that is similar to Solin's minimum spanning tree algorithm. At each step a receiver is added to the tree. The receiver that is added has the shortest path between itself and the currently existing tree, just as the node that is added in Solin's algorithm has the shortest path. The difference between the two procedures is that the path in Solin's algorithm is a single link, allowing a straightforward search, while the path in the the T-M heuristic may contain several links.

The T-M heuristic can be implemented as a combination of the minimum depth tree and the minimum spanning tree algorithms. The nodes that are permanently connected to the tree are assigned a depth of zero, as in our implementation of Solin's algorithm. Initially, only the source is permanently connected. At each step the minimum depth algorithm is applied, and nodes are temporarily connected to the tree, until a receiver is temporarily connected. When this occurs, the links and nodes between the tree and the new receiver are made permanent and each permanent node is assigned a depth of zero. The links and nodes that are temporarily connected, are removed from the tree. This allows us to search for a shorter path from the nodes that were temporarily connected to the old tree and the new permanent tree. The operation of the algorithm with the source at node 1 and receivers at nodes 2 and 4 is demonstrated in figure 3. The dotted lines have not been assigned, the dashed lines are assigned temporarily, and the solid lines are assigned permanently.

In the example in figure 3, the T-M heuristic produces the optimal tree. However, this will not always be the case. Consider the network in figure 4 with the source at node 1 and receivers at nodes 2 and 3. The T-M heuristic results in the tree shown, where the sum of the link costs is 200. The optimum solution, which is also shown, is 153. The T-M heuristic is known to have a solution that is within a factor of two of the optimum, however, in most networks, the performance is much better.

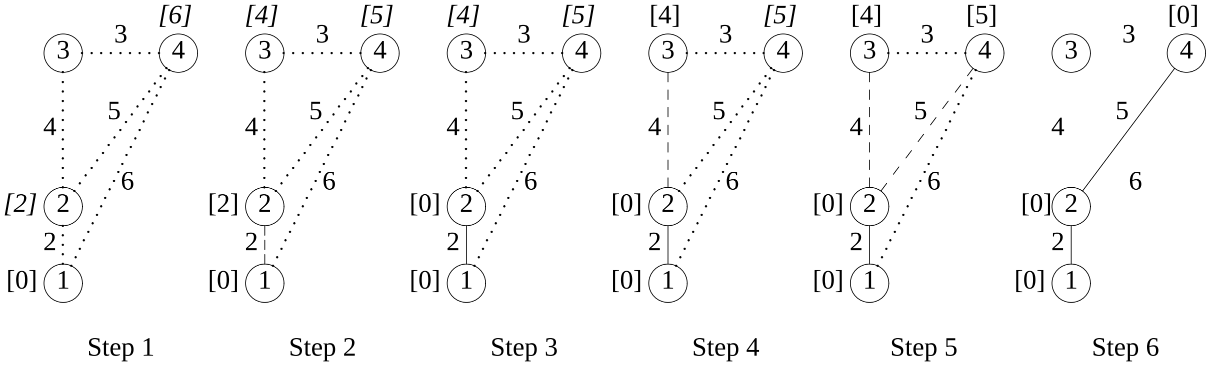


Figure 3. Example of the T-M Heuristic for Steiner trees

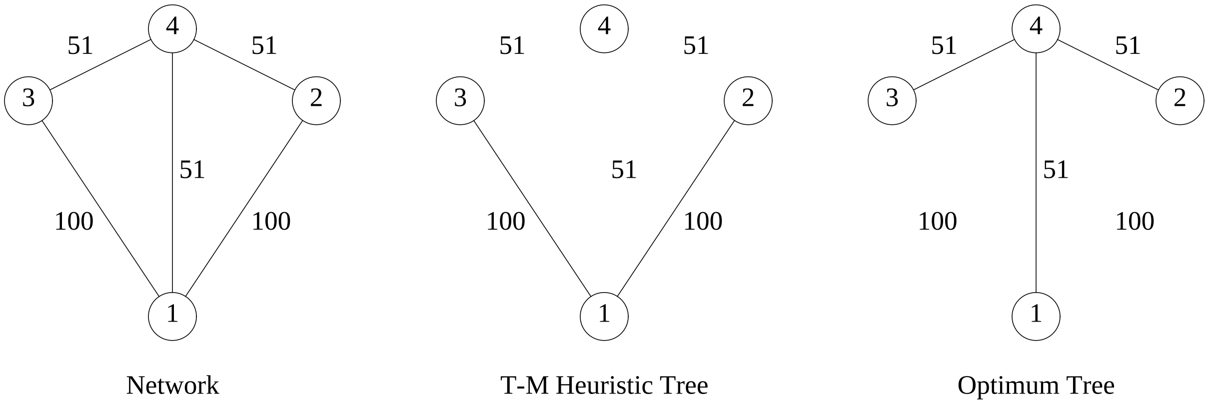


Figure 4. Heuristic tree derived from T-M heuristic and minimum cost tree for a network with a source at node 1 and receivers at nodes 2 and 3

3.2.2 Modified T-M Heuristic An important difference between the problem of video multicasting and the conventional Steiner tree problem is that the cost of a link is not fixed but depends upon the maximum rate of the receivers that share the path. In the T-M heuristic, assume that we add a node, N_1, to the tree and calculate the link costs based upon the rate of that node. If at a later step we add a higher rate node, N_2, to N_1 or any of the nodes that are connected to it, then the link costs for N_1 increase. The true cost of adding N_2 to the tree is the cost of connecting to N_1 plus the increase in cost of the path from N_1 to the root of the tree. With the higher link costs for N_1, we may not have added that node to the tree as early as we had.

A simple network that demonstrates this type of operation is shown in figure 5. The cost of a link is the basic cost of the link, times the rate of the highest rate receiver that uses the link. The cost of the tree generated by the T-M Heuristic is 28 and the minimum cost tree is 26.

When a new receiver is added to the tree by connecting it to a receiver of the same or higher rate, the cost of the links that are in the tree are not affected. When there is a significant difference between the transmission costs of different receivers, it is more important to make certain that the higher cost receivers use less transmission facilities than to make certain that the lower cost receivers use less facilities. These two observations form the basis for modifying the T-M Heuristic. In the modified algorithm, we first form a Steiner tree with the a high rate set of receivers and then successively add lower rate sets of receivers to that tree.

