EFFICIENT MULTICAST ROUTING IN WIRELESS ATM NETWORKS

Chao-Tsong FangTsou*, Shiann-Tsong Sheu+, Wu-Hsiao Hsu*, and Ming-Ching Hsiao*

* Department of Computer Science and Information Engineering
+ Department of Electrical Engineering
Tamkang University
Tamsui, Taipei Hsien, Taiwan 251, R.O.C.

Abstract

This paper proposes an algorithm, called Probability-Based Multicast Tree (PBMT) algorithm, for multicast routing in wireless ATM networks. The PBMT is established from a fixed Base Station (BS) by several steps. First, network is modeled as a graph with nodes denote the BS and edges represent links between BS. An edge is added into any two nodes when their coverage areas overlap each other. Then, the degree of each node in the graph is computed. Based on the computed degree of each node, the probability of a mobile user moves to a new node can be determined. In order to identify the importance of each node, the cost of the edge is weighed based on the probabilities of corresponding nodes. Finally, a node with maximal degree is selected repeatedly until a minimal-spanning tree is formed. Nodes do not belong to the multicast members are pruned. Simulation has been done to compare the performance analysis by our algorithm with these of other algorithms.

1. Introduction

ATM (Asynchronous Transfer Mode) has been advocated as an important technology for all types of services and networks. Due to the success of ATM on wired networks, considerable interest has recently begun to focus on the extension of broadband wired ATM into the wireless medium. This extension has been motivated by the existing technologies for supporting mobile users. Examples include cellular telephony [1], [2], personal communication services (PCS) [3], and mobile Internet Protocol [4]. Regardless of the specific protocol, all such technologies must support two fundamental mechanisms [5]: one is to locate users prior to or during connection establishment; the other is to reroute connections when users move. In addition, different scales of mobility, sometimes referred to as *roaming* and *handover* [6], [7] should be supported. Support roaming means an user is allowed to travel to another network domain and attach his host to the network at the foreign location, whereas support handover means that the connections should be maintained when a user moves from one wireless BS to another.

Many new applications require point-to-multipoint or multicast communications, which send the same packet to a group of destinations, in order to consume a minimal amount of resources. A common solution is to model the network as a graph $G=(V, E)$ where the nodes represent routers or switches and the edges represent links between nodes and generate a multicast tree including the source and destination nodes. In a point-to-point or unicast network, we are interested in finding the *shortest path* between a pair of nodes so that an application can send the packet to just one endpoint. On the other hand, in the multipoint problem, we are interested in the minimum cost subtree or multicast tree which connects all destination nodes. How to find this '*shortest*' subtree, which is known as Steiner multicast tree (SMT) [8] in graphs, is a complicated problem. In particular, it has been shown that deriving the SMT is NP-complete [9].

Routing multicast connections in a network is a difficult issue, especially in wireless ATM network because of the special feature of mobility for hosts. Previous works on a multicast tree for a wired network can be classified into two types. First, the destination nodes are unchanged during the transmission period. At this type, we focus on finding a heuristic algorithm to minimize the tree cost to near-optimum solutions. Several polynomial-time heuristic algorithms [10], [11], [12] have shown, at worst, twice the optimum. Secondly, the nodes are allowed to leave or join the multicast tree during the transmission period. Two heuristic algorithms, the greedy algorithm [9] and the weighted greedy algorithm (WGA) [10], are used to

This work is supported by the National Science Council, Republic of China, under contract NSC 87-2213-E-032-010

0-7803-4984-9/98/$10.00 ©1998 IEEE.

solve the dynamic problem. In addition, the work to establish a multicast tree for a wireless network is described on [6]. It adopts the *open signaling* approach, which can be summarized as the use of programmable software architecture for network controls, to construct the multicast tree.

In this paper, a new algorithm to construct a multicast tree for a wireless ATM network is proposed. It is quite different from previous works. The point is focused on the probability of a mobile host moves to a new node. A node with maximal probability is selected repeatedly until the minimum spanning tree is formed. This means the topology changed of the multicast tree is minimal when a user moves from a BS to another. This can save much computation time.

