

On Finding Feasible Solutions to the Group Multicast Routing Problem

Ning Wang and Chor Ping Low

School of Electrical & Electronic Engineering
Nanyang Technological University
Singapore 639798
Republic of Singapore
{P145139905, icplow}@ntu.edu.sg

Abstract. Group multicast routing problem (GMRP) is a generalization of multicasting whereby every member of the group is allowed to multicast messages to other members from the same group. The routing problem in this case involves the construction of a set of low cost multicast trees with bandwidth requirements for all the group members in the network. The traditional solutions only care for the low cost of multicast trees and sometimes the algorithms will fail during the construction due to the inefficiency of the bandwidth allocation. In this paper we study the feasible solutions to GMRP by proposing a new algorithm to improve the success rate of constructing multicast trees. Simulation results show that our new algorithm performed better in terms of bandwidth utilization and success rate of building multicast trees compared with existing algorithms.

1 Introduction

Nowadays most of the network applications require large amount of bandwidth to deliver multimedia information to multiple destinations simultaneously. One possible way to meet this requirement is via multicast communication. Multicasting allows a source to send information to multiple destinations through a network at the same time. Multicast routing can be solved by building multicast trees over the network topology, and transmit data from the source to all the destinations. A least cost multicast tree is referred to as a *Steiner tree*. The problem of finding a Steiner tree has been proved to be NP-complete by [4].

Group Multicast Routing Problem (GMRP) is a generalization of multicast routing whereby each member node from a group may multicast data to all other members from the same group, i.e. each member node being both an information source and destination. A typical example is that of remote teleconferencing in which every participant is able to send out messages and receive information presented by other group participants concurrently. It can be easily inferred that group multicasting will need higher bandwidth resources than the corresponding “single source” multicast routing.

One possible solution to group multicast routing problem is to use Core Based Tree (CBT) technology [1] with only one shared tree generated, rooting at the “multicasting core” and spans to all the group member nodes with minimal cost. However, using CBT will result in side effects such as network congestion on some tree edges connected to the “core” because data from each source will have to traverse these edges. Too large delay variation is another problem of using CBT algorithm. An alternative method is to create one Steiner tree for each source separately, as [5] and [9] have proposed. However, one common deficiency of the two algorithms lies in the unbalanced data concentration throughout the network. Edges with low cost are always burdened with high traffic loading while other edges are seldom utilized. Extremely, since the two greedy algorithms always select low cost edges to build multicast trees, there exists the situation that the bandwidth of some critical edges are used up and the network becomes disconnected, thus resulting in the failure of the tree construction.

In this paper, we propose a new routing algorithm for group multicast routing problem called Feasible Solutions using adapted TM algorithm for GMRP (FTM). Our main goal is to provide better distribution of data in the network and more important, lower tree failure rate. The new heuristic is based on the adaptation of the *TM algorithm* by H. Takahashi and A. Matsuyama [11]. Extensive simulations are carried out to compare the performance of our proposed algorithm with the traditional algorithms concerning the tree failure rate, variance of traffic loading and the edge saturation which can be used to demonstrate data distribution in the network.

2 Problem Formulation

Generally the network is modeled as a directed graph $G(V, E)$ with node set V ($|V|=n$) and edge set E . Each edge $(i, j) \in E$ has two parameters, namely available bandwidth b_{ij} and cost c_{ij} . We assume that the available bandwidth on each edge is asymmetric in general, i.e. b_{ij} may not be equal to b_{ji} . For each edge (i, j) , b_{ij} is known as the *input bandwidth* from node i to node j . For simplicity, we assume that the cost of each edge (i, j) is symmetric, i.e. $c_{ij} = c_{ji}$. However, our proposed algorithm can be easily adapted to the case in which the cost of each edge is asymmetric. Let $P(u, v)$ denote the shortest path (in terms of cost) from u to v , where $u, v \in V$.

Given a network graph $G=(V, E)$, let $D = \{v_1, v_2, \dots, v_m\}$ be a group of nodes in G , where $D \subset V$ and $|D|=m$. Each node $m_k \in D$ has a bandwidth requirements of B_k units. The group D is called the *multicasting group* and each node in D is called a *member node*. The bandwidth requirement B_k of each node in D is specified by the user. The *group multicast routing problem (GMRP)* is that of finding a set of directed routing trees $\{T_1, T_2, \dots, T_m\}$, one for each member of group D which satisfy the following requirements:

$$\text{minimize } \sum_{k=1}^m \sum_{i, j \in T_k} c_{ij} X_{ij}^k, \quad i, j \in V \quad (1)$$

$$\sum_{k=1}^m B_k X_{ij}^k \leq b_{ij} \quad \text{where} \quad X_{ij}^k = \begin{cases} 1 & \text{if } (i, j) \in E_k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Each tree $T_i=(V_i, E_i)$, where $V_i \subseteq V$ and $E_i \subset E$, represents the tree rooted at node $v_i \in D$ which spans all nodes in D . The tree T_i may also contains some nodes from the set $V-D$, which we call the *relay nodes*. We note here that all leaves nodes in each tree T_i must be member nodes of D . Otherwise, the relay nodes and their corresponding links can be deleted from the tree resulting in a lower cost solution.

