Heuristic Algorithms for Constructing Interference-Free and Delay-Constrained Multicast Trees for Wireless Mesh Networks

Wen-Lin Yang[1], Chi-Chou Kao[1] and Cheng-Huang Tung[2]
[1] Dept. of Computer Science and Information Engineering, National University of Tainan,
Tainan, Taiwan
[e-mail: {wlyang, cckao}@mail.nutn.edu.tw]
[2] Dept. Computer Science and Information Engineering, National Pingtung Institute of Commerce
Pingtung, Taiwan
[e-mail: chdong@npic.edu.tw]
*Corresponding author: Wen-Lin Yang

Received November 16, 2010; revised December 21, 2010; accepted January 12, 2011;
published February 28, 2011

Abstract

In this paper, we study a problem that is concerning how to construct a delay-constrained multicast tree on a wireless mesh network (WMN) such that the number of serviced clients is maximized. In order to support high-quality and concurrent interference-free transmission streams, multiple radios are implemented in each mesh node in the WMNs. Instead of only orthogonal channels used for the multicast in the previous works, both orthogonal and partially overlapping channels are considered in this study. As a result, the number of links successfully allocated channels can be expected to be much larger than that of the approaches in which only orthogonal channels are considered. The number of serviced subscribers is then increased dramatically. Hence, the goal of this study is to find interference-free and delay-constrained multicast trees that can lead to the maximal number of serviced subscribers. This problem is referred as the *MRDCM* problem. Two heuristics, load-based greedy algorithm and load-based MCM algorithm, are developed for constructing multicast trees. Furthermore, two load-based channel assignment procedures are provided to allocate interference-free channels to the multicast trees. A set of experiments is designed to do performance, delay and efficiency comparisons for the multicast trees generated by all the approximation algorithms proposed in this study.

Keywords: Wireless mesh networks, delay-constrained multicast, channel assignment, multiple radios, orothgonal channels.

A preliminary version of this paper appeared in ICIMT 2010, December 28-30, HK. This research was supported by a research grant from the National Science Council, Taiwan, R.O.C. [NSC-98-2221-E-024-004-MY2].

DOI: 10.3837/tiis.2011.02.002

1. Introduction

Recently, the IEEE 802.11-based wireless mesh network (WMN) is expected to be a new promising tecnology to provide last-mile broadband services and ubiquitous connectivity to homes and businesses in rural, suburban, or metropolitan areas. The backbone of a WMN consists of a number of static mesh routers (MRs) equipped with one or more radio interfaces, where one of them acts as the gateway node connected with the wired network. The other MRs access Internet through the gateway node by multiple-hop communications. The end-users serviced by MRs are called mesh clients. They connect to WMNs through access points built on MRs [1].

It is well-known that the performances of multi-hop wireless networks are greatly affected by the interference between links with overlapping channels in vicinity areas. For binary interference model adopted in reference [2], two interfering links cannot transmit and receive packets at the same time if they are allocated with the same channel. In fact, interference can be classified into two categories, co-channel and adjacent-channel [3]. For links within interference range, the co-channel interference occurs from transmissions on the same channel, whereas the adjacent-channel interference occurs from transmissions on adjacent and overlapping channels. As a result, the network aggregate capacity in WMNs may be greatly reduced. Generally, there are two approaches for solving the problem. One approach is to consider how to fully utilize multiple channels for WMNs where each MR is equipped with one radio interface. The single interface has to switch among all the channels frequently in order to communicate with the neighbors [5][6]. This operation may cause significant delay for IEEE 802.11 hardware. As a result, it is an inefficient approach for WMNs with single-interface. A study on modification of the operating system kernel is then represented for supporting frequent channel switching [7].

The second approach is that each MR is equipped with multiple radio interfaces and each interface can tune to one channel at a time. Multiple channels can be allocated to multiple interfaces at the same time. As a result, multiple concurrent transmissions among MRs are allowed. However, interference from among neighboring links may still exist if they are allocated with the same or partially overlapping channels. Therefore, the key issue of this approach is how to bind each interface to a radio channel in a way that the interference is minimized and the capacity is maximized. This problem is recognized as the interference-aware channel assignment (CA) problem for the multi-channel multi-radio (*MCRM*) WMNs. For ease of the difficulty for solving the interference-aware CA problem, only non-overlapping (orthogonal) channels are considered for most of the CA algorithms proposed in literature [9][10][11].

For links within the interference range, orthogonal channels can be used to achieve full-duplex transmissions without causing any interference. Unfortunately, the number of orthogonal channels is few. For example, only 3 out of 11 channels in IEEE 802.11b/g standard are orthogonal channels [3]. With this constraint, it would be difficult to support multiple concurrent transmissions without incurring any interference among links, even for the moderate-size networks. Therefore, by taking both the network connectivity and interference into account, the goal of most CA algorithms published is to minimize the overall network interference in order to increase network capacity [13][14][15][16][17][18]. However, it may not be enough for networks with minimal interference to support some multimedia real-time applications like high-resolution net-TV where stable high-bandwidth transmission channel is required to provide high-quality video/audio streams. The interference-free networks would be a better choice for this kind of applications.

