한국경영과학회지
제41권 제2호
2016년 5월
http://dx.doi.org/10.7737/JKORMS.2016.41.2.053

다중 홉 무선 네트웍에서 지연을 고려한 멀티케스트 루팅

손희석 · 이채영†
한국과학기술원 산업공학과

Delay Guaranteed Bandwidth-Efficient Multicast Routing in Wireless Multi-hop Networks

Hee-Seok Sohn · Chae Y. Lee
Department of Industrial and Systems Engineering, KAIST

■ Abstract ■

Static wireless multi-hop networks, such as wireless mesh networks and wireless sensor networks have proliferated in recent years because of they are easy to deploy and have low installation cost. Two key measures are used to evaluate the performance of a multicast tree algorithm or protocol : end-to-end delay and the number of transmissions. End-to-end delay is the most important measure in terms of QoS because it affects the total throughput in wireless networks. Delay is similar to the hop count or path length from the source to each destination and is directly related to packet success ratio. In wireless networks, each node uses the air medium to transmit data, and thus, bandwidth consumption is related to the number of transmission nodes. A network has many transmitting nodes, which will cause many collisions and queues because of congestion. In this paper, we optimize two metrics through a guaranteed delay scheme. We provide an integer linear programming formulation to minimize the number of transmissions with a guaranteed hop count and preprocessing to solve the aforementioned problem. We extend this scheme not only with the guaranteed minimum hop count, but also with one or more guaranteed delay bounds to compromise two key metrics. We also provide an explanation of the proposed heuristic algorithm and show its performance and results.

Keywords : Wireless Multi-Hop Network, Multicast Routing, Minimum Node Tree, Integer Linear Programming, Preprocessing

논문접수일 : 2015년 11월 04일　논문게재확정일 : 2016년 01월 19일
논문수정일(1차 : 2016년 01월 06일)
† 교신저자, cylee@kaist.ac.kr

1. Introduction and Motivation

A wireless multi-hop network provides a means of communicating data among a set of nodes that have the ability to receive and send packets independently using wireless links. A given node cannot transmit data directly to other nodes because of its limited transmission range. In this paper, we focus on static wireless multi-hop networks, such as wireless sensor networks (WSN) and wireless mesh networks (WMN) [1].

Multicast routing employs directed transmission tree from one source to many destinations. Multicast routing protocols try to make a tree by minimizing the total edge cost. Multicast routing in communications is considered same as the Steiner tree problem (STP), because the total edge cost needs to be minimized in both problems [7]. However, in wireless networks, a multicast tree is not identical to the STP due to the wireless multicast advantage (WMA). The wireless multicast advantage shows that one node can transmit to all neighbor nodes in a wireless network due to the omnidirectional characteristic of wireless broadcast. Thus, this unique characteristic should be considered in wireless multicast routing [4, 14].

Wireless multicast routing tree problems usually deal with end-to-end delay [10], and the average cost of each path from the source to the destinations [15, 16]. Delay is one of the most important quality of service measures to evaluate multicast routing performance. In this paper, we assume that each node has no ability to control power and the transmission range is fixed. We then construct a multicast tree with guaranteed delay, or hop count. Also, as mentioned above, WMA should be considered to minimize the average path length of the tree.

One of the prominent multicast routing trees is the shortest path tree (SPT), which guarantees the minimum path length of all source-destination pairs. It is simply the union of the shortest path constructed by Dijkstra's algorithm for each source-destination pair. In [9], the procedure and the shape of the SPT are explained. SPT can be easily exploited in protocols and algorithms because simple request and reply packet flows can detect the shortest path of every node pair. In wireless ad hoc networks, there are also hundreds of distributed protocols using the SPT. Two notable protocols among them are the on demand multicast routing protocol (ODMRP) [8] and the multicast ad hoc on-demand distance vector routing protocol (MAODV) [11].

Recently, [3, 9, 12, 13] suggested that reduction of bandwidth consumption in wireless networks is another key factor to affect the performance of multicast routing trees. Bandwidth consumption can be measured by counting the number of transmissions in a network and each transmission has its own bandwidth requirement. The number of transmissions is an important measurement in wireless environments because congestion and collisions originate from numerous transmissions in heavy traffic conditions, decreasing the total throughput of multicast routing. This also causes a low success packet delivery ratio and high energy consumption. In response, [12] proposed a new multicast tree called the minimum transmission node tree (MNT). In [9], an example and the shape of the MNT are discussed. A bandwidth delay product based multicast routing scheme [3] in MANET is proposed to enhance the reliability with ring mesh backbone.

The MNT, however, has much longer path length than the SPT, because the algorithm does not consider the delay of each path or hop count. Consequently, the SPT deals with the average path length and not the number of transmission nodes whereas the MNT is based on the opposite principle. These two metrics are not in the role of cooperators, but rather are in a trade-off relation.