The first constraint ensures that the total cost of generated trees is minimized while the second constraint is to ensure that the total bandwidth utilized on each link does not exceed its available bandwidth. Since GMRP is the generalization of single Steiner tree problem, it is obviously NP-complete. A set of trees $\{T_1, T_2, \dots, T_m\}$ which satisfy constraint (2) is called a *feasible solution* to the group multicast routing problem, i.e., our goal is to achieve the maximum success rate of building multicast trees. In this paper we assume that when each edge is joined into the tree, one unit of its bandwidth will be consumed. An edge is said to be *saturated* if its consumed bandwidth has reached the bandwidth capacity of this edge.

Fig. 1 shows a network model as we have defined. The number in the middle of the edge is cost and the numbers near the arrows at the two ends of an edge represent the bandwidth capacity of the edge in the direction of the arrows. Note that the nodes in gray color are relay nodes and the rest are multicasting members in the network, i.e., $D=\{0, 1, 2, 3\}$.

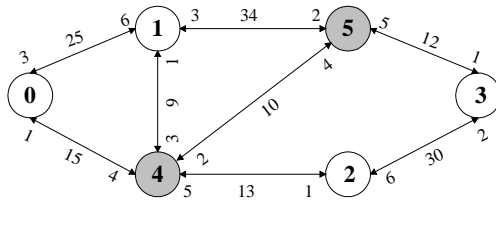


Fig. 1 Network model

3 Previous Works

Single source based multicast routing with minimum cost is referred to as the Steiner tree problem. L. Kou, G. Markowsky and L. Berman proposed a heuristic algorithm called KMB algorithm to achieve the close-to-optimal solutions [8]. The KMB algorithm begins by computing shortest paths between each pair of member nodes. Following that, the closure graph G' which contains only nodes in D is constructed. The closure graph G' is one in which each node in D is connected to all other nodes in D by the shortest paths. Thus, each edge (a,b) in the closure graph correspond to a shortest path between nodes a and b in the original graph G . The next step is to construct minimum spanning tree of G' . A greedy approach is employed to add edges

to the partial tree until all member nodes in D are included. In each iteration of this approach, a least cost edge that connects a node in the partial tree to one that is not in the partial tree is selected to be joined. Once the tree is constructed, all edges in the tree are expanded into the shortest paths which they represent in the original network.

H. Takahashi and A. Matsuyama proposed TM algorithm [11] by computing the shortest paths from all the nodes in the network to every group members. Empirical study [9] showed that TM algorithm performs better than KMB according to total cost due to its more powerful capacity of explore shortest paths. When one member node is about to join into the Steiner tree, not only the shortest paths from the member nodes already in the tree but also those from some relay nodes can be obtained to build the Steiner tree.

According to the problem of group multicast routing, multicast tree will be built on each of the member nodes and the bandwidth requirement is taken into consideration. Jia and Wang [5] first proposed a heuristic algorithm, which we call *Jia & Wang's Algorithm*, which adopts some form of coordinated strategy called *overhead comparison* to generate a set of multicast trees. Their algorithm is based on an adaptation of KMB Steiner tree heuristic. Later on another heuristic called Group multicast routing with adapted TM algorithm (GTM, [9]) was proposed. As we have mentioned above, due to the more powerful capacity of TM algorithm to explore shortest paths, GTM correspondingly provides better performance in total cost than *Jia & Wang's algorithm*. Moreover, GTM also obtains lower tree failure rate compared with *Jia & Wang's algorithm* [9].

It should be noted that the problem of finding multicast tree with minimum cost is NP-complete [4], being a generalization of this problem, none of the solutions to GMRP we have mentioned can find the optimal solution in polynomial time. We have also found that even if there exist some solutions to building multicast tree for each group member in the network, cases are that neither *Jia & Wang's algorithm* nor GTM can find any. In this paper, we will mainly focus on how to achieve the maximum success rate of building a set of multicast trees, i.e., if there do exist solutions to the problem, our proposed algorithm can always find out at least one that is feasible.