In this paper, we consider a scenario shown in **Fig. 1** that a multimedia stream delivering system is deployed on a WMN, where multimedia servers are assumed to be attached to the gateway node. In addition, each MR is equipped with a number of network interfaces and omni-directional antennas. The mesh clients may subscribe to a streaming program from the video server and want to receive the subscribed stream at the same time. For example, a large amount of users may subscribe simultaneously to watch a TV show in live on the Internet. For multimedia applications involved in real-time communications described as above, the multicast technology would be more appropriate than unicast for efficiently delivering continuous streams to a large amount of mesh clients [4]. Since each MR is equipped with multiple radios, our problem is to investigate how to implement a multicast tree on *MCRM* WMNs so that the following three goals can be achieved:

(1) The number of mesh clients who can receive the subscribed streams is maximized.
(2) The multicast tree delay is bounded by a given value.
(3) The transmissions along all the tree links are interference-free.

Since the more subscribers can be serviced, the more profits we can make. This problem is then referred as the maximum-revenue and delay-constrained multicast (*MRDCM*) problem in this study. In our previous work, only orthogonal channels are considered for the multicast trees with unbounded delay [11]. The experimental results in [11] show that a large amount of links cannot allocate any interference-free channel since they are only 3 orthogonal channels available in IEEE 802.11b/g technology. As a result, the number of destinations that are not covered by the resulting multicast tree tends to be large. That is, the number of serviced subscribers is small. This performance is much worse than that found by the algorithms presented in this paper. This performance comparison will be shown in section 5.1.

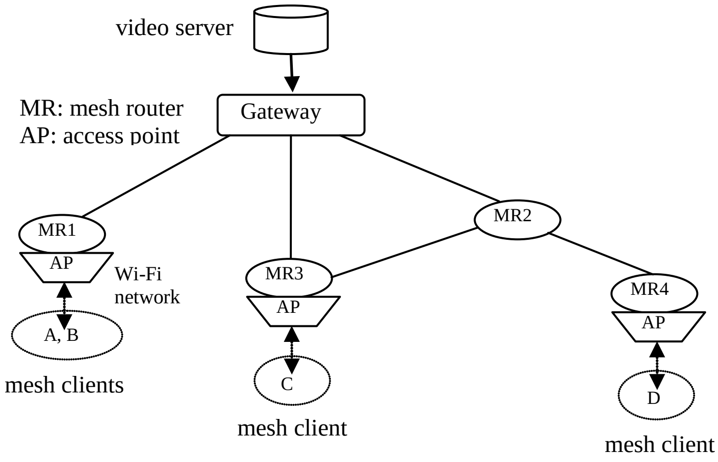

Fig. 1. A video-stream delivering system is deployed on a wireless mesh network.

Therefore, in order to increase the number of links being allocated channels, both orthogonal and partially overlapping channels are considered for the CA procedure in this paper for the *MRDCM* problem. This relaxation is based on the study on interference factor (IF) presented in references [3][15] where it is shown that the interference range is determined by both the transmission rate and the distance between two MRs. Based on IF data shown in

[15], we develop a CA method which can assign orthogonal and partially overlapping channels to the tree links without causing any conflict.

The *MRDCM* is a NP-hard problem since one of its sub-problems is the CA problem that has been proven to be NP-hard by reducing the *Multiple Subset Sum problem* to it [16]. Hence, only heuristic algorithms are investigated here. The heuristic algorithms consist of two parts: a procedure for constructing a delay-constrained multicast tree and a procedure for allocating channels. In this paper, we propose two approximation algorithms: the LMCM method and the greedy method, for constructing the multicast trees for the *MRDCM* problem. Then, based on DFS and BFS paradigms, two load-guided CA procedures are developed to allocate interference-free channels to the multicast trees.

The rest of the paper is organized as follows. Section 2 reviews past work related to multicast routing in *MCRM* WMNs. Section 3 gives the formulation of our *MRDCM* problem proposed in this study. Section 4 describes two heuristics for building muticast trees and two CA procedures for allocating channels. Section 5 presents the experimental results. We conclude our work in Section 6.

2. Related Work

The multicast routing can be implemented on networks with tree-type or grid-type topologies. Most of the proposed multicast routing algorithms are developed based on trees since they have lower implementation cost. For wired networks, multicast trees constructed may be optimized based on different objective functions such as number of hops, delay of the longest path, aggregate cost of tree edges, etc. For the multicast trees determined by the Kou's approach [8], for example, they are built based on minimizing the total cost of links in the resulting tree. As for the Shortest-Path method [14], the multicast tree is formed by merging all the shortest paths between source node and destinations. These unicast paths with minimal cost are determined by the Dijkstra's algorithm.

However, prior studies [3][12] have shown that the multicast trees designed for wired networks may be not suitable for wireless networks. Hence, multicast trees constructed based on minimizing the number of data transmissions [13] or the number of relay nodes [12] are proposed. In fact, these two optimization criteria are similar. In reference [13], a greedy-based heuristic algorithm is proposed to build multicast trees with the minimal number of forwarding nodes for a single-channel WMN. These trees are referred as MNT trees. The MNT tree is built without considering the minimization of maximum path delay of the tree.