Specifically, the heuristic operates as follows:

1. Separate the receivers into subsets according to rate. In most instances we expect a small number of rates and each subset will consist of the receivers with a specific rate. However, if there are a number of closely related rates, there may be an advantage to putting receivers with a range of rates in the

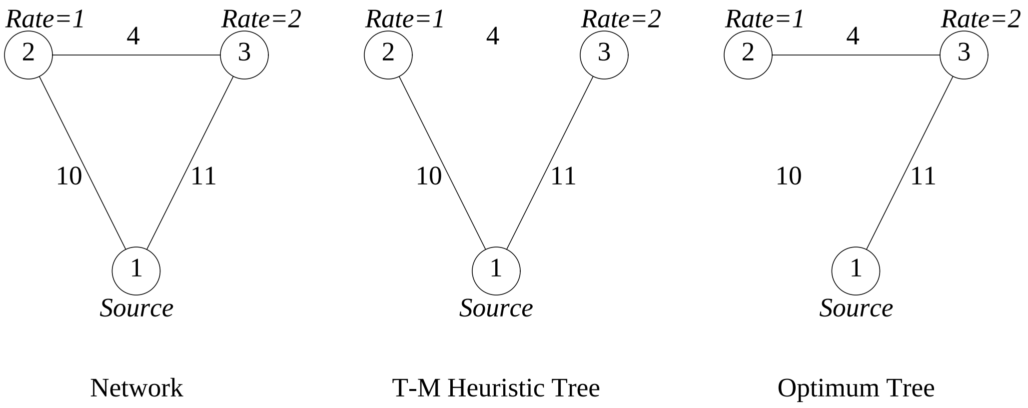


Figure 5. Tree derived from T-M heuristic and minimum cost tree for a network with a source at node 1 and receivers at nodes 2 and 3 requiring rates 1 and 2 respectively. The line costs are the cost per unit rate.

same subset. This is demonstrated in the example in section 4.3.2.

2. Run the T-M heuristic on the subset with the highest requirements.
3. Once the tree with the subset of receivers with the highest requirements has been constructed, repeat the heuristic with this tree as the starting tree for the subset of receivers with the next highest set of requirements.
4. Repeat the procedure until all subsets of receivers are connected to the tree.

In a network with a high and low bandwidth requirement, the receivers with the high requirement are used to create a backbone of high bandwidth circuits. After the backbone is created, the receivers in the low requirement subset, that were not attached as part of the backbone, are added as thinner branches of low bandwidth circuits. Following this procedure, the links that are added to the tree never increase the requirements of the links that were previously assigned.

3.3 Unified Implementation

Throughout this section we have alluded to a common structure for the four algorithms and how closely they are related when considered in our framework. In figures 7 and 8 we present the pseudo-code for a common implementation. The variables that appear in the pseudo-code and their initial values are tabulated in figure 6.

Each of the four algorithms temporarily adds nodes to a tree in order of increasing depth until a node in a target set is encountered. At that point, the node in the target set and the nodes between it and a permanently connected node are made permanent, and all of the other temporary nodes are removed from the tree. The procedure continues until all of the nodes in the target set are permanently attached to the tree.

The minimum spanning tree and the minimum depth tree algorithms differ only in the multiplier that determines the depth of a permanently connected node. The multiplier is one for the minimum depth tree and zero for the minimum spanning tree. As a result, in the minimum spanning tree the depth of a permanently connected node is zero, and in the minimum depth tree the depth of a permanently connected node is the distance to the root of the tree.

The T-M heuristic differs from the minimum spanning tree algorithm only in the target set. In the minimum spanning tree all of the nodes are in the target set, while in the T-M heuristic only receivers are in the target set. Each procedure successively connects the next closest node in the target set to the tree. However, in the minimum spanning tree algorithm, the next closest node is always connected by a single link, while in the T-M heuristic it may be connected through a sequence of intermediate nodes that are not in the target set.

The modified heuristic and the T-M heuristic differ only in the nodes that are in the target set at a particular time. In the T-M heuristic, all of the receivers are in the target set from the beginning. In the modified heuristic, all of the receivers are eventually in the target set, although low rate receivers do not enter the target set until all of the higher rate receivers are permanently assigned to the tree.

In this work we have concentrated on properly implementing the heuristics, and not on the time required to execute them. By implementing all of the algorithms with the same central code, it is less likely that errors have gone undetected. In addition, Kruskal's algorithm was implemented as an independent check on the answers obtained for the minimum spanning tree. In the experiments reported in the next section, we used networks with 1000 nodes and up to 10,000 receivers, and only took a few minutes on a SUN SPARC 20.

From the structure of the implementation, it is noted that the upper bound on the time to implement the TM heuristic is on the order of the the number of receivers times the implementation time for the minimum depth tree. The upper bound occurs when there are a small number of receivers, relative to the number of nodes, a tree consisting of all of the non-receiving nodes is constructed before a receiving node is encountered, and almost all of the non-receiving nodes are removed from the tree before trying to add the next receiving node. In the modified heuristic, the same number of receivers are added as in the T-M heuristic, only in a different order. The time to execute the modified heuristic is the same order as the T-M heuristic.

When implementation time becomes a concern, the conventional reduction techniques for speeding up Steiner tree heuristics[6] apply well to the modified heuristic. Simple reduction techniques include removing singly-connected nodes that are not receivers, replacing doubly-connected nodes that are not receivers by a link that bypasses the node, and removing the longer of two paths between receivers. The reduction techniques work best when there is a small number of receivers.

In the modified heuristic, the reduction techniques can be applied in stages to obtain greater reductions than applying them to the entire set of receivers. In the first stage, only the highest rate receivers are considered when reducing the network. After the tree of high rate receivers is formed, the complete network is restored, the nodes in the tree are collapsed to a single node, since they are all at depth zero, and the reduction procedures are applied only considering the receivers in the next set that will be added to the tree.

DEFINITIONS

node->Links[]	Set of directed links from this node	
node->rate	Maximum Rate receiver at a node	
node-> depth	current depth of node in algorithm	
node->type	= destination	if node must be part of the tree
	= intermediate	if node may be omitted from the tree
node->connect	= not_connected	if not connected to Tree
	= temporary	if temporarily connected to the tree
	= permanent	if permanently connected to the tree
link->to	The destination node of this directed link	
link->cost	cost assigned to a link	
link->reverse	pointer to a directed link that is the other half of this link’s bidirectional connection	
link->direction	= none	if not used in tree
	= root	if directed toward the root of the tree
	= leaf	if directed toward the leaf of the tree
Multiplier	Multiplier for depth of permanent nodes	
	= 0	if minimum spanning tree
	= 1	if minimum depth tree

INITIAL VALUES

node->depth	= 0	for the source
	= ∞	for all other nodes
node->connect	= permanent	for the source
	= not_connected	for all other nodes
node->type	= intermediate	for all nodes
link->direction	= none	for all links

Figure 6. Definitions and Initial Values Used in the Unified Algorithm

ALGORITHMS AND HUERISTICS

Minimum Depth Tree

```
for( each node )
   node->type = destination              All nodes must be added to the tree
Tree_construct( Multiplier = 1 )         Minimum depth permanent tree
```

Minimum Spanning Tree

```
for( each node )
   node->type = destination              All nodes must be added to the tree
Tree_construct( Multiplier = 0 )         Minimum spanning permanent tree
```

T-M Heuristic

```
for( each node )
   if(node-> rate > 0 )
      node->type = destination           Only nodes with receivers must be in the tree
Tree_construct( Multiplier = 0 )         Minimum spanning permanent tree
```

Modified Heuristic

```
for( each Rate in decreasing order ) {
   for( each node )
      if(node->rate == Rate )
         node->type = destination        Only nodes with receivers with specific rates must be
                                         Added to the tree at this step
   Tree_construct( Multiplier = 0 )      Minimum spanning permanent tree
}
```