4 Proposed Algorithm

4.1 Breadth First Search (BFS) Tree

Observe GTM and other similar algorithms which adopt greedy strategy to build multicast trees, edges with low cost are in higher priority to be selected each time so that those edges also having low bandwidth capacity could be consumed up and become saturated before all the multicast trees are generated. If unfortunately some critical edges are unavailable in the network, all the group members are not located in one connected sub-graph, the algorithm fails. One sufficient condition to prevent the network from being disconnected is to guarantee that there are no saturated edges in the network. Based on the above analysis, FTM always finds edges with large bandwidth capacity to build multicast trees. Following this scenario, we first find the "widest" path from all the nodes to every group members, as TM algorithm deals with

cost in the network. To achieve this, we use breadth first search (BFS) to detect the paths with maximum bandwidth capacity to the group members. One BFS tree is built rooted at each multicasting group member and contains the widest path from all the other nodes in the graph. When each of the nodes is visited during BFS procedure, the edges from all its neighbors (except its predecessor) to the node itself are examined, and the one with maximum bandwidth capacity is selected, with the corresponding neighbor included into the search tree. At the same time, the bandwidth bottleneck and the total cost of the path to the root is also recorded. If there are two or more neighbors connected by the edges with the same bandwidth capacity, the one with least cost is chosen. In order to implement the breadth first search, a queue is used to accommodate the nodes to be visited. Fig. 2 shows how the BFS tree is generated for node 0 in the graph given by Fig. 1. Notice that the two numbers in the bracket represent the total cost and bandwidth bottleneck of the path from this node to node 0 respectively.

Starting from node 0, since the two of its neighbors node 1 and node 4 have never been visited, they are joined into the BFS tree with bandwidth bottleneck 3 and 1 respectively as Fig 2(a) shows. Next we come to examine node 1's neighbors. Node 5 is joined into the BFS tree by connecting node 1, its bottleneck to the source is $\min(\text{bottleneck}[1], b(5,1))$, which is also 3. As to node 4, $\min(\text{bottleneck}[1], b(4,1))$ is 1, the same with $\text{bottleneck}[4]$, however the total cost of path $4 \rightarrow 1 \rightarrow 0$ is 34 which is greater than the cost of edge $(4,1)$, so node 4 is remained directly connecting to node 0. After all of node 1's neighbors have been visited, we come to examine node 4's neighbors. Since the bottleneck of both node 1 and node 5 are greater than that of node 4, they remain unchanged. As to node 2, since it has never been visited, it is joined into the tree by connecting to node 4 (Fig. 2(b)). Then we will come to check node 5's neighbors. Node 3 will be added into the BFS tree with its bottleneck being $\min(\text{bottleneck}[5], b(3,5))=3$ (Fig. 2(c)). Since $\min(\text{bottleneck}[5], b(4,5))=3$ is greater than $\text{bottleneck}[4]$, node 4 will switch its path from node 0 to node 5, also notice that the path cost and bottleneck of its current successor node 2 will also be updated to 82 and 3 respectively, as Fig. 2(d) illustrates. Next the neighbors of node 2 are visited. Since the path $3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 0$ and $3 \rightarrow 5 \rightarrow 1 \rightarrow 0$ have the same bandwidth bottleneck while the cost of the latter is smaller, node 3 will not switch its connection from node 5 to node 2. Finally the neighbors of 3 are checked and we can find that there is no need to modify any connections. After the breadth first search is finished, Fig. 2(d) is the final BFS tree for node 0 with the records of the bandwidth bottleneck and total cost of the paths from all the other nodes to the root.

4.2 Necessary Conditions for Existence of Feasible Solutions.

While the problem of finding feasible solutions for GMRP is NP-complete, there exists some criteria which could be used to determine if a feasible solution does exist for a given problem instance. Two of such criteria are given in the following lemma. Here we still assume that one unit of bandwidth is consumed when each edge is included in the multicast tree.

Lemma: There is no feasible solution for GMRP if :

- (i) all member nodes do not belong to the same connected component; or
- (ii) the total input bandwidth capacity for some member node v is less than $m-1$.