The CA problem for minimizing the interference caused by adjacent-channel and co-channel is the key problem for *MCRM* WMNs. Most CA algorithms pusblished in the literarture are designed for unicast. There are only a few CA algorithms proposed for multicast. In reference [17], a CA algorithm called BFVC based on greedy vertex coloring strategy is proposed for a given multicast tree. One problem with BFVC algorithm does not take the wireless broadcast advantage (WBA) into account. In addition, since only orthogonal frequency channels are used for channel allocation, a lot of bandwidth resource is wasted. Most algorithms designed for *MCRM* WMNs perform routing first, then followed by CA procedure [12][15]. In contrast, the “CA first, routing second” approach is considered in reference [18]. In this paper, their goal is to construct a multicast tree for a given *MCRM* network with allocated channel assignments so that the amount of network bandwidth comsumed by the routing tree is minimized. However, no study on channel assignment or transmission interference on links is reported.

In reference [12], an algorithm named MCM is used to construct multicast trees with minimum relay nodes and minimum hop counts for distances between source and destinations. Based on the concept of interference factor (IF), a CA procedure is also developed for allocating channels for *MCRM* WMNs. However, the MCM alorithm suffers from the hidden channel problem by considering interference from only one-hop neighbors, which is then solved by a CA algorithm named M4 presented in reference [15]. In M4 algorithm, the interference from one-hop and two-hop is taken into account when the optimization function is derived for minimizing the interference. All the orthogonal and partially overlapping channels are used in both MCM and M4 methods. In conclusion, the central goal of both the MCM and M4 algorithms is to deterime minimium interference multicast trees for *MCRM* WMNs. However, it is easy to have a situation that interference may occur from a neighbor that is more than two-hop away from the current node as long as $r \le IR$, where r is the distance between them and IR is the interference range.

In this paper, therefore, we are interested in constructing an interference-free multicast tree for our *MRDCM* problem using all the available channels based on the study of IF presented in references [3][15]. The main objective of this study is to maximize the number of clients who can successfully receive subscribed streams from the multimedia server embeded in the WMN. This approach is very different to the prior studies described above.

3. The Network Model and Problem Formulation

For a WMN shown in **Fig. 1**, it can be modeled by an undirected graph $G = (V, E)$ where V represents a set of nodes and E is the set of undirected edges. For an edge connecting two nodes u and v, we define a link $e(u,v)$ representing a flow from u to v. A non-negative real-value function $d(u,v)$ is associated with each link $e(u,v)$ and represents the delay that the packet experiences through passing the link e. It is assumed that $d(u,v) = d(v,u)$. We also assume that a non-negative integer req associated with each node u represents the number of subscribers required to be serviced at u. Let M denote the multicast group which is a set of destinations and can be defined as $M = \{u \mid u \in V, req(u) > 0\}$.

Suppose a video-stream delivering system is deployed on the gateway node s to provide continuous video/audio streams. For a given delay bound Δ, we are required to find a multicast tree $T = \{\overline{V}, \overline{E}\}$ rooted at s that can transmit continuous streams from s to all the nodes in M such that the following two constraints are satisfied:

(1) The delay on the path from s to u can not be greater than Δ for each $u \in M$.

(2) A number of concurrent wireless transmissions occurred among all the links in T must be interference-free.

To satisfy the above two constraints, a tree-construction procedure is needed to find the delay constrained multicast tree T, which is then processed by a CA procedure to allocate conflict-free channels to all the links of T. Each node of T is assumed to have at least two network interfaces installed, and all the channels including both the orthogonal and non-orthogonal are available for T. That is, our CA procedure must consider how to allocate multiple channels to multiple interfaces for each node in T without causing interference.

However, since the number of available channels is limited, it may be not possible to allocate interference-free channels to all the links in T, especially for the large-size multicast trees. As a result, there is a certain number of links in the multicast tree can not have any interference-free channels assigned. The links without channels allocated should be removed from T. A multicast tree assigned with interference-free channels is called *MRDCM* tree in this paper. Obviously, there are many ways to allocate channels to a multicast tree to produce *MRDCM* trees. A *MRDCM* tree is a subtree of the multicast tree T. In addition, a *MRDCM* tree that can span all the nodes in M should generate the maximum number of serviced users. Therefore, the goal of this study is to find a *MRDCM* tree which does not just satisfy two constraints described above but also maximizes the number of serviced users.

In references [3][15], the interference factor (IF) is defined to be the ratio of the interference range and the transmission range, and it is dependent on the transmission rate at the physical layer. For transmission rate of 11 Mbps in IEEE 802.11b/g standard, Table I shows the values of the interference factor versus channel separation. The channel separation of two channels c_1 and c_2 is defined as $|c_1 - c_2|$.

Based on **Table 1**, the minimal values of channel separation of two nodes are greatly affected by the distance between them. For example, if the distance between two nodes is $1.1R$, the value of channel separation must be at least 2 to avoid interference between two nodes. That is, if $c_1 = 2$ is assigned to one node, then c_2 can be 4,5,...,11, which is eligible for the other node. Based on the observation described above, one goal of this study is to allocate all the channels to the links to produce an interference-free multicast tree.