The remainder of this paper is organized as follows. Section II introduces the algorithm. Section III presents the measures of performance. Second IV shows the simulation model and results. Some conclusion remarks are given in Section V.

2. Definition and Analysis

2.1 The graph terminology for multicast routing

Before proceeding, we define the graph terminology formally. Let $G= (V, E)$ be a graph with a finite set of vertices, V, and edge set, E. Also, a cost function, $w : E \rightarrow N$, assigns a positive integer number to each edge in G. For simplicity, we will assume that this graph is undirected. That is, (u, v) and (v, u) are the same edge and $w(u, v)= w(v, u)$. The degree of a vertex is the number of edges incident on it. A path from a vertex u to a vertex v is an alternating sequence of vertices and edges u, (u, v_1), (v_1, v_2), ... , (v_k, v), v and is denoted by $u \xrightarrow{p} v$. In addition, given a set $D \subseteq V$, a mutlicast tree problem is to find a tree T_{OP} that contains all the vertices in D.

2.2 The PBMT algorithm

The most important key point for this algorithm is to find a *common* node, represented as BS with a maximal probability. That is, the *common* node denotes that most of mobile users will move to it. Thus, this node must be selected during the course of construction of the multicast tree. The proposed algorithm is analyzed as follows:

Step 1: The original graph G is constructed based on the overlapping of coverage area of BSs. It is obvious that the BS can also be modeled as a connected graph G, where the nodes represent BS. An edge is added into the graph if an overlapping of coverage area exist between any pair of nodes. For example, a graph shown in Fig. 2 is derived by Fig. 1. $BS(a)$ and $BS(b)$ is connected because there is an overlapping of coverage area. Similarly, there is an edge between $BS(b)$ and $BS(c)$, $BS(b)$ and $BS(d)$, and $BS(d)$ and $BS(c)$.

Step 2: The probability set (PS) that an user moves from a node to all its adjacent nodes is computed. The PS can be determined for each node separately. Let $d(i)$ denote the degree of node i. The PS_i of node i is given by

$$PS_i = \{P_1, P_2, \ldots, P_{d(i)}\},$$

where

$$P_j = \begin{cases} 1, & \text{if node i is root} \\ \frac{1}{d(i)}, & \text{otherwise.} \end{cases}$$

The P_j indicates the probability that user from node i moves to node j. Note that $P_1 = P_2 = \ldots = P_{d(i)}$. This means that user in node i has an equal probability to move to its all adjacent nodes. Consider the example shown in the Fig. 2. For $BS(b)$, $d(BS(b)) = 3$. Therefore,

$$PS_{BS(b)} = \{P_1, P_2, P_3\} \text{ and } P_1 = P_2 = P_3 = \frac{1}{3}.$$

Step 3: The total probability TP of each node is computed. At this step, the breadth-first search (BFS) [9] is used to search the graph G. Suppose the size of $|D|$ is M. There are totally M trees constructed by BFS and each tree is rooted in an individual multicast member. Let T_D denote the set of M trees. Also, suppose node m with degree $d(m)$ is the root of tree T_{member_vertex}, $T_{member\_vertex} \in T_D$, and is the parent of node u with degree $d(u)$, which is the parent of node i. The probability of node u is given by

$$P_u = 1 \times \frac{1}{d(m)}.$$

Similarly, the probability of node i is given by

$$P_i = P_u \times \frac{1}{d(u)} = \frac{1}{d(m)} \times \frac{1}{d(u)}.$$

The TP of node i is equal to the addition of P_i of each node i in every tree and the TP of node u is equal to the addition of P_u of each node u in every tree and so forth. Hence,

$$TP_i = \sum_{T_j \in T_D \text{ and } i \in T_j} P_i = \sum_{T_j \in T_D \text{ and } i \in T_j} \frac{1}{d(m)} \times \frac{1}{d(u)}.$$