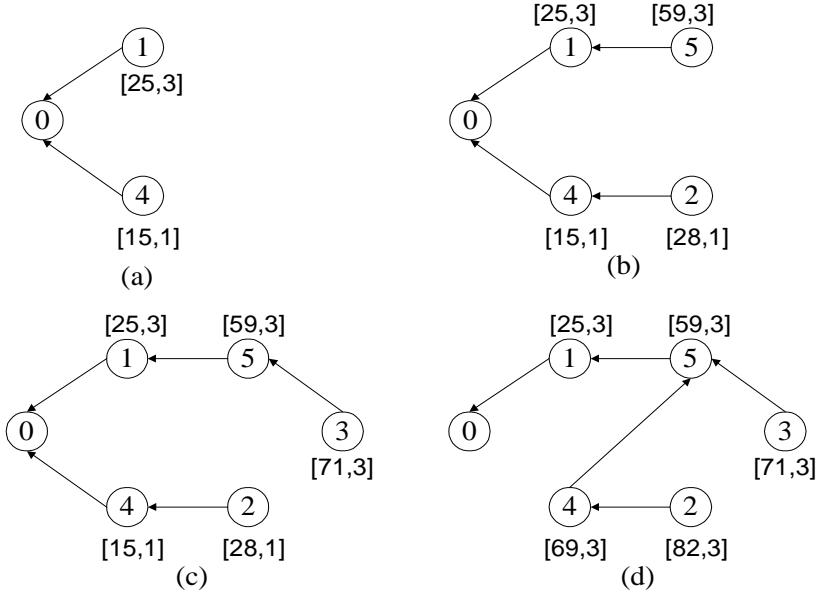


Fig. 2 Generation of BFS tree rooted at node 0

Proof: A feasible solution for GMRP is comprised of a set of multicast trees $\{ T_u : u \in D \}$. Each tree T_u is rooted at member $u \in D$ and spans all other member nodes in the network. If member nodes are located in two or more components, then it would not be possible to construct a multicast tree that connects one member node to all other member nodes in the network. It follows then that criteria (i) is a necessary condition for the existence of feasible solutions for GMRP.

In any feasible solution, each member node v will be receiving information from the other $m-1$ member nodes via $m-1$ multicast trees T_u , for each $u \in D - \{v\}$. Thus, the total input bandwidth of each member node must be at least $m-1$.

This lemma can be used as an *early-abort test* to determine in advance if a feasible solution exists before attempting to find any solutions for GMRP.

4.3 Feasible Solution to GMRP

After m BFS trees are built rooted at each of the group members, the detail information of the widest path from all the other nodes to this group member is available. Similar to the TM algorithm, our next step is to build multicast trees for the group members based on the path information obtained by BFS procedure. It should be noted that bandwidth is a concave metric in the network model, i.e., the bandwidth capacity of a path is decided by the edge with minimum available bandwidth along this path. Since the bandwidth bottleneck of the path is recorded in BFS procedure, we will use the path with maximum bandwidth bottleneck to join all the group

members into the current multicast tree. Different from TM algorithm, in this step paths are selected according to the bandwidth bottleneck instead of the edge cost. Of course, if two or more paths with the same bandwidth bottleneck are found at the same time, the one with least cost is selected. Fig. 3 gives out the psudocode of the feasible solution to GMRP.

Procedure BFS:

Input: graph $G(V, E)$, source;

Output: Maximum bandwidth path to source;

begin

```

for  $i=1$  to  $n$  do                                     /* Initialization */
    if (  $i <> source$  ) then
        node[ $i$ ].visited=0;
        distance[ $i$ ]=infinite; bottleneck[ $i$ ]=-1; predecessor[ $i$ ]=-1;
    else
        distance[ $i$ ]=0; bottleneck[ $i$ ]=infinite; predecessor[ $i$ ]=-1;
    end for;
     $T = \emptyset$ ;
    InitQueue(Q);
    for  $i=1$  to  $n$  do
        if ( node[ $i$ ].visited=0 ) then
            EnQueue(Q,  $i$ );
            while ( not QueueEmpty(Q) ) do
                 $u = DeQueue(Q)$ ;
                 $u.visited=1$ ;
                for each  $v$  such that  $(v,u) \in E$  do           /* Find the neighbor with
                                                                maximum bandwidth edge */
                    if (  $\min(\text{bandwidth}(v,u), \text{bottleneck}[u]) > \text{bottleneck}[v]$  ) or
                        (  $\min(\text{bandwidth}(v,u), \text{bottleneck}[u]) = \text{bottleneck}[v]$  ) and
                        (  $\text{distance}[v] > \text{distance}[u] + \text{cost}(v,u)$  ) then
                         $(v,u) \rightarrow T$ ;
                        bottleneck[ $v$ ] =  $\min(\text{bandwidth}(v,u), \text{bottleneck}[u])$ ;           /* Record the bandwidth
                                                                bottleneck to the source node */
                        predecessor[ $v$ ] =  $u$ ;
                        distance[ $v$ ] =  $\text{distance}[u] + \text{cost}(v,u)$ ;
                        for each  $w$  such that  $(v,w) \in T$  do
                            update  $w$ 's bottleneck and distance;
                        end if;
                    for each  $v$  such that  $(v,u) \in E$  do
                        if node[ $v$ ].visited=0 then
                            EnQueue(Q, $v$ );
                        end while
                    end if;
            end while
        end if;
    end; (Procedure BFS)

```

Procedure FTM

Input: $G(V,E)$, D ;

Output: A set of multicast trees T_v , for each $v \in D$.

begin

```

if  $G$  is not connected then stop;
for each member node  $v \in D$  do                             /* early-abort detection */
    if total input bandwidth for  $v < m-1$  then stop;
for each member  $i \in D$  do
    for each member  $i \in D$  do BFS( $G, i$ );
        set  $T_i = (V_i, E_i)$ ,  $V_i = \{i\}$ ,  $E_i = \emptyset$ , copies=1;