Table 1. Channel separation versus interference factor

channel separation	0	1	2	3	4	5
interference factor	2.0	1.2	0.7	0.5	0.2	0.0

In conclusion, the *MRDCM* problem proposed in this paper can be mathematically formulated as follows. Let $\Psi = \{(\Omega_{mrdcm}, T_{mrdcm})\}$, where Ω_{mrdcm} is a set of CA for the *MRDCM* tree T_{mrdcm}. The goal of this study is to determine an optimal pair of $(\Omega_{mrdcm}, T_{mrdcm})$ so that the following objective functions are satisfied:

$$gain = Maximize \sum_{u \in M \,\&\, u \in T_{mrdcm}} req(u) \tag{1}$$

$$\max_{u \in M \,\&\, u \in T_{mrdcm}} \left(\sum\nolimits_{e \in path(s,u)} d(e)\right) \le \Delta \tag{2}$$

$$\Omega_{mrdcm} \text{ is a set of interference-free CA.} \tag{3}$$

4. The Heuristic Algorithms

The heuristic algorithms for our *MRDCM* problem can be divided into two steps: the first step is to construct a desired multicast tree; then followed by a procedure that is to assign partially overlapping channels to the links of the multicast tree. In section 4.1 and 4.2, two approximation algorithms are proposed for constructing the best multicast trees based on the *load* value of each node. The *load* value will be defined for these two algorithms respectively. For the multicast trees found by these two algorithms, a CA procedure is needed for allocating interference-free channels to the tree links. Two CA procedures based on DFS and BFS paradigms are developed and given in section 4.3.

```
Input: Given a network $G=(V,E)$, a gateway node $s$, a delay bound $\Delta$ and a set of destinations $M$ where $M=\{u\,|\,u\in V,\ req(u)>0\}$.
1.  Starting from the gateway node, a BFS traversal procedure is used to assign each node a level. A tree mesh is then obtained by deleting all the links between any two nodes which are in the same level;
2.  Let $T=\{\bar{V},\bar{E}\}$ denote the resulting multicast tree, where $\bar{V}$ is the node set and $\bar{E}$ the link set; $\bar{V}=\phi$, $\bar{E}=\phi$;
3.  Let array $InTree[u]=1$ and $\bar{V}=\{u\}$, if $u\in M$ or $u=s$;
4.  For ($l=maxLevel$; $l\geq 1$; $l=l-1$) {
5.      $S_i=\{u_i\ at\ level\ \ l-1\}$;
6.      $S_j=\{u_j\ at\ level\ \ l\ \ and\ \ InTree[u_j]=1\}$ ;
7.      While ($S_j\neq\phi$) {
8.          Let $L_j=\{u_k\,|\,u_k\in S_j,\ u_k$ has the minimal number of parents$\}$ ;
9.          Find $t_f\in S_i$ such that $t_f$ with the maximum load where $(t_f,u_k)\in E$, $u_k\in L_j$;
10.         $InTree[t_f]=1$; $\bar{V}=\bar{V}\cup\{t_f\}$;
11.         $S_i=S_i-\{t_f\}$;  $S_j=S_j-\{u_j\,|\,(t_f,u_j)\in E\ and\ u_j\in S_j\}$
12.         $\bar{E}=\bar{E}\cup\{(t_f,u_j)\,|\,(t_f,u_j)\in E\ and\ u_j\in S_j\}$;
13. } }
14. Starting from $s$, compute the path delay of each node $u\in\bar{V}$;
15. Remove the subtree rooted at $u$ if path delay of $u$ is greater than $\Delta$;
16. Recursively remove the leaf node $u$ if $req(u)=0$;
Output: A delay-constrained multicast tree $T$.
```

Fig. 2. The LMCM algorithm

4.1 The Load-based MCM Algorithm

In reference [15], a MCM tree construction algorithm is proposed for building a multicast tree with multi-channel and multi-interface. The first step to construct the MCM tree is to generate a tree mesh using BFS traversal procedure, which can partition all nodes in the given network into different levels. With BFS traversal, the number of hops of each path from the root to a destination is expected to be minimized. The minimal number of relay nodes in the tree mesh is then determined to construct the embedded multicast tree.

Based on similar ideas, the MCM algorithm is revised in this subsection by taking into account the *load* value of each node during the phase that the relay nodes are determined. This revised MCM algorithm is named load-based MCM (LMCM) (see **Fig. 2**) where at line 10 the best relay node is selected for each loop executed at line 8. The selection principle is based on the *load* value of each node. During the formation of the multicast tree, the *load* value of a node u is defined to be the total subscribers of a sub-tree rooted at u. And, the node with the largest *load* value has the highest priority to be selected as the relay node at line 10 in **Fig. 2**. The selection criterion is very different from the criterion used in the original MCM algorithm [15].

The time complexity of LMCM algorithm is determined by the nested loop executed at line 4 and line 7. For the “while” loop in line 7, it takes $O(|V|^3)$ steps since the number of nodes in

S_j is bounded by $O(|V|)$ and it takes $O(|V|\log|V|)$ steps at line 8 to setup set L_j at the worst case. In total, the LMCM algorithm takes $O(|V|^2 \times \log|V| \times L)$ steps where L is the maximal number of levels used in the "for" loop at line 4.

4.2 The Load-based Greedy Algorithm

We propose a greedy approach for constructing the desired multicast tree T in this section. This greedy algorithm consists of the following two steps:

(1) The first step is required to build up a node-weighted network $\overline{G}$. It is an auxiliary graph constructed from the graph $G = (V, E)$ that represents the given WMN. This procedure is discussed in subsection 4.2.1.