Consider the example shown in Fig. 3. The degree of each node is shown in Fig. 3(a). The size of group member is 5. Therefore, there are totally 5 trees derived by BFS from the original graph. The probability P of each node for the tree rooted in s is as follows:

$$P_{root=s} = \begin{bmatrix} s & a & b \\ g & c & d \\ h & e & f \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{12} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{24} \\ \frac{1}{16} & \frac{1}{4} & \frac{1}{72} \end{bmatrix},$$

That is, P_s=1 and we have

$$P_a = 1 \times \frac{1}{d(s)} = 1 \times \frac{1}{4} = \frac{1}{4} \text{ and}$$

$$P_b = P_a \times \frac{1}{d(a)} = \frac{1}{4} \times \frac{1}{3} = \frac{1}{12}.$$

Similarly,

$$P_b = \frac{1}{3} \text{ for } T_a,$$

$$P_b = \frac{1}{3} \text{ for } T_d,$$

$$P_b = \frac{1}{4} \text{ for } T_e, \text{ and}$$

$$P_b = \frac{1}{3} \text{ for } T_f.$$

Therefore, we have

$$TP_b = \frac{1}{12} + \frac{1}{3} + \frac{1}{3} + \frac{1}{4} + \frac{1}{3} = 1\frac{1}{3}.$$

Step 4: The cost of each edge in G is put more or less weight based on the computed total probability. The intent is to identify the importance of an edge (v_i, v_j) corresponded to the pair of nodes. Because the probability from v_i moving to v_j is different from v_j moving to v_i, we need to suppose that nodes are connected by directed edges logically. That is, the cost of edge (i, j) and edge (j, i) is different. Let $w'(i, j)$ and $w'(j, i)$ denote the weighed edge from v_i to v_j and from v_j to v_i, respectively. $w'(i, j)$ and $w'(j, i)$ is given as follows,

$$w'(i,j) = \frac{TP_i}{TP_i + TP_j},$$

$$w'(j,i) = \frac{TP_j}{TP_i + TP_j}.$$

The weighed costs for edges (i, j) and (j, i) are given by

$$c^W(i,j) = c(i,j) \times w'(i,j),$$

$$c^W(j,i) = c(i,j) \times w'(j,i).$$

For example shown in Fig. 4(b), we have

$$w'(s,a) = \frac{0.694}{0.694 + 0.535} = 0.565,$$

$$w'(a,s) = \frac{0.535}{0.694 + 0.535} = 0.435.$$

Thus,

$$c^W(s,a) = 1 \times 0.564 = 0.564,$$

$$c^W(a,s) = 1 \times 0.435 = 0.435.$$

Step 5: The edges incident on a node with maximal *TP* are chosen repeatedly until a minimum spanning tree is reached. For illustration, consider all TP_i shown in Fig. 4(b). Because the maximal probability is c, all edges incident on it are chosen as shown in Fig. 4(c). Next, node b is chosen. Notice that a loop is formed as shown in Fig. 4(d). Since nodes with smaller probability are less visited, these corresponding edges are deleted as shown in Fig. 4(d). Similarly, node d is chosen. After all edges incident on node d are chosen as shown in Fig. 4(e), all nodes have been selected. Therefore, a MST has established.

Step 6: The T_{OP} is obtained by pruning the MST. The non-multicast leave node is deleted from the MST and the parent of it is repeated pruned until a multicast vertex is reached. For example in Figures 4(e) and 4(f), nodes b, g, and h are pruned directly. Note that node c is an internal node.

We now present the algorithms for deriving PBMT formally.

Algorithm for PBMT

Input : an undirected network $G=(V,E)$ with weight function $w : E \rightarrow N$, a set $D \subseteq V$

Ouput : T_{OP} is the found optimal tree

Remark: $T_{BFS(V)}$ denotes the tree established by BFS and rooted in V. $Adj(v)$ denotes all adjacent vertices with non-multicast member v.