Figure 7. Implementation of the Minimum Depth Tree, the Minimum Spanning tree, the T-M Heuristic, and the Modified Heuristic Using a Common Tree Construction Algorithm

Tree_construct(Multiplier)

```
while( permanent nodes can be added to the tree ) {

        Find the closest node that is not connected to the tree, but must be
        Any node for MST and MDT, select nodes for Heuristics
        while( next_node->type != destination) {

                Find the closest node that is not connected to the tree
                for( each node ) {
                        if( node-> connect != not_connected ){ Node IS connected
                                for( each link in node->Links[] ) {
                                        next_node = link->to
                                        if( next_node->connect == not_connected) {
                                                depth = link->cost + node->depth
                                                if(depth < min_depth ) {
                                                        min_depth = depth
                                                        next_link = link
                }       }       }       }       }

                Temporarily connect the closet node to the tree
                next_link->direction = leaf
                next_link->reverse->direction = root
                next_node = next_link->to
                next_node->connect = temporary
        }

         Make next_node and the nodes that connect it to the permanent tree, permanent
        while( next_node->connect != permanent) {
                next_node->connect = permanent
                next_node->depth = Multiplier*next_node->depth   Multiplier is different for MST and MDT
                for( link in next_node->Links[] ){
                        if(link->direction == root) {
                                next_node = link->to
                                break
        }       }       }

        Disconnect temporary nodes, and unassign links between nodes that are not permanent
        There are no temporary nodes or unassigned links in the MST and MDT
        for(each node) {
                if(node->connect != permanent){
                        node->connect = not_connected
                        node->depth = ∞
                }
                for( each link in node->Links[]){
                        if((node->connect != permanent) || (link->to->connect != permanent)) {
                                link->direction = unassigned
}       }       }       }
```

Figure 8. Common Tree Construction Algorithm for Minimum Spanning Tree (MST), Minimum Depth Tree (MDT), T-M Heuristic, and the Modified Heuristic

4. Comparing the Algorithms

In this section, we generate random networks and receiver distributions, and use minimum depth routing, truncated minimum spanning trees, the T-M heuristic and the modified heuristic to construct multicast trees. In section 4.1 we describe a general technique for generating networks with clusters of nodes, and in section 4.2 we describe the characteristics of the random networks that we use in our examples. In section 4.3 there are four sets of examples to show the characteristics of the modified heuristic. Although we emphasize the regions where the modified heuristic improves upon the other techniques, we also show a region where it can result in more costly networks.

4.1 Generating Random Networks

The multicast routing procedures are tested on randomly generated networks because the topology of commercial networks is difficult to obtain, and testing procedures on a specific network may inadvertently tailor them to that network. Testing the procedures on a number of randomly generated networks makes it likely that the procedures are robust, and will be useful as networks evolve.

The problem with using randomly generated networks is that they may not have some potentially important characteristics of real networks. For instance, if nodes are uniformly distributed over an area and connected together at random, then nodes that are on opposite edges of the network are as likely to be connected together as nodes that are close together. This characteristic is unrealistic, it results in networks with longer links and fewer nodes between distant receivers than exist in real networks. Conclusions that are based upon completely random networks, particularly as they apply to routing, are not always valid.

Efforts have been made to make randomly generated networks more realistic by adjusting probabilities so that it is more likely that links are placed between nodes that are close together[5]. Recently, it has been shown that the distribution of link lengths in this type of random network approximates the lengths in certain existing networks[16].

A problem with random networks that has not been addressed is the distribution of nodes. The nodes in random networks are typically distributed uniformly over a surface, while in real networks nodes tend to be clustered. For instance, in a national network there are more nodes around major population centers and in an international network there are more nodes on land masses. The fact that receivers are clustered has been implicitly assumed in practical tree construction mechanisms for the Mbone. For instance, in rapidly changing, multicast networks, center based trees[2] form a permanent backbone. The underlying assumption in a center based tree is that there are a small number of points in the network that are near most of the nodes.

In the random networks used in these examples we simulate the type of clustering that may occur in a national or international network. Clusters of nodes around cities in a national network is simulated by dividing a surface into a grid and randomly labeling the grid as a high density or low density region. Nodes are placed on the surface by picking a region according to a probability distribution, then randomly placing the node inside the region. International networks are similar to national networks, except that they not only have high and low density areas but also large areas that are uninhabited, such as oceans, that increase distances between populated regions of the grid. International networks are modeled by three types of regions, high density, low density, and zero density.

Once the nodes have been placed in the network, links must be placed between the nodes. As in reference 5, links are placed by picking a node at random, calculating a probability for destination nodes as a function of their distance from the first node, and selecting the destination.

In reference 5, a random network is generated and if it is not completely connected, it is discarded and a new network is generated. We found that when there are large numbers of nodes, between one and ten thousand, and possibly large distances between groups of nodes, as in the international networks, the probability of generating a completely connected network is extremely small. Typically, more than 90% of the nodes are connected together and there is a small group that is not connected to the main group.

In order to generate large, completely connected networks reasonably quickly, we select a node at random and as long as it is connected to more than 90% of the nodes, we consider set of nodes connected to that node to form a starting network. If the node is connected to less than 90% of the other nodes we generate a new network. The nodes are divided into two sets, the nodes in the starting network are in the connected set, and all other nodes are in the disconnected set. A link is selected at random. If the link is not connected to any singly connected node, it is removed from the network. A node is selected at random from the disconnected set, the distance function to the nodes in the connected set is calculated, and a node in the connected set is selected using the distance function to form a distribution function. A link is added between the two nodes. The node in the "disconnected" set, and all of the nodes that it is connected to, are moved to the "connected" set. The procedure is repeated until the disconnected set is empty.

4.2 Networks used in the Examples

The networks in the examples have 1000 nodes. In the national networks 20% of the surface is high density and contains 80% of the nodes. The networks are generated on an eight by five grid, with 8 randomly spaced high density regions, with an average of 100 nodes each and 32 low density regions with an average of 6.25 nodes each. Nodes are placed by first selecting a region, then selecting a location in the region according to a two dimensional uniform distribution. Each high density region has a probability of .1 of being selected, and each low density region has a probability of .00625 of being selected. A typical distribution of one thousand nodes by region is shown in figure 6.

118	1	2	10	3
108	5	10	6	7
6	6	93	8	7
90	5	94	4	6
12	3	8	8	5
7	6	94	9	7
3	98	4	5	6
105	6	9	10	6

Figure 9. A typical distribution of 1000 nodes on a 5x8 grid that represents a national network with 20% of the regions having 80% of the nodes

In the international networks, 80% of the area is unpopulated, and of the remaining 20% of the area, 80% is low density and 20% is high density. In our examples the networks are generated on a 20 by 10 grid, with 160 unoccupied regions, 32 low density regions, and 8 high density regions. A uniform distribution is used to place both the low and high density regions on the grid. There are the same number of high and low density regions in both the national and international networks. A typical distribution of nodes by region, when 80% of the nodes are in high density regions, is shown in figure 7.

In each of the networks there are three times as many links as nodes. On the average each node has six connections. The probability of connecting nodes is inversely related to the square of the distance between the nodes.