(2) Based on $\overline{G}$, the multicast tree T is constructed by a heuristic algorithm named *load-based prim's* procedure. This procedure is described in subsection 4.2.2.

In total, the time complexity of the load-based greedy algorithm is bounded by $O(|V|^2 L)$, where $|V|$ is the number of nodes and L is the tree height. The analysis of running time is given in the following subsections.

4.2.1 The Procedure for Generating Node-weighted Networks

The node-weighted network $\overline{G} = (V, E)$ is built based on the procedure shown in **Fig. 3**. In this procedure, the first step is to determine each node a value called *level* based on the topological sorting in breath-first fashion.

```
Input: an undirected graph $G = (V, E)$ represents the given network, a gateway node $s$ and
$req(u)$ is the number of subscribers connected to each mesh router $u$.
Starting from $s$, a topological sorting procedure is carried out for the network $G$ to
    produce a directed acyclic graph;
Starting from $s$, each node $u \in V$ is assigned a value called $level$. The value of
    $level(s)$ is set to be 0;
Let $maxLevel = \max_{u \in V} level(u)$, and $load(u) = req(u)$ for each node $u \in V$;
For ($l = maxLevel$; $l \geq 0$; $l = l - 1$) {
    For each node $u$, if ($level(u) == 1$) {
        For each node $u$, if ($v$ is adjacent to $u$ and $level(v) < level(u)$) {
            $load(v) = load(v) + load(u)$;
} } }
Output: a node-weighted network $\overline{G} = (V, E)$.
```

Fig. 3. A node-weighted network generation procedure

The *level* value of a node is used to indicate the number of hops away from the root. For each node u with the largest *level* value, the *load* value is set to be the *req* value. A *load* value of a node u is then recursively defined to be the summation of the number of subscribers of nodes which are downstream nodes of u. The definition about the term "*load*" is slightly different from the one described in section 4.1. In this procedure, we define downstream nodes of u to be nodes which are recursively adjacent to u or neighbors of u, and have *level* values which are less than that of u. The time complexity of this procedure is bounded by the nested loop executed at line 4, 5, 6. It takes $O(|V|^2 \times L)$ steps at the worst case where L is the maximal number of levels used in "for loop" at line 4.

For the mesh network with *level* and *req* values shown in **Fig. 4**, the *load* values are computed based on the procedure described in **Fig. 3** and are given in **Fig. 5**. Consider the node g in **Fig. 5**, its *load* value is four which is the summation of *req* values of k, h and g. It is resulted from that $level(g) < level(k)$ and $level(g) < level(h)$ where k and h are neighbor nodes of g. Note that the *load* values of k and h are equal to their *req* values respectively, because they are assigned the largest *level* values based on topological sorting. As for the other neighbor node d, its *req* value does not have any effect on the $load(g)$ since $level(d)$ is not less than $level(g)$.

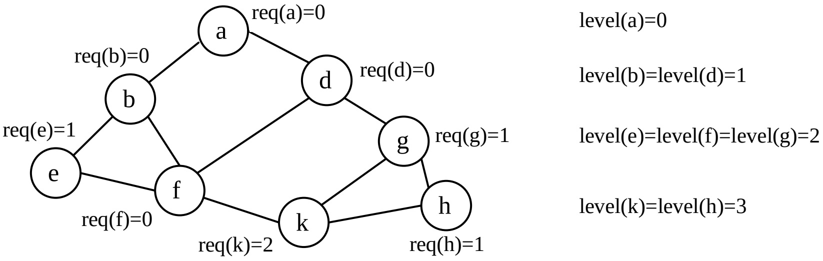

Fig. 4. A mesh network example.

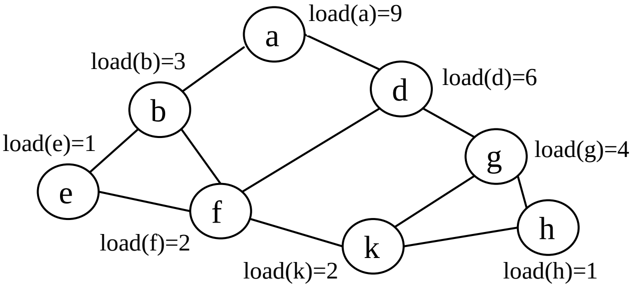

Fig. 5. A node-weighted mesh network.

4.2.2 The Load-based Prim's Procedure

The Prim's algorithm is a conventional algorithm for finding minimum spanning tree for a connected weighted undirected graph. The similar idea is used in this section for developing our *load-based prim's* procedure for constructing a multicast tree for the node-weighted graph $\overline{G}$. The *load-based prim's* procedure is shown in **Fig. 6**. The procedure begins with the Gateway node which is set as the root, and then adds one node into the tree at a time based on the *load* value of each uncovered node, where the *load* value of each node has been computed for $\overline{G}$ in the procedure described in **Fig. 3**.