1) Initialize T_{OP}, $T_{BFS(V)}$, TP_V, $Adj(v)$, and $NM \leftarrow V - D$
2) Compute the degree of each node in G
3) While $D \neq \varnothing$ do
4) Extract vertex v_i from D
5) Apply BFS to build a tree $T_{BFS(V_i)}$ rooted

in v_i
6) Compute P_{v_i} of all vertices in $T_{BFS(v_i)}$
7) $TP_{v_i} = TP_{v_i} + P_{v_i}$
8) $D = D - \{ v_i \}$
9) End While
10) Repeat
11) Choose the vertex with maximal TP in G
12) Choose all the edges incident from v_{Max} and form a tree T_{OP}
13) If T_{OP} forms loop
14) Repeat
15) Find a vertex v_{Min} adjacent to v_{Max} with a minimal probability and prune the edge from v_{Min} to v_{Max}
16) $Adj(v_{Max}) = Adj(v_{Max}) - \{ v_{Min} \}$
17) Until loop is removed
18) $V = V - \{ v_{Max} \}$
19) Until all vertices in V contained in T_{OP}
20) Repeat
21) Extract the vertex $v_{Non\text{-}member} \in NM$
22) Prune $v_{Non\text{-}member}$ in T_{OP} if it is a leaf node
23) $NM = NM - \{v_{Non\text{-}member}\}$
24) Until $NM = \varnothing$
25) Return T_{OP}

2.3 The complexity of PBMT Algorithm

The operation to compute the degree of each node in G take time $O(|V| \cdot (|V|-1))$. Because the BFS has a worst cast time $O(|E|+|V|)$, from lines 3 to 9 will take time $O((((|E|+|V|) \cdot |V|) \cdot |D|))$. To remove the loop from line 14 to 17, it take time $O(|V| \cdot |V|)$. Thus, the repeat loop from 10 to 19 take time $O((|V|+|E|+|V| \cdot |V|) \cdot |V|)$. Finally, the time spent on lines 20 to 24 is $O((|V|-|D|) \cdot lg|V|)$. Therefore, the running time of the entire algorithm is

$$\begin{aligned} &O\,(\,(|V| \cdot (|V|-1)) + ((((|E|+|V|) \cdot |V|) \cdot |D|)) + \\ &((|V|+|E|+|V| \cdot |V|) \cdot |V|) + ((|V|-|D|) \cdot lg|V|)\,) \\ &= O(|V|^2|D||E| + ((|V|-|D|) \cdot lg|V|)). \\ &= O(|V|^2|D||E|). \end{aligned}$$

3. Cost measure for a multicast tree

The performance of a multicast routing can be considered by the following two different cost measures.

- Total cost C_{CT} : when user in node d moves to an adjacent non-multicast member node v, node v is dynamic joined into the multicast tree by using WGA [10] based on c^w. Let SP denote the shortest path from v to the multicast tree and Adj denote the set containing all the adjacent non-multicast members with d.

$$C_{CT} = \sum_{d \in D} \sum_{v \in Adj} SP_v$$

- Cost of total number of virtual connection (VC) established C_{CV} : let NE denote the total number of established VC when the node v is connected to the multicast tree. The C_{CV} is defined as follows:

$$C_{CV} = \sum_{d \in D} \sum_{v \in Adj} NE_v \,.$$

Our optimization goal is to find a multicast tree for a given set of destinations with the minimal costs of C_{CT} and C_{CV}. In general, the costs can be combined to form a total cost [13]

$$C_{Total} = \alpha\, C_{CT} + \beta\, C_{CV} \,.$$

where α and β are weights assigned to the C_{CT} and C_{CV}, respectively to reflect their importance.

4. Simulation Model and Results

4.1 Simulation Model

The network models used in the simulation are constructed using the approach given in [4]. N nodes are randomly distributed in a rectangular coordinate grid. Each node is located in an integer coordinates and represents a BS, which have some of the characteristics of an actual network. Also, each edge represents a bi-directional physical link and is introduced between pairs of node u, v with a probability that depends on the distance between them. The edge probability is given by

$$P(\{u,v\}) = \beta \exp\frac{-d(u,v)}{L\alpha},$$

where $d(u, v)$ is the distance from node u to v, L is the maximum distance between two nodes, and α and β are parameters in the range (0, 1] and control the characteristics of the graph produced. Increasing β increases the average vertex degree of the graph and increasing α increases the ratio of longer edges relative to shorter ones. Finally, the cost of each edge is set to the distance between its endpoints.