The nodes in the network are switching centers and can support any number of receivers. The source and receivers are placed on the network by selecting nodes at random.

4.3 Examples

The networks are loaded by placing a single source and between 20 and 10,000 receivers on a network. Each receiver can require one of two rates. Each load is applied to 10 different networks and on each network the load is distributed in 10 different ways.

The methods for connecting receivers to the source are compared by calculating the percentage increase in cost of each of the techniques relative to the modified heuristic. For these examples, the cost of a link is

112	7								
9									
								3	
4	7				9			6	
						3			5
	5				111				
		2			9			5	8
101									
4		15		101			4		
					7				
			103						
					5		5		
					2				
7	96		7			6		12	
						5	4		2
	12			91				5	
85								8	
		8							

Figure 10. A typical distribution of 1000 nodes on a 10x20 grid that represents an international network with 80% of the regions unoccupied, and 20% of the occupied regions having 80% of the nodes

proportional to the length of the link and the rate that is required on the link. The distance used to measure the link lengths is normalized so that each square on a grid is one unit by one unit. The cost is also normalized, so that the cost of a low rate link for 1 unit if distance is 1 unit of cost. In the shared systems, the rate that is required is the rate of the highest rate receiver that uses the link. The cost of a network is the cost of the links that are used, and equals $D_L + C_R \times D_H$, where D_L, and D_H are the lengths of the low and high rate links and C_R is the cost of the high rate link relative to the low rate link. In the point-to-point system, the rate is the sum of the rates of all receivers that use the link. The cost of the network is $\sum_i \left[N_{i,L} + C_R N_{i,H} \right] \times D_i$, where the summation is over all of the links in the network, D_i is the length of the link, and $N_{i,L}$ and $N_{i,H}$ are the number of low and high rate receivers that use link *i*. The cost of a tree generated by an heuristic relative to the tree generated by the modified heuristic is calculated as

$$\frac{Cost\ of\ Heuristic\ X - Cost\ of\ Modified\ Heuristic}{Cost\ of\ Modified\ Heuristic} \times 100.$$

4.3.1 Example 1: In this example we determine the basic characteristics of the methods for connecting receivers to a source and determine the differences between national and international networks. Loads of 20, 40, 100, 1000, and 10,000 receivers are placed on the networks. In each case, 20% of the receivers are high rate and $C_R = 5$. The results of this experiment are summarized in table 1.

There are 8 high density regions in each of the networks. Therefore, when the network load is 20 receivers, there are only 4 high rate receivers, and not every high density region has a high rate receiver. With this load, the difference between the national and international networks is the greatest. The percent increase of the TM heuristic and the truncated spanning tree over the modified heuristic are 16% and 90% for international networks versus 10% and 60% for the national networks. The difference occurs because the high density regions are further apart in the international network, and, when not every high density region has a high rate receiver there is the best chance that we can eliminate long distance, high rate lines from the tree generated by the T-M heuristic. Note that the 10% to 16 % cost reduction that the modified heuristic obtains over the T-M heuristic is based upon providing shorter routes for the 20% of the receivers that

Number of Receivers	Dedic. Lines	Shared Lines		
	Minimum Distance	Min. Depth Tree	Min. Spanning Tree	T-M Heuristic
	National Network			
20	134	58	60	10
40	233	85	49	12
100	370	118	49	14
1000	1380	137	12	8
10000	7352	217	0	0
	International Network			
20	115	60	91	16
40	191	83	69	14
100	470	124	36	11
1000	2183	155	10	6
10000	12986	239	0	0

Table 1. Percentage increase of supporting receiver loads on national and international networks using dedicated line, minimum distance shared trees, truncated minimum spanning trees, and a Steiner tree heuristic, rather than the modified heuristic. Twenty percent of the receivers require five times the rate of the others

require a higher rate. Generally, the links used by the 80% of the receivers that requires low rates will be longer, as discussed in section 4.3.2.

Of all of the loads in this example, the 20 receiver load is the only load at which the minimum distance, shared link assignment is less costly than the truncated, minimum spanning tree assignment. The minimum distance assignment is likely to be less costly than the minimum spanning tree assignment whenever the ratio of receivers to nodes is small. The T-M heuristic, because it is constructed as a combination of the minimum depth and minimum spanning tree assignment, will always perform at least as well as the best of these two algorithms.

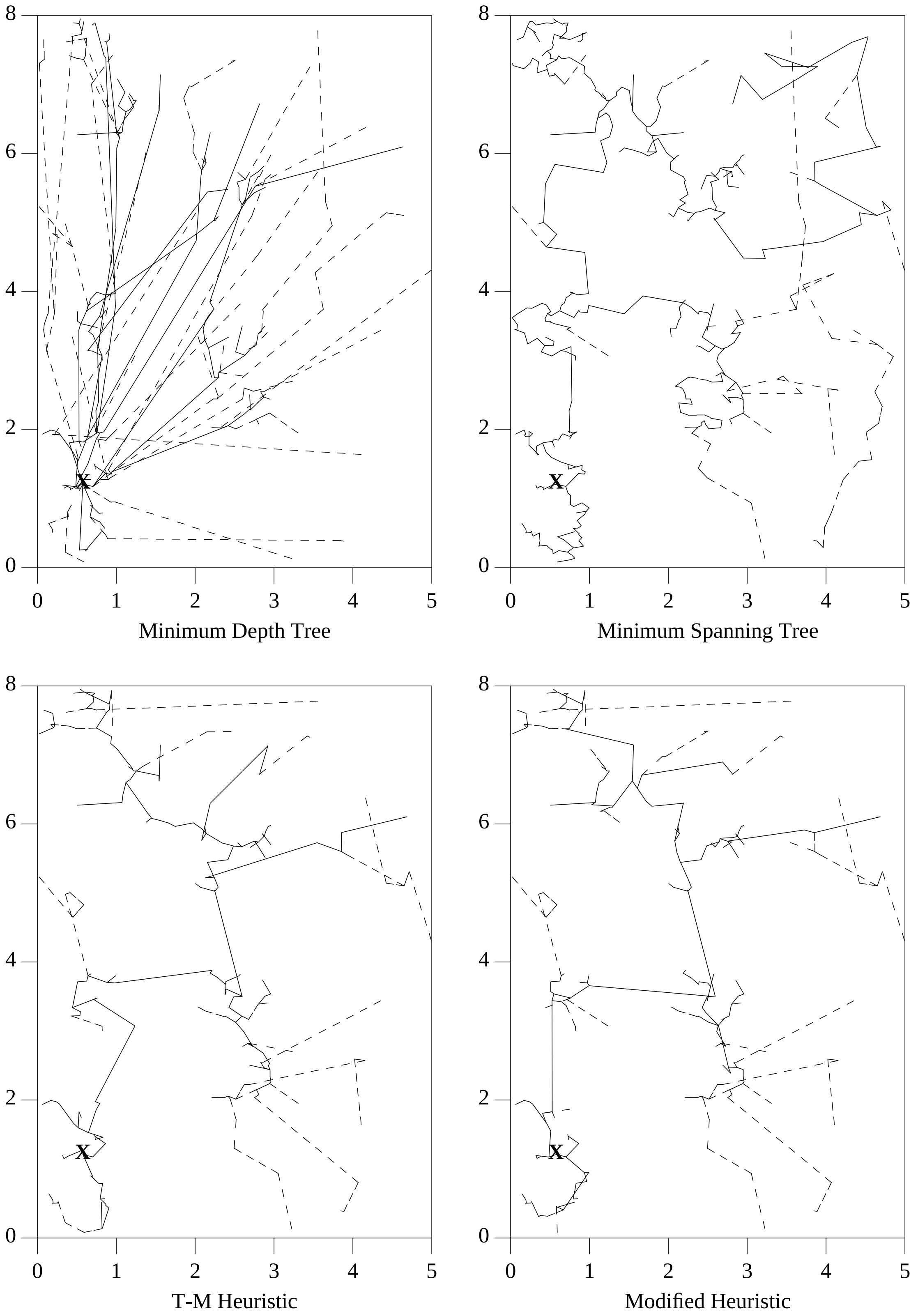
When there are 100 total and 20 high rate receivers, most high density regions have high rate receivers but only 10% or less of the switches have any receivers. When there are 10,000 receivers, there are ten times as many receivers as switches, so that most, if not all, of the switches have receivers. In addition, there are twice as many high rate receivers as switches, so that most switches also have high rate receivers.