The most important setp in **Fig. 6** is the step 3 for selecting the best uncovered node to be included into the current tree. As the before, the uncovered node with the largest *load* value is selected in the highest priority. The procedure stops when all the destinations are included in the tree. The time complexity of this procedure is the same as that of original prim's algorithm. It takes $O(|V|^2)$ steps at the worst cases.

```
Input: A node-weighted network $\overline{G} = (V, E)$, a gateway node $S$, a delay bound $\Delta$ and a set of destinations $M$ where $M = \{u \mid u \in V,\ req(u) > 0\}$.
1.  Let multicast tree $T = \{\overline{V}, \overline{E}\}$, where $\overline{V} = \{s\}$ and $\overline{E} = \phi$;
2.  While ($M \neq \phi$) {
3.      Find a $link(u,v)$ such that $load(v)$ is maximum where $u \in \overline{V}$ and $v \in V - \overline{V}$;
4.      If ($v \in M$) {$M = M - \{v\}$;}
5.      $\overline{V} = \overline{V} \cup \{v\}$;
6.      $\overline{E} = \overline{E} \cup \{link(u,v)\}$;
7.  }
8.  Starting from $S$, compute the path delay of each node $u \in \overline{V}$;
9.  Remove the subtree rooted at $u$ if path delay of $u$ is greater than $\Delta$;
10. Recursively remove the leaf node $u$ if $req(u) = 0$;
Output: A delay-constrained multicast tree T.
```

Fig. 6. A load-based prim's procedure.

4.3 The DFS-based and BFS-based Channel Assignment Procedures

4.3.1 Minimum Distance versus Channel Separation

Given two distinct links $e_1 = (u, v)$ and $e_2 = (x, y)$ in a tree supporting concurrent transmissions, the interference range is determined by the minimum distance δ between any two nodes from different links. Based on the δ value and the IF value defined in Table I shown in section 3, three situations for supporting interference-free channel assignments for e_1 and e_2 are considered as follows. Let η denote the separation for channels assigned to e_1 and e_2 and R is the transmission range.

(1) $\delta = 0$ and $u = x$, e_1 and e_2 are sibling links and they may share the same channel based on WBA, so $\eta \geq 0$.

(2) $\delta = 0$ and $(v = x \text{ or } u = y)$, e_1 and e_2 have one end-node in common, so $\eta \geq 5$.

(3) $\delta > 0$, several cases are considered in the **Table 2**.

Table 2. Minimum distance between two links versus channel separation.

Minimum distance between two links	Channel separation
$0 < \delta < 0.2R$	$\eta \geq 5$
$0.2R \leq \delta < 0.5R$	$\eta \geq 4$
$0.5R \leq \delta < 0.7R$	$\eta \geq 3$
$0.7R \leq \delta < 1.2R$	$\eta \geq 2$
$1.2R \leq \delta < 2R$	$\eta \geq 1$

$2R \leq \delta$	$\eta \geq 0$

4.3.2 The Load-based DFS Channel Assignment

Based on the DFS traversal and the discussion about channel separation presented in section 4.3.1, non-orthogonal and orthogonal channels can be allocated to the given multicast tree T in a way that no interference occurs between any two links in T. This interference-free CA procedure is presented in **Fig. 7**, where the *load* value of each node is computed first at line 1 and is used to guide the CA process.

```
Input: A multicast tree $T=\{\overline{V},\overline{E}\}$ and a set of channels $\Phi$.
1.  Recursively compute the $load(u)$ value for each node $u$, where the $load(u)$ is
    defined to be the total subscribers of a subtree rooted at $u$.
2.  Compute the interference matrix $IM$ for all the links in $T$; In $IM$, the minimum
    distance between any two links and the corresponding interference factor are
    determined;
3.  The channel assignment procedure is started at the root and recursively carried out in
    DFS fashion; It is shown as follows:
4.  Load_based_DFS($u$) {
5.      For node $u$, set $L=\{(u,v_1),\ldots,(u,v_k)\}$ be the set of directed links started from
        $u$, where $(u,v_i)\in T$ ,$1\leq i\leq k$ and $load(v_1)\geq\ldots\geq load(v_k)$;
6.      For ($i=1; i\leq k; i++$) {
7.          Check if $(u,v_i)$ can share the channel $c_h$ assigned to a sibling link
     $(u,v_j)$ without causing any interference, where $1\leq j\leq i-1$;
8.          If it is yes, then {
9.              Channel[$(u,v_i)$]=$c_h$; Load_based_DFS($v_i$); }
10.         Else {
11.             Try to find an interference-free channel $c_k\in\Phi$ for $(u,v_i)$;
12.             If it is successful, then {
13.                 Channel[$(u,v_i)$]=$c_k$; Load_based_DFS($v_i$); }
14.             Else { $\overline{E}=\overline{E}-(u,v_i)$; }
15. }   } }
Output: A MRDCM tree $T_{mrdcm}$
```

Fig. 7. A load-based DFS channel assignment procedure.

Since our goal is to find a *MRDCM* tree with the largest *gain* value in equation (A), a link lead to a node with larger *load* value should have higher priority to be allocated a channel. Hence, at line 5 in **Fig. 7**, the sibling links are sorted based on *load* values stored at terminal points of these links. And, at line 6, the sorted links are then allocated channels in sequence. For a new link e_i processed at line 7, a common channel allocated to its sibling links should be considered first. This step is to exploit the wireless broadcast advantage (WBA). If it fails, an interference-free channel is searched for e_i at line 11. The link e_i is removed if no such channel can be found. This CA process is then recursively executed in DFS fashion at line 9 and 13. This process takes $O(|V|^2\log(|V|))$ steps to finish, since there are $O(|V|)$ recursive calls for “Load_based_DFS” at the worst case and it takes $O(|V|\log(|V|))$ steps to do link sorting at line 5 for each recursive call.