For dynamic joining and leaving, we also use the same probability model as [4] to determine if a node is an addition or deletion from the multicast tree. The function is

$$P_C = \frac{\alpha(n-k)}{\alpha(n-k)+(1-\alpha)k}.$$

Here, P_c is the probability that a node is addition, k is the number of nodes in set D of the current multicast tree, n is the number of nodes in the network, and α is the parameter in (0,1). The value of α is the fraction of nodes in the multicast tree at equilibrium.

4.2 Simulation Results and discussion

A graph contains 200 nodes is considered. The subset D contained nodes selected at random from the 200 node graphs. Both α and β are set to 0.4 and L is set to 50. Also, three different algorithms, PBMT, minimum spanning tree (MST), and shortest path tree algorithm (SP), are simulated on the same network model.

The simulation results are illustrated in figures 5(a) and 5(b). Obviously, the proposed algorithm PBMT obtains a best result. The reason is that the *common* node with maximal *TP* is always chosen. That is, the costs of the edges from all-adjacent nodes of *common* node to *common* node have least C^W. Contrarily, SP performs the worst because the *common* is not chosen. Therefore, the total cost of the connected path will be maximal. Results obtained by MST and SP is very close to each other. Actually, the MST performs slightly superior to SP.

5. Conclusion Remarks

An algorithm based on probability for establishing a multicast tree was proposed in this paper. The total probability of each node in G is computed. Also, the weighed cost is determined based the computed total probability. The multicast tree can be found by using the weighed cost. A best result can be obtained when uses the proposed algorithm. The reason is that the *common* node is contained in the found multicast tree. This not only reduces the C_{CT} but C_{CV}.

An improved version of our algorithm is still under construction in order to improve the time complexity. Also, how to reduce the number of topology changed of the found multicast tree when a host moves to another BS is our important goal. Finally, how to find a better method for analyzing the *common* node is also an important issue for our future research.

REFERENCES

[1] EIA/TIA, "Celluar Radio-Telecommunications Intersystem Operations," *Tech. Rep.* IS-41, rev B, 1991.

[2] M. Rahnema, "Overview of the GSM System and Protocol Architecture," *IEEE Commun. Mag.*, Apr. 1993.

[3] Y. B. Lin and S. K. DeVries, "PCS Network Signaling using SS7," *IEEE Pers. Commun.*, June 1995.

[4] IETF, "IP Mobility Support," *RFC 2002*, Oct. 1996.

[5] Arup Acharya, Jun Li, Bala Rajagopalan, and Dipankar Raychaudhuri, "Mobility Management in Wrieless ATM Networks," *IEEE Commun. Mag.*, pp. 100-109, Nov. 1997.

[6] Lek-Heng Ngoh, Hongyi Li, and Weiguo Wang, "An integrated Multicast Connection Management Solution for Wired and Wireless ATM Networks," *IEEE Commun. Mag.*, pp. 52-59, Nov. 1997.

[7] Osama Kubbar and Hussein T. Mouftah, "Multiple Access Control Protocols for Wireless ATM: Problems Definition and Design Objectives," *IEEE Commun. Mag.*, pp. 93-99, Nov. 1997.

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

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, Cambridge, MA: MIT Press, 1990.

[10] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Selected Area on Communication*, vol. 6, no. 9, pp. 1617-1622, Dec. 1988.

[11] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Math. Japonica*, vol. 24, pp. 573-577,1980.

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

[13] M.H. Ammar, S.Y. cheung and C.M. Scoglio, "Routing multipoint connections using virtual paths in an ATM network," *Proceedings of IEEE INFOCOM'93*, pp. 980-986, 1993.