When 10% of the nodes have receivers, the Steiner tree can reduce the cost of a truncated minimum spanning tree. However, when all of the nodes have receivers the Steiner tree and the minimum spanning tree are identical. Similarly, when approximately 20% of the nodes that have receivers require higher rates than the other nodes, the tree generated by the modified heuristic can cost less than the tree generated by the T-M heuristic. However, when all of the nodes with receivers require a high rate connection, then trees generated by the modified heuristic and T-M heuristic are identical. The results reported in table 1, support these conjectures. For the national networks with 100 receivers, the trees generated by truncating a minimum spanning tree and the T-M heuristic are 49% and 14% more costly than the tree generated by the modified heuristic, with similar results for international networks. By the time there are 10,000 receivers, there is no cost difference between the minimum spanning tree, the T-M heuristic and the modified heuristic. It should also be noted that as the number of nodes becomes large, the trees generated for minimum distance cost more than 200% more than the trees generated by the other techniques.

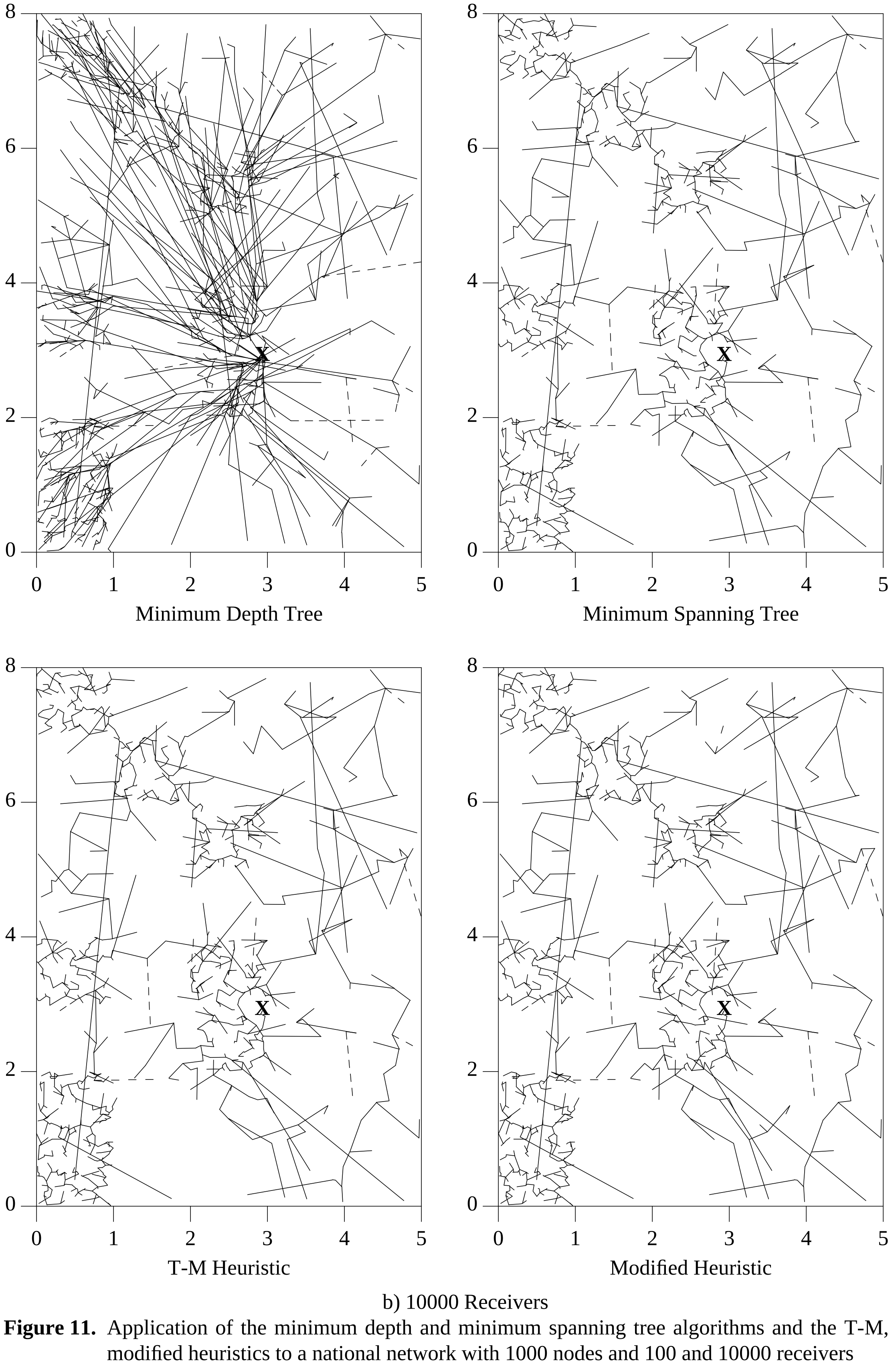
The improvement of shared networks over dedicated networks increases as the load increases. This is expected, because more receivers share the links. In a network where the nodes are switching centers, that support many receivers, once the switching center is connected to the network, increasing the number of receivers at the switching center does not increase the cost of a shared network.

Figures 11 and 12 are typical of the multicast trees that are generated by the four approaches on national and international networks with 100 and 10,000 receivers. "X" marks the source. The solid lines are high rate links, the dashed lines are low rate links, and the links in the network that are not part of the tree are not shown. Each of the four trees in a figure connects the same set of receivers to the source, and does not display any links that are not required to reach a receiver. However, the different techniques generate different paths to the receivers. The underlying networks are the same networks that resulted in the node distributions in figures 9 and 10.