```

```

while (copies<m)
  find a path  $P(v, u)$  where  $v \in V_i, u \in D - V_i$ , such that /* Select the path with
   $bottleneck(v, u) = \max_{i \in V_i, j \in D - V_i} bottleneck(i, j)$  maximum bandwidth bottleneck */
  if there exist more than one  $P(v, u)$  with the same bottleneck then
    select the path with least cost;
     $V_i = V_i \cup \{ \text{all nodes in } P(v, u) \text{ except } v \}$ ;
     $E_i = E_i \cup \{ \text{all edges in } P(v, u) \}$ ;
    copies=copies+1;
  end while
  update bandwidth status;
end for
end; (Procedure FTM)

```

Fig. 3 Pseudocode of feasible solution to GMRP

4.4 Time Complexity Analysis

In the function of BFS, it takes $O(n+e)$ to visit each node in the graph using breadth-first-search where e is the number of edges in the graph. The time complexity of visiting all the neighbors of each node is $O(n)$ while updating the distance and bandwidth bottleneck of successors takes $O(n)$. Hence the time complexity of BFS is $O((n+e)n^2)$. In the function of FTM, BFS dominates most of the computing time when building multicast trees for each group member. On the whole, building m multicast trees will take $O(m(n+e)n^2)$ which is actually the time complexity of our proposed FTM algorithm.

4.5 An Example

We also take the network model shown in Fig. 1 as our example. Since node 0,1,2,3 are multicasting group members, one multicast tree will be built for each of them respectively. First we use the traditional greedy method to solve the problem. Fig. 4(a) shows the Steiner tree rooted at node 0 with optimal total cost. However, after tree 0 is finished, we find that edge (4, 2) and (5, 3) have been saturated, resulting in the separation of node 2, 3 from the rest part of the graph, as shown in Fig. 4(b). Definitely there will be no feasible solutions to building the rest of the multicast trees due to the fact that the network has been disjointed.

In our proposed algorithm, the multicast trees are built based on the path information provided by the BFS trees rooted at each group member. In order to generate each multicast trees, two tables must be available: the table of bandwidth bottleneck and the table of path cost, both of the tables are $m*n$ in size. Fig. 5 shows the two tables needed to build multicast tree 0. In the first table, element (i, j) represents the bandwidth bottleneck from node (j) to group member (i) and in the other, element (i, j) represents the total cost of the path from node (j) to group member (i) . Notice that the data in the i th row of both tables are obtained by the BFS

tree rooted at node i . Starting from node 0, we check the first column of table 1 and find that among the bottlenecks of all the nodes, node 1 has the widest path to node 0 which is 6, so it is selected into the multicast tree rooted at node 0. After node 1 is joined into the multicast tree, we find that both node 2 and node 3 have the bandwidth bottleneck of 1 from the partially built tree which currently contains node 0 and node 1 (shown as the left-down 2 by 2 elements in table 1), so table 2 is checked and node 2 is selected into the tree by taking the path $1 \rightarrow 4 \rightarrow 2$ whose cost is 22 which is less than all the other possible paths from node 3 (58 from node 0 and 46 from node 1). Finally node 3 is added into the multicast tree by taking the path $2 \rightarrow 3$, as shown in Fig. 6(a). Similarly the multicast trees rooted at the rest of the members are built as Fig. 6(b),(c),(d) shows.

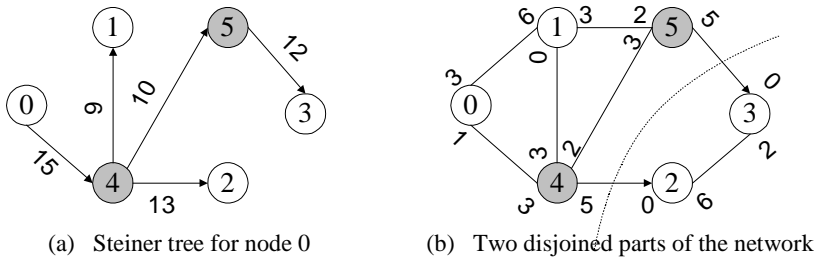


Fig. 4 Traditional algorithms for GMRP

	0	1	2	3	4	5
0	--	3	3	3	3	3
1	6	--	3	3	3	3
2	1	1	--	6	1	1
3	1	1	2	--	1	1

	0	1	2	3	4	5
0	--	25	82	71	69	59
1	25	--	57	46	44	34
2	28	22	--	30	13	23
3	58	46	30	--	43	12