We therefore focus on this trade-off and propose a novel algorithm to provide an optimal compromise between these two factors in the latter part of this paper.

2. Related works

The latest simulation research [3] compares the performance of two multicast routing tree algorithms, SPT and MNT. In the simulation results, MNT has a smaller packet success ratio than SPT and the gap between them becomes greater as the node density or traffic load becomes higher in a single multicast tree scenario. This demonstrates that the hop count is much more important than other metrics in wireless networks, because each hop has lower success probability to transmit data and the total packet success ratio of a path is expressed by multiplying each ratio of hops in the path. Unlike a single multicast tree scenario, the opposite result is generated in a multiple multicast trees scenario. The remarkable change is caused by the total number of transmissions in network topology. Whenever a new multicast group is added to a network, a greater number of transmissions is produced by SPT compared with MNT and the excessive transmissions cause serious problems such as congestion and collisions, thereby reducing the packet success ratio and total throughput. These simulation results reveal that the average path length and the number of transmissions are very important metrics for a multicast routing tree in wireless networks, but the delay is slightly more significant because general networks have moderate density and traffic loads.

In [12], the authors proved NP-Completeness of MNT and proposed a node minimized algorithm. They showed that, similar to minimum Steiner tree problems, MNT is also a NP-Complete problem, because this problem can be represented by the well-known vertex cover problem [5]. In wireless networks, the simple sum of edge costs is useless due to the wireless multicast advantage. Thus an algorithm to construct a node minimized multicast routing tree is proposed [5]. The main idea underlying the algorithm is to cover as many destinations as possible without considering the path length. If one node covers many destinations, one transmission from the node will satisfy all destinations in its transmission range and the process finally constructs a MNT. More sophisticated algorithms have also been developed, including geographical multicast routing (GMR) [13]. It uses the information of geographical location by GPS in each sensor node. Neither of the MNT algorithms considers the average path length. Also, we previously saw that delay plays a major role in wireless multi-hop networks. Thus, we can state that MNT has a problem in delay minimization.

The multi-channel multicast algorithm (MCM) [18] deals with the trade-off between the two major metrics, the number of transmissions and the average path length. However, many ties occur with the algorithm due to no tie breaking mechanism. In this regard, the algorithm is not sophisticated due to the tendency of an arbitrary SPT. Also, it guarantees only the minimum hop count for each path, and as such cannot provide an optimal compromise between the delay and the number of transmissions. In this paper, we propose an ILP formulation and algorithm that provides tie-break schemes and a multicast routing tree.

3. Problem Formulation

3.1 ILP Formulation

We formulate the bandwidth efficient multicast routing problem with integer linear programming. The model employs flow variables for source-destination pairs that can be easily applied to many routing problems with various objective functions and constraints as in [17]. In our model we are interested in bandwidth efficient multicast routing with guaranteed delay in the route. The general multi-commodity flow variables are introduced as follows.

$x_{ij}^d =$ Flow from source to destination d on link

$$z_{ij} = \begin{cases} 1 & \text{if } link\ (i,\ j) \text{ transmits data,} \\ 0 & \text{otherwise.} \end{cases}$$

The variables and notations specialized for our model are denoted, including the level of nodes. The binary variable indicates whether a node transmits or not. Additional hop count is also defined for the delay bound.

$$z_i = \begin{cases} 1 & \text{if node } i \text{ transmits data,} \\ 0 & \text{otherwise.} \end{cases}$$

L(i) = the level of node i assigned by the breadth first search (BFS).

α = the allowed number of additional hop count.

We then formulate ILP in a network graph $G(V,\ E)$ with source node $s$and destination set D as follows.

[ILP multicast formulation]

Objective function : *minimize* $\sum_{i \in v} z_i$ (1)

Subject to :

Flow conservation property

$$\sum_{j:(i,j) \in E} x_{ij}^d - \sum_{j:(i,j) \in E} x_{ji}^d = \begin{cases} 1 & \text{ifnode } i \text{ is source } s \\ -1 & \text{if node } i \text{ is d} \in D \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Link selection

$$x_{ij}^d \le z_{ij} \ \forall d \in D,\ \ \forall (i,\ j) \in E \quad (3)$$

Node selection

$$z_{ij} \le z_i \quad \forall (i,\ j) \in E \quad (4)$$

Hop count bound

$$\sum_{(i,j) \in E} x_{ij}^d \le L(d) + \alpha, \ \ \forall d \in D \quad (5)$$

Binary variable constraint

$$z_{ij},\ z_i,\ x_{ij}^d \in \{0,\ 1\}, \ \ \forall (i,\ j) \in E \quad (6)$$

The BFS process which initially sets L(i) of each nodes is as follows.