The 8 high density regions are easily picked out, particularly in the international network. The figures clearly show the results discussed in the previous paragraphs. With 10,000 receivers, the trees generated by the minimum spanning tree, the T-M heuristic and the modified heuristic are almost identical, and clearly use fewer links than the minimum depth technique. With 100 receivers, the improvement provided by the Steiner tree heuristics over the minimum spanning tree is clear, as is the reduction in the high rate links in the tree constructed by the modified heuristic.

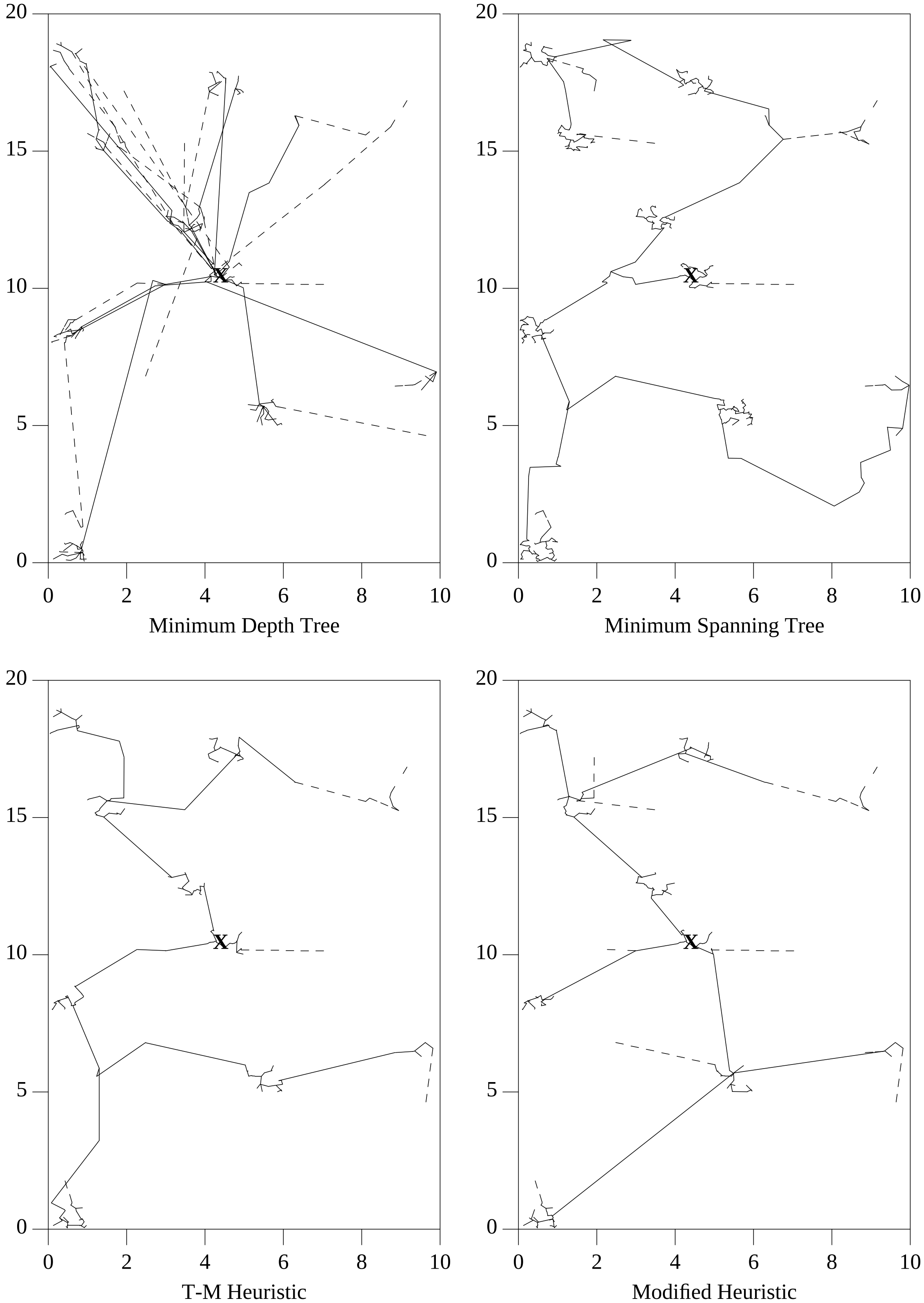


a) 100 Receivers

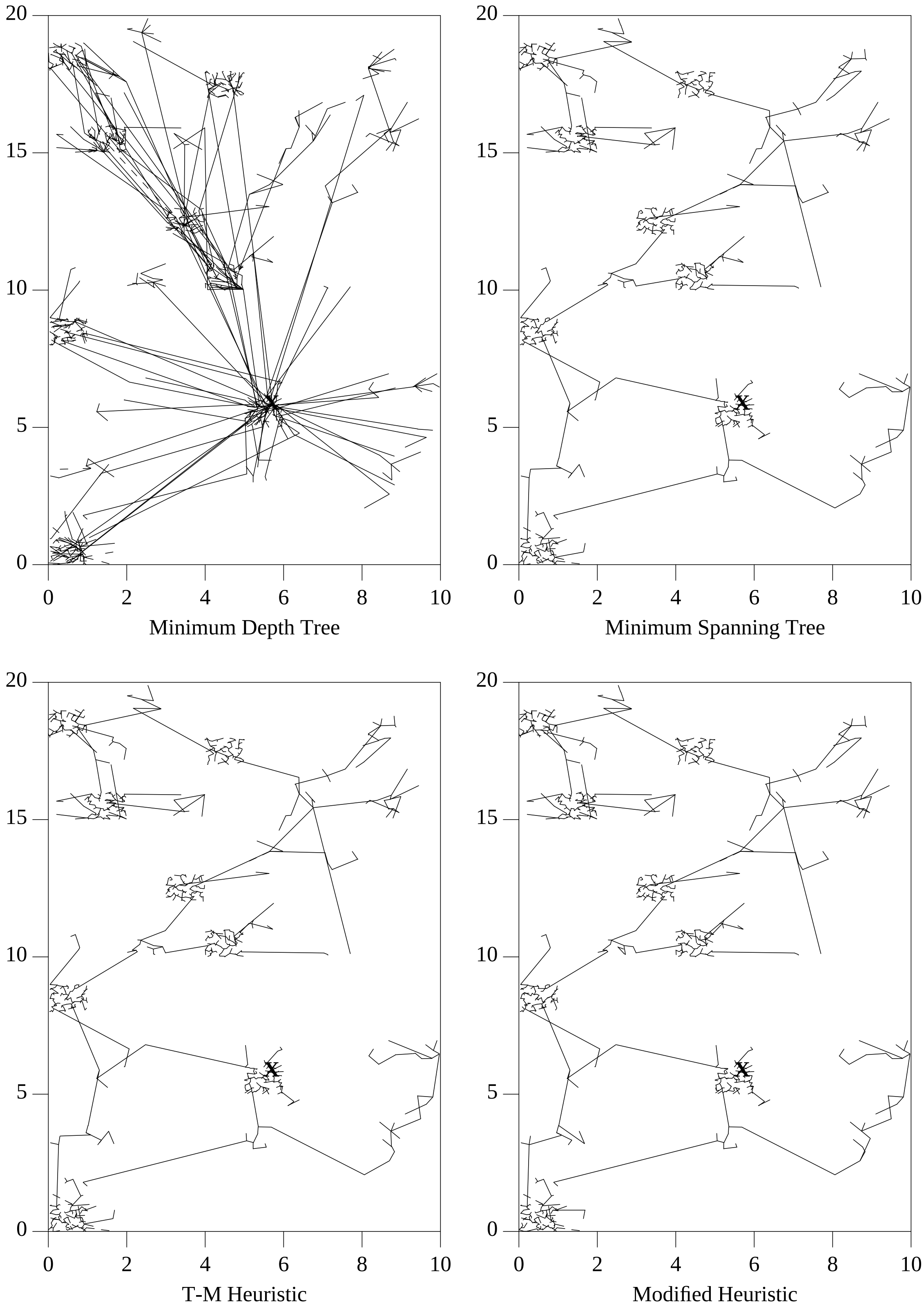


b) 10000 Receivers

Figure 11. Application of the minimum depth and minimum spanning tree algorithms and the T-M, and modified heuristics to a national network with 1000 nodes and 100 and 10000 receivers



a) 100 Receivers



b) 10000 Receivers

Figure 12. Application of the minimum depth and minimum spanning tree algorithms and the T-M, and modified heuristics to a international network with 1000 nodes and 100 and 10000 receivers

4.3.2 Example 2: In this example we demonstrate the effect of changing the cost differential between low and high rate circuits. We also give a more detailed breakdown in the costs of links in the various networks. The breakdown of costs between low rate and high rate connections gives insight into the similarities and differences in the relationship between the truncated minimum spanning tree and the T-M heuristic and the relationship between the T-M heuristic and the modified heuristic. The relationships point out a danger in creating groups of receivers in the modified heuristic that have rates that are close to each other.

The cost differential between high rate and low rate links in the previous example is 5:1. There are valid reasons for considering larger differentials. Video conferencing equipment operates from rates that use *DS* 0 voice circuits, 64 kbps, up to *DS* 1 circuits, approximately 1.5 Mbps. The rate differential in video conferencing equipment is 24:1. In reference 14 a video gateway for the Internet is described that converts "multi-megabit JPEG" to 128 kbps H.261. Presumably, the rate differential between these two encodings is greater than 25:1. In table 2 we show the effect of increasing the cost differential from 5:1 to 20:1 on the set of international networks with 1000 nodes and 100 receivers.

	Dedic. Lines	Shared Lines			
	Minimum Distance	Min. Depth Tree	Min. Spanning Tree	T-M Heuristic	Modified Heuristic
Length of Links in the Trees					
High Rate	174	90	64	52	44
Low Rate	572	121	24	23	35
TOTAL	745	211	88	75	79
Cost of Links in the Tree at a 5:1 Cost Ratio					
High Rate	871	448	319	259	218
Low Rate	572	121	24	23	35
TOTAL	1443	569	343	282	253
Cost of Links in the Tree at a 20:1 Cost Ratio					
High Rate	3784	1791	1276	1037	873
Low Rate	572	121	24	23	35
TOTAL	4056	1912	1300	1060	908
Percent Cost Increase Relative to the Modified Heuristic					
5:1	352	125	36	11	
20:1	347	111	43	17	
∞:1	295	105	46	18	

Table 2. Effect of Increasing the Ratio of the Cost of High Rate Links to Low Rate Links on International Networks with 1000 Nodes, 80 Low Rate Receivers and 20 High Rate Receivers

In the limit as $C_R \to \infty$, the cost relative to the modified heuristic is calculated as $\frac{D_{H,X} - D_{H,M}}{D_{H,M}}$ where $D_{H,X}$ and $D_{H,M}$ are the distances spanned by the high rate links in heuristic "X", and the modified heuristic. The cost of the low rate links is insignificant in calculating the cost of the network.