(1) Table of bandwidth bottleneck

(2) Table of path cost

Fig. 5 Two tables for building multicast tree 0

5 Simulations

5.1 Random Graph Generation

To ensure that simulation of the effects of different routing algorithms are fairly evaluated, random graphs with low average degrees, which better represent the topologies of common point-to-point network, e.g. NSFNET, are constructed. The nodes are randomly placed on a rectangular grid and nodes are connected with the probability function:

$$P(u, v) = \lambda \exp\left(\frac{-d(u, v)}{\rho L}\right)$$

where $d(u,v)$ is the distance between node u and v and L is the maximum possible distance between any pair of nodes. The parameters λ and ρ ranging $(0,1]$ can be modified to create the desired network model. For example, a large value for λ gives nodes with a high average degree, and a small value for ρ increases the density of shorter links relative to longer ones. In our simulation, λ and ρ are set to 0.4 and 0.2, respectively. We use the distance between node u and v as the cost of the edge. The bandwidth capacity of each edge is allocated using the following function:

$$B(u,v) = B_m + (-1)^k \times r \text{ mod } B_m$$

where B_m is the given mean bandwidth while k and r are random numbers. Using this function no bandwidth with negative value will be generated and the bandwidth capacity of all edges will range from 1 to $2B_m - 1$. Graphs are generated and tested until member nodes are connected in a single sub-graph.

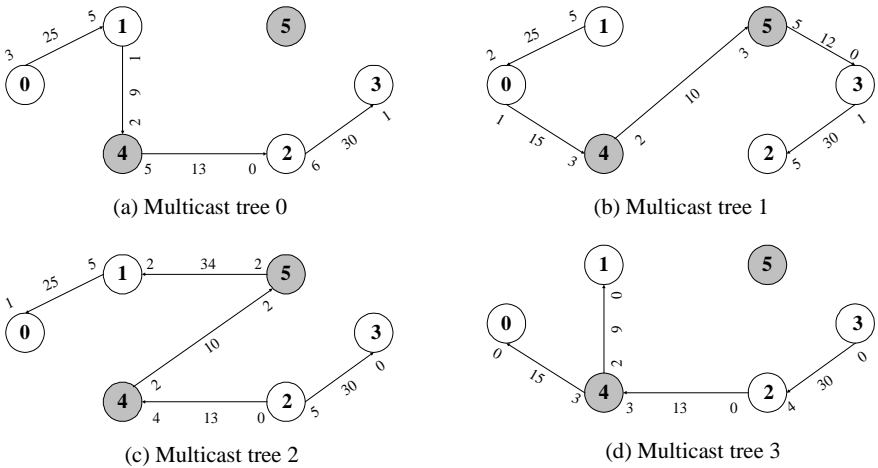


Fig. 6 A feasible solution

5.2 Algorithm Performance Study

In our simulation, we will mainly compare the performance between the proposed FTM algorithm and such existing heuristics as *Jia & Wang's algorithm* and GTM.

First we consider the data distribution in the network. We define link loading factor L for each edge:

$$L_{ij} = \frac{\sum_{k=1}^m B_k X_{ij}^k}{b_{ij}}$$

and network loading factor:

$$\bar{L} = \frac{\sum_{(i,j) \in E} L_{ij}}{N_e}$$

where N_e is the number of edges in the network. From definition, we can see that L_{ij} is actually the rate between the consumed bandwidth and the bandwidth capacity of this edge. \bar{L} is the average loading of all edges in the network. If the variations of L_{ij} between each edge are small, we can draw the conclusion that the data traffic is well distributed in the network.

Fig. 7 shows the network loading factor over the group size. The network size is fixed at 100 and the mean bandwidth is fixed at 30. From Fig. 7 we can see that the network loading factor increases as the group size grows larger. This phenomenon is expected because with the increase of the group size, not only more multicast trees will be built, but the size of each tree will become larger. We also find that the network loading factor of FTM is higher than that of *Jia & Wang's algorithm* and GTM by a small scale.

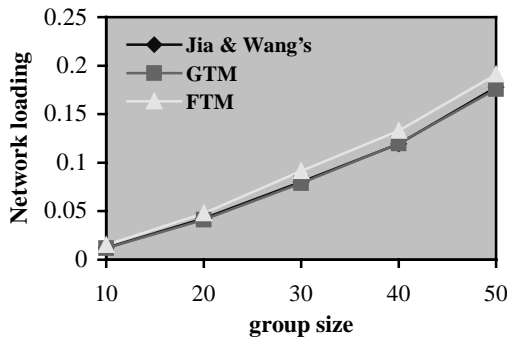


Fig. 7 Traffic loading factor vs. group size

Since the network loading factor is comparable between the two algorithms, we will continue to study the data distribution in the network by computing the variance of loading factor δ . We define:

$$\delta = \sum_{(i,j) \in E} P_{ij} \times (L_{ij} - \bar{L})^2$$

where P_{ij} is the probability that edge (i,j) is selected into the multicast trees. P_{ij} can be expressed as the number of times that edge (i,j) is selected into the multicast trees over the total times that all edges are selected. Since one unit of bandwidth is consumed when each edge is joined, we can express the probability with the consumed bandwidth as the following:

$$P_{ij} = \frac{\sum_{k=1}^m B_k X_{ij}^k}{\sum_{(u,v) \in E} \sum_{k=1}^m B_k X_{uv}^k}$$

