LOCALIZED MULTICAST ROUTING

Anees Shaikh, Shirong Lu and Kang Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

Abstract - Multicast is a fundamental issue in distributed computing and networking, especially for recent applications such as audio and video transmission. The minimal cost route selection problem for multicasting is an NP-complete problem even for regular network topologies such as meshes and hypercubes. We therefore present a simple heuristic algorithm for multicast route selection in arbitrarily-connected point-to-point communication networks. Several other heuristics have been presented for finding the minimal multicast route, but most of them are global in the sense that the source uses global cost information to construct a multicast tree. Our algorithm does not require the use of global cost information; it uses cost information only from neighboring nodes as it proceeds which makes it more practical from an implementation point of view. The performance of the algorithm is analyzed through empirical comparisons and is shown to perform as well against algorithms which use global information.

I. INTRODUCTION

The recent emergence of multimedia and cooperative computing in distributed systems provides a new incentive to system designers to include support for multicast, or one-to-many, communication. In this mode of communication, a single source node in a system sends identical messages to multiple destinations. Single-destination (unicast) messaging and broadcast to the entire network are both special cases of multicast.

A fundamental issue in multicast communication is the determination of an efficient message route, commonly known as *multicast routing*. A widely-used approach in solving the multicast routing problem requires tree construction. Popularity of the tree-based approach arises due to the ability to potentially share many links in transmitting the message to the destination set. Also data replication is minimized; messages must be replicated only at forking nodes. This is in contrast to multicast achieved through multiple unicast operations, requiring a copy of the message for every operation.

Regardless of the method used to construct the multicast tree, most algorithms attempt to optimize tree construction for minimal cost. Total tree cost is generally defined as the sum of costs of all edges in the multicast tree. The problem of finding the least cost tree is the Steiner tree problem in graphs. It is formulated as follows [1]:

GIVEN:

An undirected network $G = (V, E, c)$ and a nonempty set of destinations D, where V and E are the set of vertices and edges of the network, respectively, and c is a cost function associated with each edge in G.

FIND:

A subnetwork $T_G(D)$ of G such that:

i) there is a path between every pair of nodes in $D \subset V$, and
ii) the total cost, $\sum_{e \in T_G(D)} c(e)$, is minimized over all possible trees.

The vertices included in the final solution that are not members of D are called *Steiner nodes.* Finding a Steiner tree is known to be NP-complete in the most general case [2]. Due to the wide application of Steiner trees, however, several heuristic algorithms have been developed which construct near-optimal multicast trees. Many of these suffer the drawback that the source node must maintain global cost information for the entire network [3–5] which may be impractical from an implementation standpoint.

It is important, therefore, to have the ability to construct good (i.e., near-optimal) Steiner trees without using global information. We present a simple heuristic algorithm to construct a multicast tree using local information at each node. It is shown through empirical comparisons that the algorithm produces results near those of global algorithms.

The remainder of this paper is organized as follows. Section II presents a brief overview of several heuristics for the Steiner tree problem applicable to multicast routing in point-to-point communication networks. Our localized heuristic algorithm is described in detail in Section III. Section IV presents results of simulations to gauge the performance of the algorithm against the optimal Steiner tree solution as well as other representative heuristics. Finally, the paper concludes with Section V.

II. HEURISTICS FOR MULTICAST ROUTING

Since the performance of a distributed system de-

The work reported in this paper was supported in part by the National Science Foundation under Grant MIP-9203895. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the NSF.

0-7803-2509-5/95 US$4.00 © 1995 IEEE

pends greatly on the *efficiency* of the message routing method, the Steiner tree problem is especially relevant. Karp [2] showed, however, that finding a Steiner tree in an arbitrary graph with arbitrary costs is an NP-complete problem. Furthermore, the problem remains NP-complete even in the cases where all edge costs are equal, the graph is bipartite, or the graph is planar. It is strongly NP-complete if rectilinear distances on a plane are used to represent cost, and strongly NP-hard if Euclidean distance in a plane is used. There are, however, some special cases of the rectilinear Steiner tree where polynomial time algorithms are available [6]. Finding a minimal multicast tree in parallel computers with regular topologies, such as 2-D meshes and hypercubes, is also an NP-complete problem [5]. Several exact algorithms for solving the Steiner tree problem are available, though with exponential complexity [1, 7, 8].