[BFS Level Assignment]

1. Assign level 0 to the source node; Denote a node set C and initially C contains the source node only; Set i = 0.
2. Repeat

   Assign level i+1 to level-unassigned child of node n, for $\forall n \in C$

   C consists of newly assigned nodes.

   i = i+1.

   Until C is empty

Objective function (1) minimizes the total number of transmission nodes. The objective is to minimize the number of nodes in the multicast routing tree except leaf nodes in graph $G(V, E)$. Constraint (2) represents flow conservation in the multi-commodity flow model. Constraint (3) converts flow variables to a link cost. Links without any flow are not selected by the constraint. Constraint (4) selects the transmitting nodes. When a link from a node is selected to support multicast service, the node is also selected, because it should send data as a transmitting node. The main idea underpinning our approach is constraint (5). It bounds the total length of the path from the source to each destination. The bound can be the shortest path length of the source-destination pair with additional hops. Longer path can be allowed with the value of α. When we make α large enough, the optimal tree will be MNT. By this constraint, we can find the minimum node multicast tree with a guaranteed path length.

3.2 Preprocessing

When the problem in the graph is NP-Complete or NP-Hard, preprocessing is usually used before optimization of LP or IP formulations. Preprocessing reduces the nodes and links that cannot be used in the optimal trees. The eliminated nodes and links thus have a clear reason to be deleted. The reduced topology provided by performing preprocessing allows a solution to be obtained in short time due to the low complexity of the graph. This useful technique is adapted to the minimum Steiner tree problem in [6]. There are several major preprocessing schemes, but they are not available in our problem due to the variation of the tree characteristic decided by α value. The hop count bound in this paper has a powerful role in preprocessing with the level of nodes and intuition from [6]. We discuss this later, and this preprocessing is divided into three cases : $\alpha=0$, $\alpha=1$, $\alpha \geq 2$. The reason why the process is changed by the allowed additional hop count value is shown in the following propositions.

Proposition 1 : In graph $G(V, E)$, let all nodes be assigned levels by the breadth first search (BFS) from the highest level 0 to the lowest level M. If a tree T of G contains a link from a lower level node to a higher level one, in other words, an *uplink*, then there is a destination node v such that the length of s-v path in T is at least $L(v)+2$.

Proof : It is well known fact that $L(v)$ is the length of the shortest s-v path in G. Suppose that a tree T has a link (u, d) such that $L(u) \geq L(d)+1$ and the link is included in the path from the source to v. The shortest path length from the source to node u is $L(u)$, and the link (u, d) adds one hop count to the path from the source to node d. Finally, the path to node d from the source is obtained by adding the path from the source to node u and link (u, d). Therefore, $L(u)+1$ is the path length, which is at least $L(d)+2$. Obviously, the shortest path from node d to node v has at least $L(v)-L(d)$ hop counts. If it is not true, $L(v)$ should be less than the current value. Finally, s-v path in T has length of at least $L(v)+2$.

Proposition 2 : In graph $G(V, E)$, let all nodes be assigned levels by the BFS method. If a tree T of G contains a link between two same level nodes, in other words, a *sibling link*, then there is a destination node v such that the length of s-v path in T is at least $L(v)+1$.

Proof : Exactly same as the proof of Theorem 1. When link (u, d) is connected, the length of the path from the source to d is L(u)+1 and L(u) = L(d). Hence the path has at least one or more edges than the shortest path.

According to the above propositions, when $\alpha=0$, in other words all paths from source to destinations are guaranteed to bethe shortest path, we can delete all *uplinks* and *sibling links*. One of the basic preprocessing steps in the Steiner tree problem is the deletion of leaf nodes that are not destinations [17]. If we select those nodes or links to the nodes, unnecessary costs are added to the resulting tree. Combining the preprocessing scheme and our theorems provides a highly reduced network topology. First, we delete all *uplinks* and *sibling links*, and then we have many leaf nodes. Each leaf node is easily deleted if it is not a destination, and links connected to the node also can be deleted. This procedure generates new leaf nodes and is iterated from the lowest level which is the set of nodes farthest from the source. The complete procedure is written below.

[Preprocessing, $\alpha=0$]

1. Delete all links between nodes in the same level.
2. Delete all links from lower level nodes to high level nodes.
3. Find the lowest level among the destinations and denote it as level M.
4. Delete all nodes i and links connected to the nodes, when L(i)> M is satisfied.
5. Delete all nodes $i$and links connected to the nodes, when L(i) = M and the condition that i is not in D is satisfied.
6. Proceed until M is 1

   Find the node set N subset $V-D$ whose level is M-1.

   Delete all nodes i in N such that have no child.