4.3.3 The Load-based BFS Channel Assignment

The procedure of load-based BFS channel assignment is similar to the load-based DFS procedure. However, for multicast trees with large height, the interference-free channels may be used up before the links connected to destinations have a chance to be allocated a channel, since the nodes are visited in breadth-first order in load-based BFS procedure. Therefore, load-based BFS procedure is more suitable for the wide-and-short multicast trees.

5. The Experimental Results

A number of experiments are designed in this section to do the performance and delay comparisons for the *MRDCM* trees, which are generated by applying the load-based DFS channel assignment procedure to the multicast trees constructed by the LMCM, the load-based greedy and the shortest-path (SP) algorithms. The first two algorithms are described in section 4.1 and 4.2. As for the SP algorithm, it constructs the multicast tree by merging all the shortest paths from the gateway node to all the destinations. Each shortest path is determined by the Dijkstra's algorithm. The multicast trees found these three algorithms are processed by the same load-based DFS channel assignment procedure to produce the desired *MRDCM* trees.

Each data presented here is taken on the average of 100 runs on 100-node, 50-node and 30-node random networks generated on a 1250×1250 grid. The simulations are based on IEEE 802.11b/g standard and parameter settings are given in Table 3. All the programs are written in C++ and can be efficiently executed on laptop computers.

Table 3. Simulation settings.

Simulation parameter	value
transmission range	R=250 units
interference range	2R=500 units
destination ratio	10% ~ 50% of total nodes
subscribers	1 ~ 5
delay of each link	1 ~ 5
radio interface of each node	2
channels	11
antenna	omni-directional

5.1 Performance Comparisons

For measuring the performances of three tested algorithms, a metric term "performance ratio" θ is defined as follows:

$$\theta = \frac{gain}{\Gamma} \times 100\% \tag{4}$$

where $gain$ is defined in equation (A) in section 3, and Γ is the number of total subscribers initially assigned to the given networks. Note that Γ is an upper bound for the optimal value.

Since the *MRDCM* is a NP-complete problem, only optimal values of very small-size networks can be found. Therefore, we use Γ to substitute for the optimal value to do performance comparisons for tested algorithms.

(1) CA with orthogonal and partially overlapping channels

According to the results shown in **Fig. 8**, **Fig. 9**, and **Fig. 10**, the performance ratio θ of LMCM method is consistently superior to the other two methods for various destination ratios in different sizes of networks when the path delay is set to be 15. We also note that the difference of performance ratios between LMCM and the other two methods becomes larger as the number of nodes increases. That is, the LMCM is more suitable for the *MRDCM* problem in large-size networks. Obviously, the design paradigm of LMCM, "taking advantage the nature of wireless broadcast", has a good impact for exploring more destinations with the same number of channels. As a result, the number of subscribers serviced by the LMCM is the largest.

Since the height of multicast tree is greatly affected by the delay bound, we can expect that the performance ratios are inversely proportional to the delay bounds. That is, the performance ratio decreases as the delay bound increases. This phenomenon is verified in **Fig. 11** where the simulations for LMCM algoithm on 100-node networks is tested.

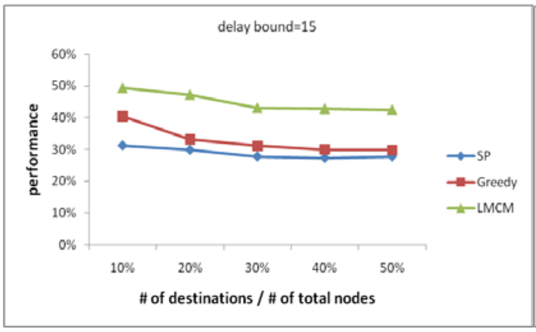

Fig. 8. Performance comparisons for 100-node networks.

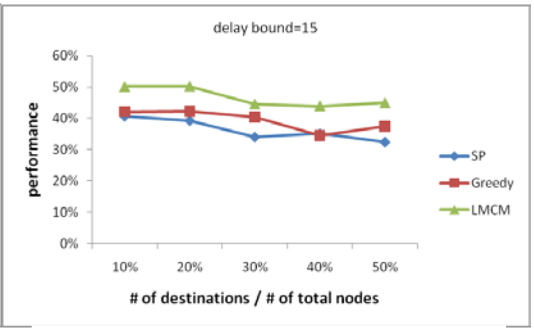

Fig. 9. Performance comparisons for 50-node networks.

(2) CA with orthogonal channels only

In this set of experiments, we study the impact on the performance for the *MRDCM* problem when only orthogonal channels can be allocated to the links of the multicast trees. As the results shown in **Fig. 8** and **Fig. 12**, the average value of performance ratios generated by the CA with all the channels available is about twice as large as that generated by the CA with orthogonal channels only for the LMCM algorithm. The same phenomenon holds for the other two algorithms. This improvement of performance ratio is resulted from the fact that the number of channels available for channel assignment is increased. Hence, we can conclude that compared to the previous approaches the CA method proposed in this paper has made great improvements for the multicasting supporting interference-free transmissions.