Fig. 8 shows the variance of loading factor over group size. Notice that all the external conditions are same as those of Fig. 7. From the figure we can see that the variance of loading factor of FTM is significantly lower than that of the other two

algorithms, typically when the group size is 50, variance of FTM is lower than that of GTM by 67.9% and *Jia & Wang's algorithm* by 69.2%. Since the traffic loading are very close to each other (see Fig. 7), low traffic loading variance reflects better distribution of data in the network. In FTM, the total traffic is evenly allocated to a large number of network links and each of them has a moderate traffic loading. On the contrary, edges with low cost are burdened with high traffic loading while those with high cost are seldom utilized in *Jia & Wang's algorithm* and GTM, resulting in the unbalanced data distribution.

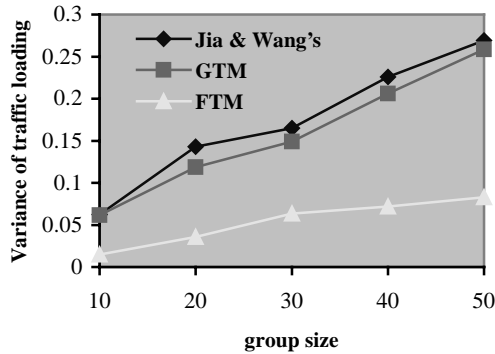


Fig. 8 Variance of traffic loading vs. group size

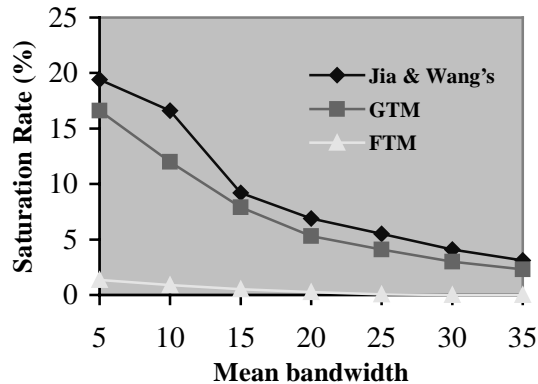


Fig. 9 Saturation rate vs. mean bandwidth

Next we will study the relationship between mean bandwidth and edge saturation rate. Edge saturation rate is defined as the number of saturated edges over the number of all edges in the network after the algorithm terminates. Fig. 9 shows the saturation rate over the mean bandwidth. The network size is fixed at 100 and the number of group members is 30 while the mean bandwidth ranges from 5 to 35 in steps of 5. From the figure we can see that all of the three algorithms have fewer saturated edges as the mean bandwidth increases. However, the saturation rate of FTM is obviously lower than the other two, e.g. when the given mean bandwidth is as low as 5, the edge

saturation rate of GTM reaches 17% while that of FTM is only 1.35% under this condition. The phenomenon can be explained by the fact that the edges with lower bandwidth capacity have lower priority to be selected to build the multicast trees so that the bandwidth is conserved. By studying the edge saturation condition, we can also infer that FTM can provide better distribution of data in the network. Since the edges with large bandwidth capacity is selected each time, their priority decreases as the bandwidth is consumed step by step while the priority of other edges will become higher. Following this scenario, a dynamic balance among the edge selection is achieved and data can be more evenly distributed in the network.

Fig. 10 shows the tree failure rate over the mean bandwidth. Similarly, the network size is also fixed at 100 and the number of group members is 30 while the mean bandwidth ranges from 5 to 35 in steps of 5. Observe that tree failure rate of both algorithms converges to zero as the mean bandwidth increase. This phenomenon is expected as the chance of finding paths with sufficient bandwidth capacity increases as the mean bandwidth increases. From Fig. 10 we can also see that the tree failure rate of FTM is significantly lower than that of *Jia & Wang's algorithm* and GTM. The reason for this is due to the low edge saturation rate as we have illustrated above. When the given mean bandwidth is high, each of the edges is not easy to be saturated. In this case, as long as all the group members are located in one connected graph in the initial condition which we will examine before each run, the topology of the graph won't change if no edges are saturated and there must always exist solutions.