As a result of the exponential complexity of exact algorithms, many heuristic algorithms have been developed, a few of which are discussed below. For a more complete survey of existing heuristics and exact algorithms, readers are referred to papers by Winter [9, 10] and Salama [11], and the text by Hwang *et al.* [1].

One of the most widely-recognized heuristics is the KMB algorithm [3] which belongs to the class of algorithms known as distance network heuristics. It consists of finding the distance network for the source, s, and all $v \in D$. This is a graph whose edges are the minimum distances (costs) between each pair of nodes. Then some minimum spanning tree (MST) algorithm (e.g., Prim [12]) is run on this network to find the MST of the distance network. This tree, T_D, is then mapped back into the original network such that each edge of T_D is replaced by the corresponding shortest path in the original network, G. Finally, an MST is found in the subnetwork induced by T_D in G and pruned to remove all nodes u with degree 1 where $u \notin \{D, s\}$.

Two other "classic" algorithms are the shortest path heuristic (SPH) [13] and the average distance heuristic (ADH) [14], each of which claim slightly better performance than KMB. The SPH algorithm works simply by starting with the source and adding to the tree, T_S, a shortest path to the next closest node in the destination set until all nodes in D have been reached. It is possible to improve the algorithm by finding an MST in G using the vertices of T_S, similar to the last steps of the KMB algorithm [14]. ADH incrementally connects subtrees together through a central (Steiner) node by a shortest path. The algorithm begins with $|s| + |D|$ subtrees of one node each and the central nodes are chosen through an appropriate measure that appears in several forms in the literature.

Another, more recent, approach to multicast routing is to address the potentially dynamic nature of multicast connections. The idea is to reduce the cost of recomputing new trees each time a node is added to or deleted from the destination set assuming relatively frequent additions and deletions. One method, called naïve multicast routing simply finds a shortest path from the source to each destination to be added and uses the nodes in the route as Steiner nodes. Performance is claimed to exceed that of KMB [15]. The weighted greedy algorithm uses a weighted function to add a new node to the multicast tree via a shortest path either from the source or another node already in the tree, depending on the weighting [16]. It is also noted, however, that performance is substantially degraded after several modifications to the destination set without full reconfiguration of the multicast tree.

In light of the interest in developing multicast protocols for multimedia-type applications such as audio and video transmission, some attention is now given to routing algorithms that can meet the time constraints inherent in these types of applications. The constrained multicast tree algorithm [4] follows the basic steps of the KMB algorithm while restricting addition of nodes to those which do not cause deadline violations. The Tenet group proposed the constrained Bellman-Ford algorithm [17] which finds independent shortest paths from a source to a set of destinations until the longest constraint is violated. Shortest paths are based on the Bellman-Ford single-source shortest path algorithm [18].

The two delay-constrained algorithms mentioned above require global information at the source to make decisions based on costs of links throughout the network. In a large network this may be impractical, and as a result, local algorithms are necessary to improve the feasibility of multicasting in a large network. The nearest neighbor algorithm, for example, transfers the message from the source to the destination closest to it, along with the responsibility for all other destinations. This process is continued until all destinations have received the message [19]. Note that although routing responsibility travels with the message, it may still be necessary to maintain widespread cost information.

In this paper we present an algorithm that uses only local cost information at each node. Each node uses a greedy strategy to incrementally build the multicast tree that is based on lowest cost among incident links. The performance of the algorithm is similar to some of the classic Steiner tree heuristics described above. The simplicity of the algorithm, which will become apparent in Section III, also highlights to its utility.