Observe that the increase in the improvement of the modified heuristic over the T-M heuristic and the spanning tree is small as the cost ratio increases from 20;1 to ∞:1. This occurs because the cost of the low rate links is small relative to the cost of the high rate links when $C_R = 20$. When the cost of the low rate links is insignificant, applying the modified heuristic to first generate a Steiner tree for the high rate receivers, rather than a tree with all of the receivers, has the same justification as generating a Steiner tree rather than truncating a minimum spanning tree when all of the receivers have the same cost. We will expand upon this statement in example 3.

Because of the way the T-M heuristic is implemented, the length of the links in the tree it generates is always less than or equal to the length of the links in the tree generated by truncating the minimum spanning tree. Similarly, if we compare trees of high rate receivers generated by the first step of the modified heuristic and by truncating the tree of all receivers generated by the T-M heuristic, the length of the links in the first tree is less than or equal to that in the second. If we start with the tree of receivers generated by the T-M heuristic, and add the other nodes, we cannot generate a tree whose link lengths are less than those generated by the minimum spanning tree. Similarly, the tree generated by the modified heuristic, which successively applies the T-M heuristic to subsets of the receivers, seldom generates a tree with link lengths less than that generated by the T-M heuristic applied to all of the receivers at once.

It is clearly demonstrated in table 2, that the modified heuristic, when compared with the T-M heuristic, reduces the length of the high rate links at the expense of increasing the lengths of the low rate links, and the length of all of the links in the final tree. In this example the high rate links are reduced by 8 length units while the low rate links are increased by 12 units. The cost of the tree generated by the modified heuristic is less than that generated by the T-M heuristic, because the high rate links are either 5 or 20 times more expensive than the low rate links. However, if $C_r < 1.5$ the cost of the tree generated by the modified heuristic would be greater than that generated by the T-M heuristic. This problem does not occur when comparing the T-M heuristic with the truncated minimum spanning tree, because the links that are not in the Steiner tree cannot add cost to the network.

This result indicates the need to take care when partitioning receivers into sets. If there are small differences in the rates required by receivers in the different groups, the cost of the network can increase. It is best to place receivers with rates that are close to one another in the same set. If there is no clear way to partition the receivers, the modified heuristic should be tried with several partitions.

4.3.3 Example 3: In this example we demonstrate a region where the modified heuristic provides a large improvements over the alternative techniques. The region is located by observing when the Steiner tree heuristics provide a large improvement over truncated spanning trees. Steiner tree heuristics generally give the greatest improvement over truncated minimum spanning trees when a small fraction of the nodes in the network are in the tree. Similarly, we expect the modified heuristic to provide the greatest reduction in the cost of high rate links when the high rate receivers are a small fraction of the total receivers. Our concern with the modified heuristic is that the cost of a large number of low rate receivers may increase.

In order to determine the effects of small fractions on receivers to nodes and high rate receivers to receivers, two cases are examined. In each case, five percent of the receivers are in the high rate group. In the first case, slightly less than 10% of the nodes are in the Steiner tree, and in the second case most of the nodes are in the Steiner tree. The results are summarized in table 3.