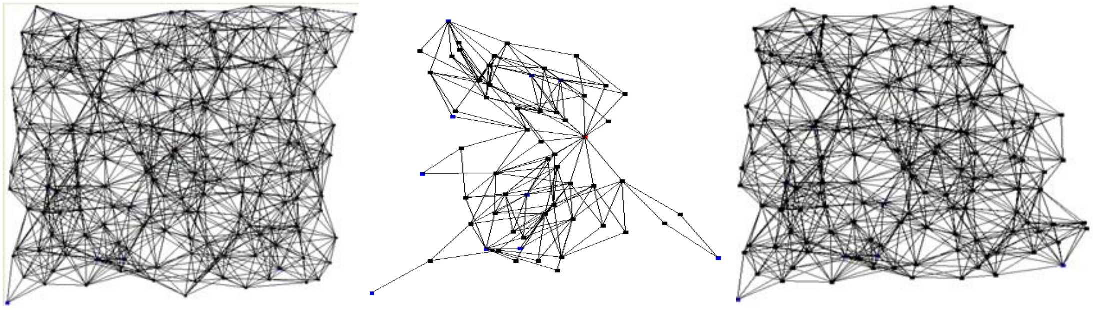
   M = M-1.

When $\alpha=1$, *sibling links* are available due to Theorem 2, and hence we need to modify step 1 and5 in the preprocessing procedure, $\alpha=0$. We cannot delete nodes in the range of destinations because one more path length is acceptable in this case. If a path arrives in the shortest hop count at a node that is a sibling of destination d, then it is possible to reach d with the shortest path length+1 hop count. Consequently, step 1 is deleted, and to modify step 5, we redefine the destination set D as the original destinations and their sibling nodes. The modified preprocessing when $\alpha=1$ is as follows.

[Preprocessing, $\alpha=1$]

0. Denote the destination set D; Li is the level of node $i$and each level of node is defined by BFS.
1. D = D ∪ sibling nodes of d, $\forall d \in D$
2. Do the same as $\alpha=0$ case from step 2 to step 6

If $\alpha \geq 2$, both *uplinks* and *sibling links* are available. Therefore, we should delete step 1, 2, and 4 and modify step 5 in the Preprocessing $\alpha=0$. Moreover, we cannot delete the nodes that are in α hop range of destinations in step 5, similar to extension of the $\alpha=1$ case. We conclude that this modification of pre-processing will hardly reduce the topology size due to the doubled number of links. We assume that the effi-



(a) (b) (c)

[Figure 1] The Shape of the Nodes and Connections of Original Graph (a), $\alpha=0$ Preprocessed Graph (b), $\alpha=1$ Preprocessed Graph (c)

ciency of preprocessing will be very low, and do not suggest preprocessing in the case of $\alpha \geq 2$.

The following figures are the results of preprocessing in a random graph. The figures represent all nodes and links in graph G, which has 200 nodes and 10 destinations. The source node is located in the center of the topology. [Figure 1](a) is the original graph and the number of links is approximately 1,500. [Figure 1](b) and [Figure 1](c) are the case of $\alpha=0$ and $\alpha=1$ respectively. The number of links is approximately 100 and 800 when $\alpha=0$ and $\alpha=1$ respectively. We can see that when $\alpha=1$, preprocessing does not reduce the topology well due to the numerous *sibling links*. However, the remarkable reduction when$\alpha=0$ allows us to always obtain the optimal solution even in a high density topology. The $\alpha=0$ case is easily optimized by ILP formulation and preprocessing and we propose an algorithm to solve the case of $\alpha \geq 1$.

4. Proposed Algorithm

Considering the difficulty of the problem when $\alpha \geq 1$, we provide a heuristic algorithm that guarantees the path length and minimizes the nodes in the multicast tree by adapting the α value to the path length and covering many nodes once. Also, we try to find a routing tree that has lower average path length with the same number of transmissions. We first propose some basic ideas to make this kind of tree.

First, we select the node that covers as many destinations as possible. If one node can cover many destinations or nodes in a pathway to other destinations and we select the node in the tree, many destinations can be connected at the cost of only one transmission. This directly creates a minimum node multicast tree. Therefore, we try to find the node providing the most coverage and select that node in our algorithm.

Second, we take a branching point of the tree very far from the source node. This can be easily seen in [Figure 2](a), where the left tree uses 5 transmissions and has an average path length of 3; on the other hand, the right one uses 4 transmissions and has an average path length of 4. If a tree branches earlier, each branch has to proceed to its destination individually and this causes additional nodes to transmit data to the same destinations. For this reason, the proposed algorithm goes from the lowest level of the graph to the source node and tries to make a branch point in an earlier step of the algorithm.

Third, we prohibit uplinks in the multicast tree. The previous two ideas are related to a minimum node tree, but the last is for reducing the average path length. As we show in Section 3, uplinks make a high (two or more) cost tree in terms of average path length. We can convert one *up-link* to two *sibling links* without loss of the transmission number, and this will cover the same nodes with a lower average path length. An example is given in [Figure 2](b), where both trees use the same number of transmissions but the right tree has shorter average path length.