III. A LOCALIZED MULTICAST ROUTING ALGORITHM

The localized multicast (LMC) algorithm arises from the observation that Prim's spanning tree algorithm [12] and Dijkstra's shortest path algorithm [20] use essentially the same type of greedy strategy. The only difference between these two is the main subroutine embedded in the algorithms. If we modify this subroutine to distinguish between destination nodes and non-destination nodes, the result is a multicast tree routing algorithm which generalizes the methods of both Prim and Dijkstra.

```
Multicast (G, s, D)
1  for each vertex v ∈ V do
2      d[v] = ∞
3      π[v] = NIL
4  d[s] = 0
5  S = ∅
6  Q = V
7  while Q ≠ ∅ do
8      u=Pop-Min(Q)
9      S = S ∪ {u}
10     for each vertex v ∈ Adj[u]
11         if d[v] > I_D(u)d[u] + w(u, v)
12             if v ∉ S
13                 d[v] = I_D(u)d[u] + w(u, v)
14                 π[v] = u
```

Fig. 1: Localized multicast routing algorithm

A. *Algorithm Details*

For our purposes, the communication network is modeled as an undirected graph $G = (V, E)$ where V is a set of processor nodes and E is the set of communication links. We assume that the cost $w(u, v)$ is non-negative for each link $(u, v) \in E$. Given a source $s \in V$ and a set of destinations $D \subset V$ such that $s \notin D$, a multicast route is a subtree of G rooted at s, which contains all nodes from D, and whose leaves consist of nodes from D.

To distinguish nodes as being in the destination set, we define an indicator function, I_D as follows.

Definition: Given a set $D \subset V$, the indicator function $I_D : V \mapsto \{0, 1\}$ of D is defined as $I_D(u) = 0$ if $u \in D$, and 1 if $u \notin D$.

The LMC algorithm, shown in Fig. 1, maintains a subset S of vertices whose path from the source has already been determined. It repeatedly selects the vertex $u \in V - S$ with the minimum key value $d[u]$. The key represents the shortest-path estimate from either the source or the destinations found so far. The selected vertex is inserted into S, and all edges leaving u are relaxed. We maintain a priority queue Q that contains all the vertices in $V - S$ sorted by their d values. The pointer $\pi[u]$ points to the parent node of u in the multicast tree.

Lines 1–6 initialize the algorithm so that all nodes except the source have key values of infinity and parent nodes are yet undetermined. In addition, the priority queue Q is created and filled. The indicator function is used in line 11 so that the incremental distance is zeroed if u is a destination. That is, from a destination, the only cost incurred is the additional link cost, not the accumulated cost. This causes the destination node to behave like a new 'source'. The reason for this is that any nodes reached from a destination node incur only an incremental additional cost since we must absorb the cost for reaching the destination anyway. The **for** loop in lines 10–14 relaxes the edges leaving the current node by adjusting the key values and resetting the parents if necessary. The condition in line 12 ensures that cycles in the tree are avoided by preventing consideration of an adjacent node whose route has already been determined. Note that LMC constructs a spanning tree rooted at s for the whole graph G. A multicast routing tree from s to D is obtained by trimming this tree so that all leaves are destination nodes.

B. *Algorithm Operation*

We provide a detailed example to show how the LMC works. Figure 2(a) shows the initial network with costs on each edge that may represent delay or link congestion. The initialization steps are shown in Fig. 2(b) and Figs. 2(c–g) show the progression of the algorithm, with each branch added to the tree shown in boldface on the network diagram. The final multicast tree is shown in Fig. 2(h), along with its cost. Nodes A and G are Steiner nodes as they are not members of D. In this particular example the LMC was able to find the minimal Steiner tree.

IV. PERFORMANCE COMPARISON

We find the primary advantages of the localized multicast algorithm to be threefold. First, it is not necessary that each node know link cost information throughout the network, only that of its links to neighboring nodes. Second, the algorithm's operation is simple. The multicast route is constructed by visiting each node once and performing a link relaxation operation before moving to the next node. There is no need to compute auxiliary trees (KMB) or shortest paths at once through the network (SPH, ADH). This simplicity makes LMC attractive for actual implementation. Finally, our results for several example networks show that LMC performs comparably to other algorithms that use global information. We chose the DNH, SPH, and ADH algorithms for comparison because they represent the basic techniques adopted by the majority of existing multicast algorithms.

In the example in Section III (Fig. 2) the LMC algorithm's solution was equivalent to those obtained by the SPH and ADH algorithms. These also turned out to be optimal solutions. The KMB, based on the DNH algorithm, reached a solution that was suboptimal by one cost unit. Generally, DNH algorithms perform worse than ADH and SPH algorithms [14] but there is no dominance relationship between them. That is, it is possible to construct examples where each outperforms the others.