Number of	Dedic. Lines	Shared Lines		
Receivers	Minimum Distance	Min. Depth Tree	Min. Spanning Tree	T-M Heuristic
100	213	62	55	29
1000	1124	144	29	26

Table 3. Percentage increase of supporting receiver loads on an international networks using dedicated line, minimum distance shared trees, truncated minimum spanning trees, and a Steiner tree heuristic, rather than the modified heuristic. Five percent of the receivers require 20 time the rate of the others.

In the first case the receivers are a small fraction of the nodes in the network and the high rate receivers are a small fraction of the receivers. The modified heuristic results in nearly a 30% improvement over the T-M heuristic. This is particularly high when one realizes that the paths for 5% of the receivers were improved at the expense of the other 95% of the receivers. The cost of the truncated spanning tree and the T-M heuristic tree are 1.55 and 1.29 times the cost of the modified heuristic tree. Therefore, the truncated spanning tree is $(1.55 - 1.29)/1.29 = 20\%$ higher in cost than the T-M heuristic tree.

In the second case, the number of receivers approaches the number of nodes and the T-M heuristic provides only a $(1.29 - 1.26)/1.26 = 2\%$ improvement over the truncated spanning tree. However, only a small fraction of the nodes in the Steiner tree have high rate receivers, so that there is still a 26% improvement provided by the modified heuristic. The second case demonstrates a region where the modified heuristic significantly reduces the cost of the Steiner tree heuristic, even though the Steiner tree does not significantly reduce the cost of the truncated minimum spanning tree.

4.3.4 Example 4: The Steiner tree heuristics generally provide a greater improvement over truncated spanning trees as the fraction of nodes in the network that are in the tree decreases. A similar effect is noted for the modified heuristic with respect to the number of nodes in the high rate set. In table 4 the receivers in the high rate set vary from 5% to 30% of the total receivers.

Number of High Rate Receivers	Dedic. Lines	Shared Lines		
	Minimum Distance	Min. Depth Tree	Min. Spanning Tree	T-M Heuristic
5	213	62	55	29
10	253	86	51	24
20	347	111	43	17
30	433	130	40	13

Table 4. Percentage increase of supporting receiver loads on an international networks using dedicated line, minimum distance shared trees, truncated minimum spanning trees, and a Steiner tree heuristic, rather than the modified heuristic, on networks with 100 receivers, of which 5, 10, 20 or 30 require 20 times the rate of the others.

5. Conclusion

The tools being developed for the Internet Mbone make it possible to deliver audio and video programs to sparsely distributed groups using standard computer equipment. As the use of this service increases, it becomes more important to use the transmission facilities as efficiently as possible.

In this work we relate the problem of designing low cost networks for this application to the problem of designing Steiner trees. The significant difference between the two problems is that the link costs in the Steiner tree problem are fixed, but the link costs in the new problem can change depending upon the receivers that use the link.

An heuristic algorithm is developed to solve the new problem. The new heuristic is related to T-M heuristic for Steiner trees, Solin’s algorithm for designing minimum spanning trees, and minimum distance routing. A common framework is established for all four techniques, and the pseudo-code of a common implementation is provided to emphasize that the four techniques are closely related.

Minimum distance routing, truncated spanning trees and Steiner tree heuristics are currently the common techniques used to design multicast trees. A set of examples are used to compare the trees generated by these techniques with the tree generated by the new heuristic. It is noted that the regions where the new technique can provide the greatest improvement over the existing techniques is related to the regions where the Steiner tree heuristic provides the greatest improvement over minimum spanning trees, and for similar reasons. An example is also shown where the Steiner tree results in almost no improvement over the minimum spanning tree, but the modified heuristic results in a reasonably large improvement over both.

There are differences in the reason for the cost reduction in the original Steiner tree problem and the new problem. The Steiner tree heuristics never increase the cost of the truncated minimum spanning tree. However, if the new heuristic is used without understanding these differences, the cost of the network can be greater than the cost of a Steiner tree network.

REFERENCES

[1] S. Deering, "Multicast Routing in Internetworks and extended LAN's," ACM SIGCOMM '88, Aug. 1988, Stanford, Calif., pp 55-64.

[2] J. C. Pasquale, G. C. Polyzos, G. V. Xylomenos, The Multimedia Multicasting Problem," UCSD/CSE Tech. Report, CS93-313.

[3] L. F. de Moraes, S. B. Weinstein, "The Internet Multicast from ITS: How it was Done and Implications for the Future," IEEE Commun. Mag., Jan. 1995, vol. 33, no. 1, pp. 6-8.

[4] D. J. Mitzel, D. Estrin, S. Shenker, L. Zhang, "An architectural comparison of ST-II and RSVP," Proc. of IEEE Infocom '94, June 1994, Toronto, Canada.

[5] B. M. Waxman, "Routing in Multipoint Connections," IEEE Trans. Commun., vol. 6, no. 9, Dec. 1988, pg. 1617-1622.

[6] P. Winter, "Steiner Problem in Networks: A Survey," Networks, vol. 17, 1987, pp 129-161.

[7] H. Takahashi, A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," Math. Japonica, 1980, vol. 6, pp. 573-577.

[8] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource Reservation Protocol," IEEE Network, Sept. 1993, vol. 7, no. 5, pp. 8-18.

[9] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," Proc. of IEEE Infocom '92, May 1992, pp. 2107-2114.

[10] V. Kompella, J. Pasquale, G. Polyzos, "Multicast Routing for Multimedia Communications," IEEE/ACM Trans. on Networking, vol. 1, no. 3, June 1993, pp. 286-292.

[11] M. Doar, I. Leslie, "How Bad is Naive Multicast Routing," IEEE Infocom '93, Mar. 30-Apr 1, 1993, pg 83-89,

[12] G. Karlsson, M. Vetterli, "Packet Video and Its Integration into the Network Architecture," IEEE JSAC, vol. 7, no. 5, June 1989, pp. 739-751.

[13] M. Ghanbari, "Two-Layered Coding of Video Signals for VBR Networks," IEEE JSAC, vol. 7, no. 5, June 1989, pp. 739-751.

[14] E. Amir, S. McCanne, H. Zhang, "An Application Level Video Gateway," Proc. of ACM Multimedia, Nov. 1995, pp. 255-265.

[15] V. Chachra, P. M. Ghare, J. M. Moore, "**Applications of Graph Theory Algorithms**," North Holland, 1979.

[16] E. Zegura, K. Calvert, S. Bhatacharjee, "How to Model an Internetwork," Proc. of IEEE Infocom 96, Mar. 1996, pp. 594-602.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Video Distribution on Multicast Networks

N. F. Maxemchuk

AT&T Labs - Research
Murray Hill, NJ 07974

ABSTRACT

The Internet is being used to distribute video programming to small, widely distributed audiences over a multicast backbone. This type of service can be used to deliver specialized programming to the general population. As the service becomes more widely used, conserving the required transmission facilities becomes more important.

In this work, we define the problem of designing multicast networks for video distribution. The problem is related to the Steiner tree problem, with an interesting twist. The line costs are not constant. An heuristic is developed to solve the problem and the new heuristic is related to and compared with heuristics developed for the Steiner tree problem and algorithms to design minimum depth and minimum spanning trees.