We consider these factors in our algorithm as well as the newly adapted α value to loosen the path length limitation. The current additional hop count is assigned to each node and we can make a α-longer path length tree by checking and changing the value of a node. Since our algorithm runs based on the BFS level assigning and adapting α to make trees, initially we assign the level by the BFS and the current additional hop count as zero to each node in the graph.

[α-looseness Algorithm]

α : limit of the number of additional hop count to the shortest path length

An : current additional hop count value of node n

D : set of all destinations in graph

D(c) : set of all destinations with level c.

Up(n) : set of higher level nodes than n still not connected in the transmission range of node n

$\mathrm{Dn}_\alpha(n)$: set of same level nodes m with $\mathrm{A}_m < \alpha$ or lower level nodes m with $\mathrm{A}_m \leq \alpha$ in the transmission range of node n including node n

0. Li is the level of node $i$and each level of node is defined by BFS.
1. Find the lowest level among the destinations, and we denote the level M.
2. Set Ai = 0 for all nodes $i$in graph.
3. For c : from M to 1

   While $\mathrm{D}(c) \neq \varnothing$

   Select node i, $\mathrm{argmin}_{i \in \mathrm{D}(c)} |\mathrm{Up}(i)|$

   If a tie occurs, select an arbitrary node.

   Select node j, $\mathrm{argmin}_{j \in \mathrm{Up}(i)} |\mathrm{Dn}_\alpha(j) \cap \mathrm{D}|$

   If a tie occurs, select node j, $\mathrm{argmax}_j \mathrm{L}_j$

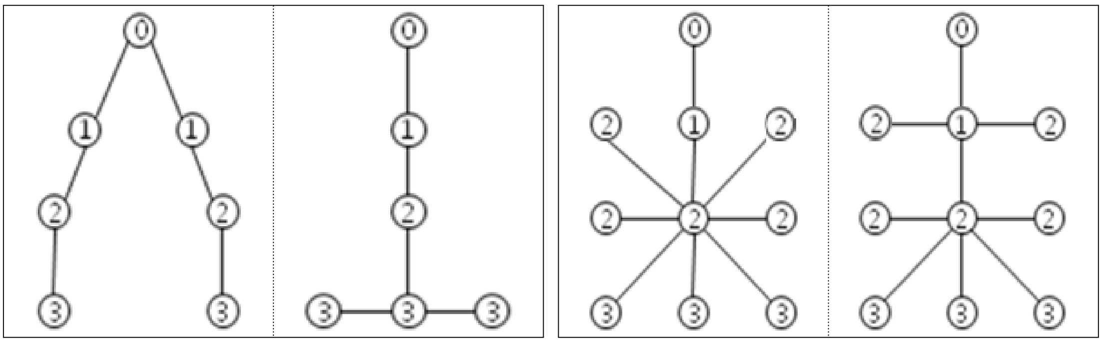
   If a tie occurs again, select node j, $\mathrm{argmax}_j |\mathrm{Up}(j)|$

   Connect link (i, k), $\forall k \in \mathrm{Dn}_\alpha(j) \cap \mathrm{D}$.

   Set $\mathrm{A}_j = \max_{k \in \mathrm{Dn}_\alpha(j) \cap \mathrm{D}} (\mathrm{A}_k + \mathrm{L}_j - \mathrm{L}_k + 1)$

   $\mathrm{D} = \mathrm{D} \setminus \{k\},\ k \in \mathrm{Dn}_\alpha(j) \cap \mathrm{D}$