A 9-node example is shown in Fig. 3. Note that in this case all global algorithms produce the same cost tree. LMC is also able to match this performance. None, however, found the minimal cost tree. A representative network from a set of randomly generated graphs is shown in Fig. 4. Here, again, all the global algorithms were able to find equal cost solutions, although their particular routes were not identical. Note that the LMC algorithm found an equivalent solution, which turned out to be the optimal solution.

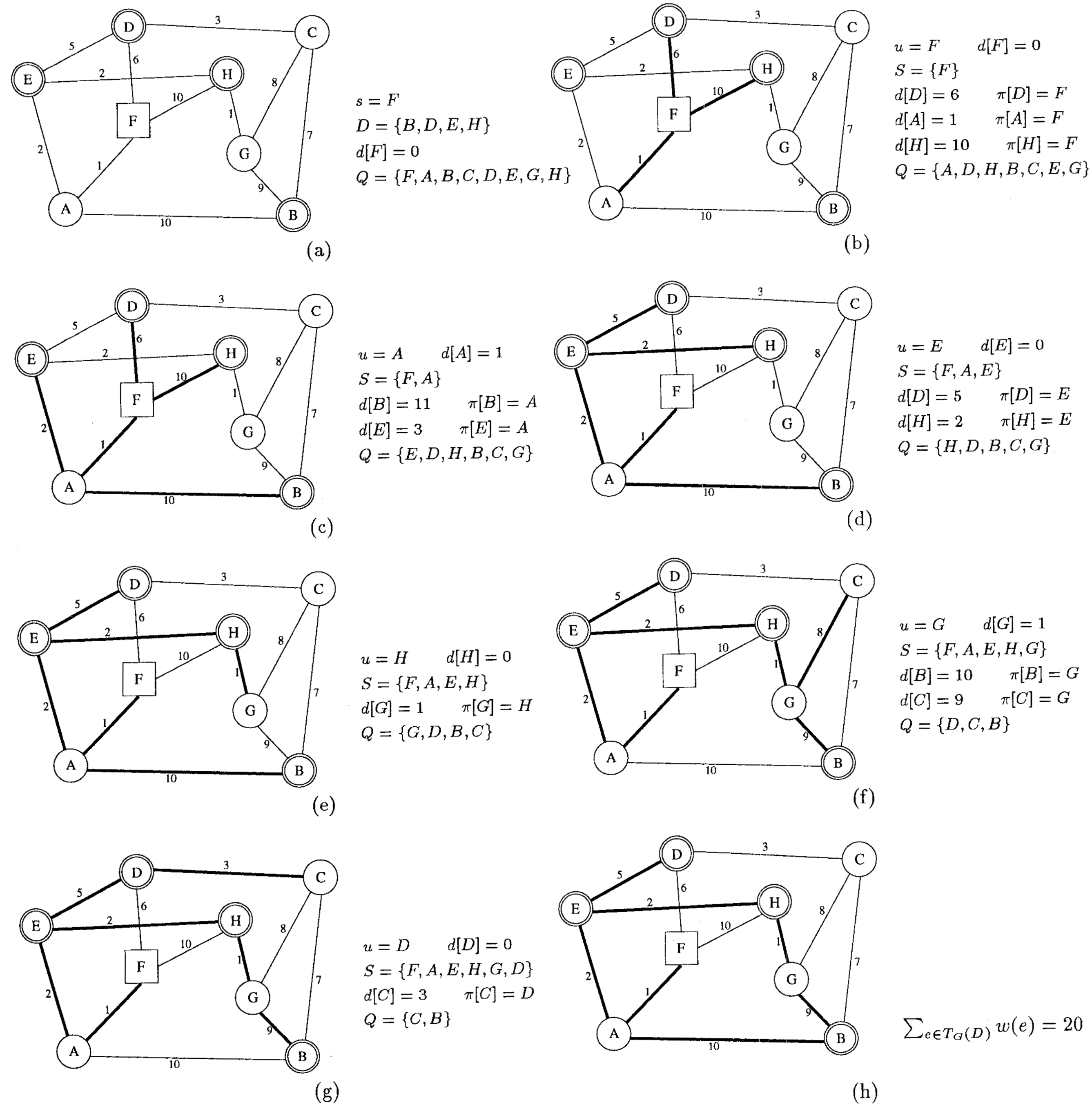

Fig. 2: Example network and algorithm operation

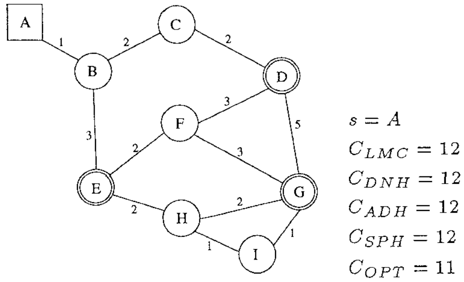

Fig. 3: Cost comparison of various heuristics (avg. node degree = 2.7)