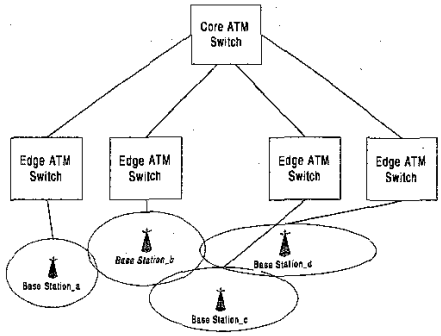

Figure 1.An example of overlapping of coverage areas of BSs in wireless ATM networks.

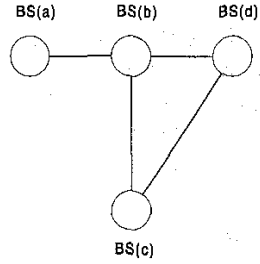

Figure 2. The graph derived by Figure 1

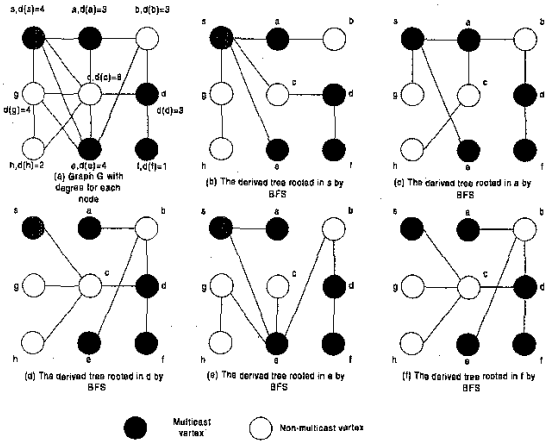

Figure 3. An example of trees rooted in multicast members and derived by BFS.

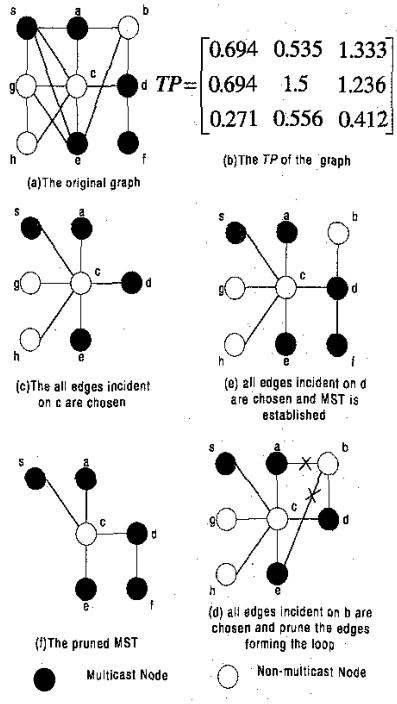

Figure 4. An example of MST and pruned MST.

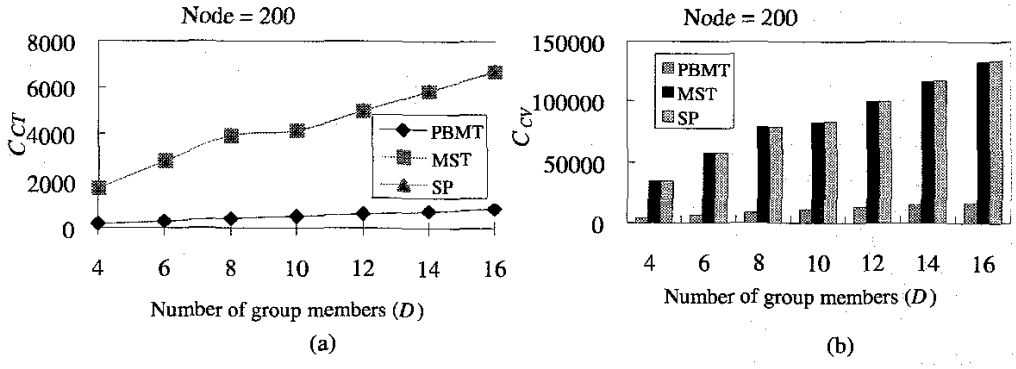

Figure 5. Comparisons of C_{CT} and C_{CV} obtained by PBMT, MST and SP under different multicast group size (D).