   $\mathrm{D} = \mathrm{D} \cup \{j\}$

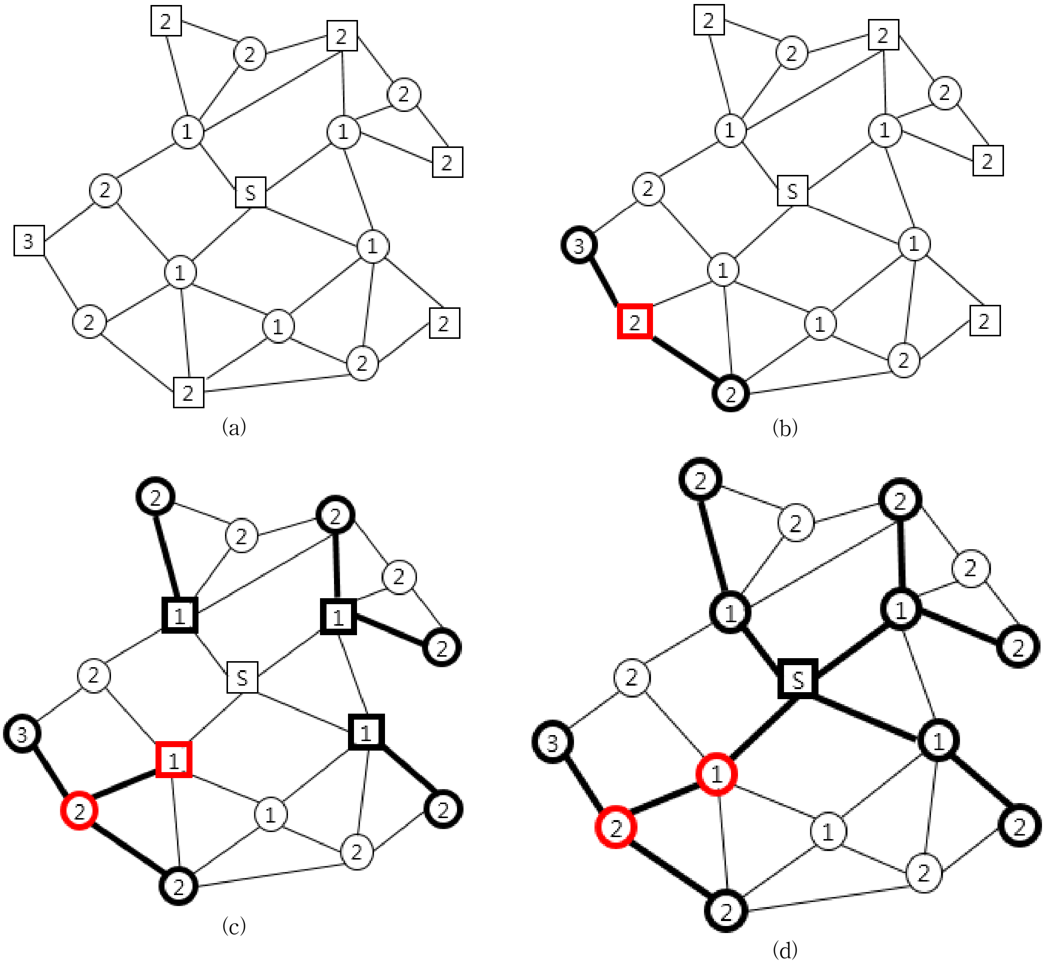


(a) (b)

[Figure 2] An Example of Branching (a) and an Example of Path Length (b)

The procedure of the proposed algorithm in the case of $\alpha=1$ is briefly described in [Figure 3]. S node represents the source node and the number of each node is the level assigned by BFS. The original graph is described in [Figure 3](a). Rectangle nodes represents the destinations and the proposed algorithm starts from the maximum level destination whose level is 3. [Figure 3](b) describes the procedure that selects a parent node in the upper node set of the level 3 destination. Tie breaking occurs and node j which covers another destination is selected. Node j is a new destination and since it connects a destination via sibling link, we set $A_j=1$ and highlight the node with red color. After this procedure arbitrary upper nodes are selected in level 2 and level 1 as we can see in [Figure 3](c). The final multicast tree constructed by the proposed algorithm is described in [Figure 3](d).

We give two criteria to break ties. One is the least upper node selection criterion and the other is the most upper node selection criterion. The

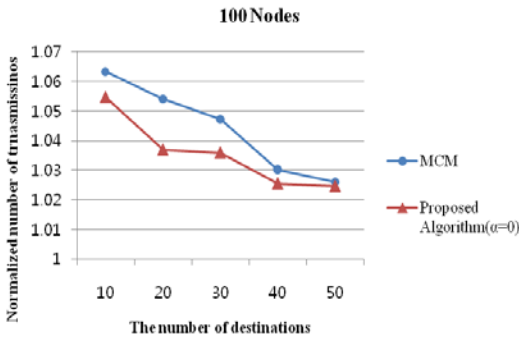


[Figure 3] Original Graph (a), After Level 3 Destination (b), Group of Level 2 Destination is Assigned (c), and Final Tree (d) former selects a destination node that has the smallest upper nodes in its transmission range, because the feature that a node has smaller upper nodes in its range means there is less chance it will be selected due to the existence of fewer parent candidates when other nodes have greater priority. The latter selects a parent node that has a high level and the most upper nodes in its range. The reason why we choose a high level is that if a high level node is selected as the parent node of some destinations, its average path length will be shorter than that of a low level node because the level difference of nodes is critical to the path length.