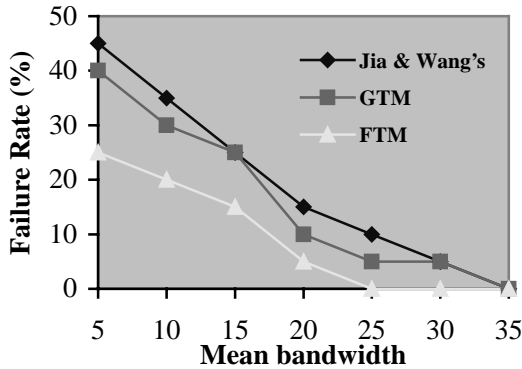


Fig. 10 Tree failure rate vs. mean bandwidth

As we have mentioned in Section 2, our proposed algorithm focuses on the feasible solution to GMRP, and the factor of cost is not our main concern. As the algorithm proceeds, only if two or more paths having the same bandwidth bottleneck are met will we consider cost efficiency and choose one path with least cost. Fig. 11 shows the performance according to total tree cost vs. group size. From the figure we can find that FTM has significantly higher cost than the other two greedy heuristics, typically when the group size reaches 50, the total cost of *Jia & Wang's algorithm* and GTM are 60.9% and 56.2% that of FTM respectively. How to provide efficient tradeoff between total tree cost and failure rate is one of the topics that our future works will focus on.

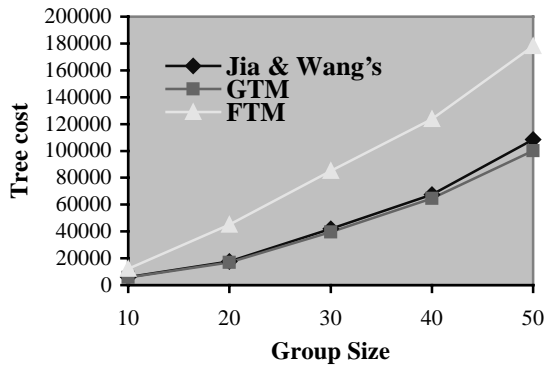


Fig. 11 Total tree cost vs. group size

6 Conclusions

In this paper, we examine feasible solutions to group multicast routing problem with bandwidth requirements. In order to provide better multicasting data distribution and lower tree failure rate, we proposed a new algorithm called feasible solution to GMRP using adapted TM algorithm (FTM). Results from our empirical study shows that our new algorithm performs better in terms of data distribution and edge saturation rate compared to other greedy algorithms. In addition, our simulations also show that FTM has a higher percentage of success in finding solutions to GMRP due to the efficient edge selecting strategy.

Currently our proposed algorithm mainly concerns on the success rate of building multicast trees in the network. Cost does not contribute much to the edge selections so that the resulting trees are of higher cost than other greedy algorithms. In addition, the proposed algorithm is based on a centralized fashion, i.e., the network topology is assumed to be obtained by all the nodes in the graph, and we will later move on to the distributed fashioned algorithms for GMRP which will be more feasible for “actual” networks. One of our future research directions will also focus on the network model with guaranteed QoS and provide trade-off between success rate and tree cost.

References

- [1]. Ballardie T, Francis P., Crowcroft J., “Core Based Trees (CBT)”, Proc. SIGCOMM '93 Conf., ACM, (1993) 85-95.
- [2]. F. Bauer. A. Varma. “Distributed algorithms for multicast path setup in data networks”, IEEE/ACM Trans. Networking 4(2), (1996) 181-191.
- [3]. Dirceu Cavendish, Aiguo Fei, Mario Gerla and Raphael Rom, “On the construction of low cost Multicast Trees with Bandwidth Reservation”, UCLA Technical Report, Doc ID: ncstrl.ucla_cs/970043.

- [4]. M.R. Garey, D.S. Johnson, “*Computers and Intractability—a Guide to the Theory of NP-completeness*”, Freeman, New York, 1979.
- [5]. X. Jia and L. Wang, “*Group multicast routing algorithm by using multiple minimum Steiner trees*”, *Computer Communications* 20, (1997) 750-758.
- [6]. R. M. Karp, “*Reducibility among Combinatorial Problems*” in Miller and Thatcher (Eds.), *complexity of Computer Computations*, Plenum Press. New York, 1972, 85-103.
- [7]. V. P. Kompella, J. C. Pasquale, G. C. Polyzos. “*Multicast routing for multimedia communication*”, *IEEE/ACM Tran. On Networking* 1(3), (June 1993) 286-292.
- [8]. L. Kou, G. Markowsky and L. Berman, “*A fast algorithm for Steiner trees*”, *Acta Informatica* 15, (1981) 141-145.
- [9]. C. P. Low, N. Wang “*An efficient algorithm for group multicast routing with bandwidth reservation*”, *Proc. IEEE International conf. on networks (ICON'99)*.
- [10]. V.J. Rayward-Smith and A. Clare, “*On Finding Steiner Vertices*”, *Networks*, Vol. 16, no. 3, (1986), 283-294.
- [11]. H. Takahashi and A. Matsuyama, “*An approximate solution for the Steiner problem in graphs*”, *Math. Japonica* 6, (1980) 573-577.
- [12]. B. M. Waxman, “*Routing of multipoint connections*”, *IEEE J. Selected Areas Commun.* 6(9) (1988) 1617-1622.