V. CONCLUSIONS

In this paper we presented a simple localized algorithm, LMC, for multicast routing in arbitrary topology networks. The most important features of LMC are its treatment of destination nodes as new sources and the fact that at each route selection step, nodes need only query their incident links for cost information. The algorithm performs close to other well-known algorithms which employ global cost information regarding the entire network, as illustrated through several examples. Although the performance of LMC may not be equal to that of global algorithms in all cases, it is better suited to actual implementation in networks requiring support for multicast communication.

Currently we are performing a more complete evaluation of LMC, which will include a more formal presentation and proof of its operation. In addition, we plan to extend its utility to real-time applications such as audio and video transmission. In these applications the link cost could be cast as link delay or network congestion in order to reserve multicast bandwidth on real-time channels [21].

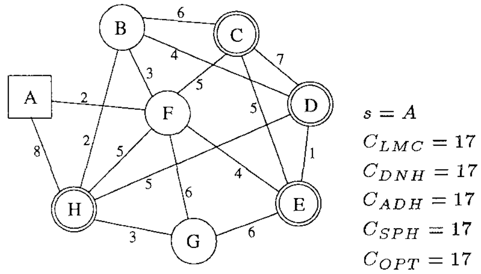

Fig. 4: Cost comparison of various heuristics (avg. node degree = 4)

REFERENCES

[1] F. Hwang, D. Richards, and P. Winter, *The Steiner Tree Problem*, number 53 in Annals of Discrete Mathematics, North-Holland, New York, 1992.

[2] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, editors, pp. 85–103, New York, 1972, Plenum Press.

[3] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees in graphs," *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[4] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 286–292, June 1993.

[5] X. Lin and L. M. Ni, "Multicast communication in multicomputer networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 10, pp. 1105–1117, October 1993.

[6] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[7] S. Hakimi, "Steiner's problem in graphs and its implications," *Networks*, vol. 1, pp. 113–133, 1971.

[8] S. Dreyfus and R. Wagner, "The Steiner problem in graphs," *Networks*, vol. 1, pp. 195–207, 1971.

[9] P. Winter, "Steiner problem in networks: A survey," *Networks*, vol. 17, pp. 129–167, 1987.

[10] P. Winter and J. M. Smith, "Path-distance heuristics for the Steiner problem in undirected networks," *Algorithmica*, vol. 7, no. 2–3, pp. 309–327, 1992.

[11] H. F. Salama, I. Viniotis, D. S. Reeves, and T.-L. Sheu, "Multicast routing algorithms for high speed networks," Technical Report Technical Report, Dept. of Computer Science, North Carolina State University, Raleigh, NC, May 1994.

[12] R. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[13] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner tree problem in graphs," *Mathematica Japonica*, vol. 24, no. 6, pp. 573–577, 1980.

[14] V. Rayward-Smith and A. Clare, "On finding Steiner vertices," *Networks*, vol. 16, pp. 283–294, 1986.

[15] M. Doar and I. Leslie, "How bad is naïve multicast routing?," in *IEEE INFOCOM*, pp. 82–89, San Francisco, California, March 1993.

[16] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, December 1988.

[17] R. Widyono, "The design and evaluation of routing algorithms for real-time channels," Technical Report TR-94-024, University of California at Berkeley, June 1994.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press (McGraw-Hill), Cambridge, MA (New York), 1990.

[19] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Communications*, vol. 31, no. 2, pp. 343–353, March 1983.

[20] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathemtik*, vol. 1, pp. 269–271, 1959.

[21] D. D. Kandlur, K. G. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 10, pp. 1044–1056, October 1994.