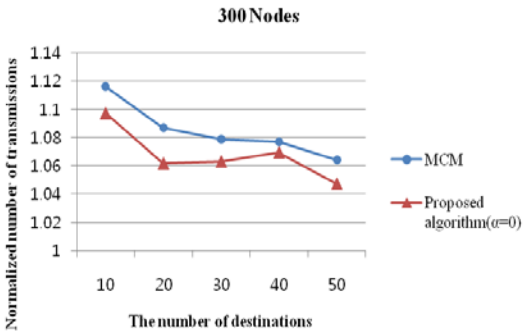
5. Simulation Results

In our simulation, we assume that all nodes are static and each node has the same fixed power to communicate with other nodes. This assumption is practical in wireless sensor networks and wireless mesh networks. Due to the fixed power assumption, the transmission range of each node is 90m, as generally used in wireless mesh networks [2]. The topology size is 500m×500m and we use a grid model. For a high confidence level, we simulate 100 times for all cases and evaluate the average values.

In [Figure 4], we compare MCM and proposed algorithm with $\alpha=0$ that both guarantee the shortest hop count to each destination in the 100 nodes topology. The normalized number of transmissions is evaluated with the optimal one. As we can see in the figure, the proposed algorithm provides slightly better solution with respect to the number of transmissions. This is due to the better tie breaking conditions which choosea parent node to cover more destinations among the candidate nodes. We can alsosee that both algorithms get closer to the optimal solution as the ratio of destinations grows. This is explained by relatively small number of parent candidates of the destinations when the ratio of destinations is high.



[Figure 4] Number of Transmissions with 10 to 50 Destinations in 100 Nodes

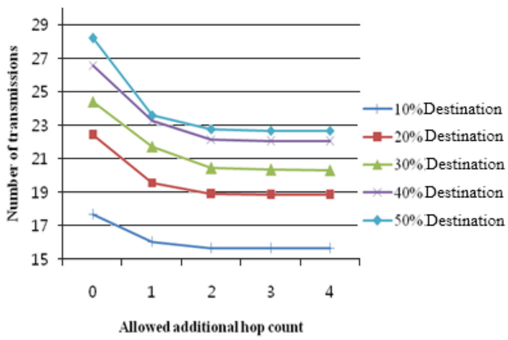


[Figure 5] Number of Transmissions with 10 to 50 Destinations in 300 Nodes

In [Figure 5], we simulate a topology with 300 nodes and we make two important observations in high density networks. First, the gap between the two algorithms is slightly bigger than that of 100 nodes. It will grow more and more in extremely high density networks due to arbitrary selection of the MCM algorithm. Second, the results of the two algorithms are far from the optimal solution because the high density network has a greater number of candidate paths to destinations and heuristics cannot find an efficient multicast routing tree properly.

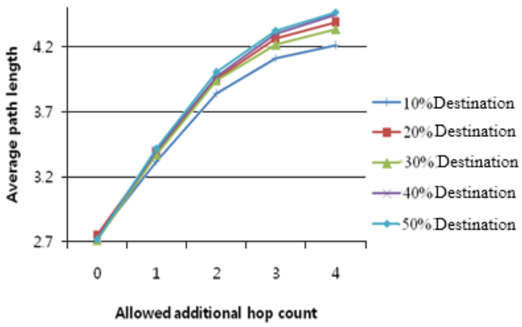
Although the proposed algorithm is not strong when $\alpha=0$, we show that it still outperforms the existing algorithm. We next show our simulation results when $\alpha \geq 1$, which is the strength of our algorithm.

In [Figure 6], the results of our algorithms for $\alpha=0, 1, 2, 3, 4$ are shown for the 300 nodes topology. The proportion of destinations varies from 10% to 50%. When we allow additional hop counts to each source-destination pair, the number of transmissions converges to certain values in each case. We also observe that the decrease of number of transmissions from $\alpha=0$ to $\alpha=1$ is remarkably high when the proportion of destinations is more than 20%. This phenomenon can be explained by the increase of gathering tendency which appears when more destinations are connected to one branch of a tree.



[Figure 6] Number of Transmissions with $\alpha=0, 1, 2, 3, 4$

[Figure 7] shows that the average path lengths for the cases of $\alpha=0, 1, 2, 3, 4$ are very similar to each other due to identical topology size. Note that an increase of the hop count by one does not lead to increase the average path length exactly by one because we try to reduce the average path length by applying our algorithm.



[Figure 7] Average Path Length with $\alpha=0, 1, 2, 3, 4$

6. Conclusion and Discussion

We have pointed out the problems of SPT and MNT, respectively, in constructing a multicast tree in wireless multi-hop networks. Each tree considers only one characteristic of wireless networks. We therefore have suggested a delay guaranteed node minimization problem and solved it by an appropriate ILP formulation. Due to NP-Completeness of the problem we have proposed an effective preprocessing procedure. This procedure efficiently reduces network complexity for $\alpha=0$. We could find optimal multicast routing trees in harsh environments such as extremely high density networks within limited time. For $\alpha \geq 1$ cases, our preprocessing is not useful because of *uplinks* and *sibling links*.

For $\alpha \geq 1$, we could not suggest adequate solutions, and thus we have proposed a new algorithm to make a multicast tree with a limited number of hop counts and minimized number of transmissions. We employed α allowance for additional path length to the multicast tree. By utilizing α, the algorithm successfully creates a hop count boundedtree.

In the simulation results, we have shown that our algorithm performs better for $\alpha=0$ than the existing multicast tree algorithm MCM. Most of all, it provides a delay guaranteed multicast routing tree for any α value and topology conditions. The main strength of proposed algorithm is flexibility to various wireless networks. Our algorithm can make an adequate tree for the given network topology with various node density, destination density, and α values. Generally, the proposed algorithm performs better in high density networks with high portion of destinations. The computational result shows that the number of transmissions is sharply reduced when $\alpha=1$and converges when $\alpha \geq 2$. As a result, we conclude that $\alpha=1$ is a proper compromise between the number of transmissions and the average path length in the proposed algorithm. We also observe that the average path length is increased only by 0.7 when we allow one more hop to each destination, which reflects the efficiency of the proposed multicast routing procedure.

References

[1] Akyildiz, I.F., X. Wang, and W. Wang, "Wireless mesh networks : a survey," *Computer Networks and ISDN Systems*, Vol.47, No.4(2005), pp.445-487.

[2] Alicherry, M., R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks," *IEEE Journal on Selected Areas in Communications*, Vol.24, No. 11(2006), pp.1960-1971.

[3] Biradar, R.C. and S.S. Manvi, "Ring Mesh Based Multicast Routing Scheme in MANET Using Bandwidth Delay Product," *Wireless Personal Communications*, Vol.66, No.1 (2012), pp.117-146.

[4] Chuah, S.-P., Z. Chen, and Y.-P. Tan, "Energy-Efficient Resource Allocation and Scheduling for Multicast of Scalable Video Over Wireless Networks," *IEEE Trans. Multimedia*, Vol.14, No.4(2012), pp.1324-1336.

[5] Even, S., "Graph Algorithms," *Computer Science Press*, (1979), pp.204-209.

[6] Koch, T. and A. Martin, "Solving Steiner tree problems in graphs to optimality," *Networks*, Vol.32, No.3(1998), pp.207-232.

[7] Kun, Z., Q. Yong, and Z. Hong, "Dynamic multicast routing algorithm for delay and delay variation-bounded Steiner tree problem," *Knowledge-Based Systems*, Vol.19, No.7(2006), pp.554-564.

[8] Lee, S.I., M. Gerla, and C.C. Chiang, "On-Demand Multicast Routing Protocol," *Proceedings of IEEE WCNC*, (1999), pp.1298-1302.

[9] Nguyen, U.T. and J. Xu, "Multicast Routing in Wireless Mesh Networks : Minimum Cost Trees or Shortest Path Trees?," *IEEE Communications Magazine*, Vol.45, No.11(2007), pp.72-77.

[10] Rouskas, G.N. and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," *IEEE Journal on Selected Areas in Communications*, Vol.15, No.3(1997), pp.346-356.

[11] Royer, E. and C.E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing Protocol," *Proc. of the 5th ACM/IEEE Annual Conf. on Mobile Computing and Networking*, (1999), pp.207-218.

[12] Ruiz, P.M. and A.F. Gomez-Skarmeta, "Approximating Optimal Multicast Trees in

Wireless Multihop Networks," *10th IEEE Symposium Comp. and Communication*, (2005), pp.686-691.

[13] Sanchez, J.A., P.M. Ruiz, J. Liu, and I. Stojmenovic, "Bandwidth-Efficient Geographic Multicast Routing Protocol for Wireless Sensor Networks," *IEEE Sensors Journal*, Vol.7, No.5(2007), pp.627-636.

[14] Sharangi, S., R. Krishnamurti, and M. Hefeeda, "Energy-Efficient Multicating of Scalable Video Streams Over WiMAX Networks," *IEEE Trans. Multimedia*, Vol.13, No.1(2011), pp.102-115.

[15] Wan, P.J., G. Călinescu, and C.W. Yi, "Minimum-power multicast routing in static ad hoc wireless networks," *IEEE/ACM Transactions on Networking(TON)*, Vol.12, No.3(2004), pp.507-514.

[16] Wieselthier, J.E., G.D. Nguyen, et al., "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *IEEE INFOCOM*, (2000), pp.585-594.

[17] Yuan, D., J. Bauer, and D. Haugland, "Minimum-energy broadcast and multicast in wireless networks : An integer programming approach and improved heuristic algorithms," *Ad Hoc Networks*, Vol.6, No.5 (2008), pp.696-717.

[18] Zeng, G., B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," *ICNP, Network Protocols, IEEE International Conference on*, (2007), pp.1-10.