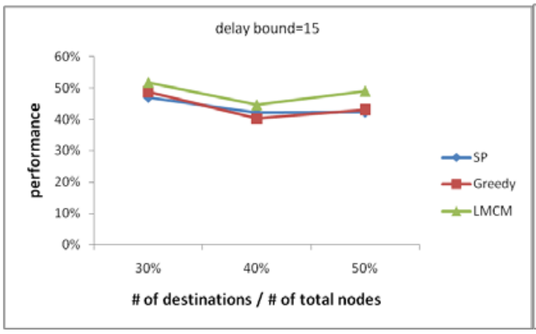

Fig. 10. Performance comparisons for 30-node networks.

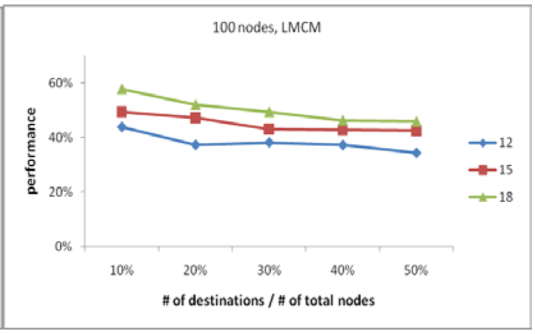

Fig. 11. Various Delay bounds for 100-node networks by LMCM method.

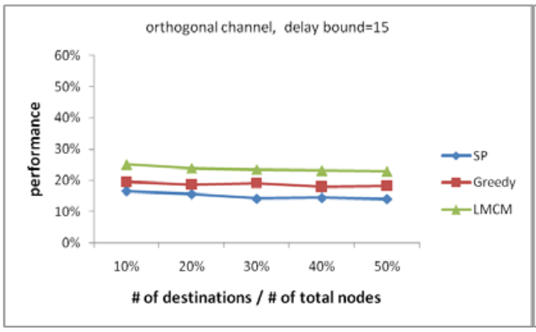

Fig. 12. CA with orthogonal channels only for 100-node networks.

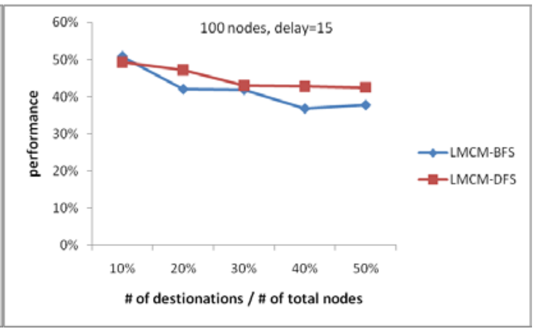

Fig. 13. DFS-based CA vs BFS-based CA.

(3) DFS-based CA vs BFS-based CA

In **Fig. 13**, the same set of multicast trees built by LMCM is used for allocating conflict-free channels based on two different CA procedures proposed in section 4.3.3. The performance of DFS-based CA procedure is consistently better than that of BFS-based CA procedure. The experimental results matches the obesrvation discussed in section 4.3.3 that DFS-based CA procedure is more suitable for multicast trees with large multicast goup since they intend to have large tree height.

5.2 Delay Comparisons

For delay-sensitive applications deployed on the multicast trees, the maximal path delay must not only satisfy the given delay constraint but also is required to be as small as possible. According to the results shown in **Fig. 14**, the average delay of *MRDCM* trees produced by the SP algorithm is the smallest, whereas the average delay of the LMCM method is the largest and very close to that of greedy method for various destination ratios for 100-node networks. The same phenomenon holds for the different sizes of networks (see **Fig. 15**).

This phenomenon is due to that the number of serviced users found by LMCM algorithm is much larger than that found by SP method. Hence, for a given network, the multicast tree constructed by LMCM tends to be the largest. As a result, the delay of the tree built by LMCM is the largest. In conclusion, the LMCM would be the best heuristic method for solving the *MRDCM* problem when both gain and delay are taken into account simultaneously.

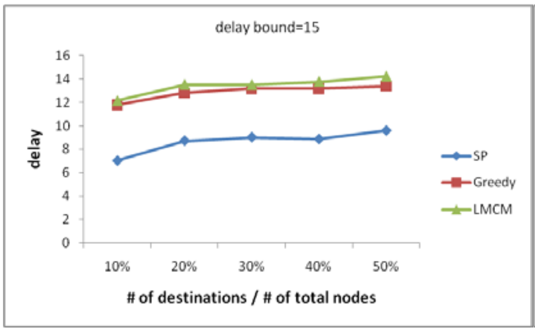

Fig. 14. Delay comparisons for 100-node networks.

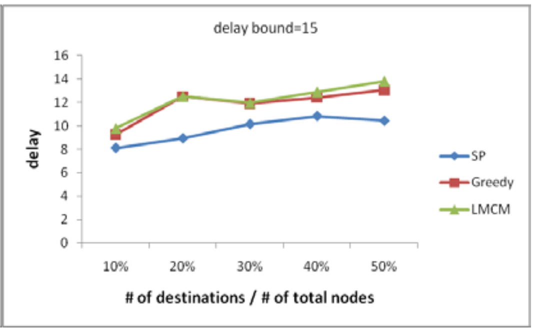

Fig. 15. Delay comparisons for 50-node networks.

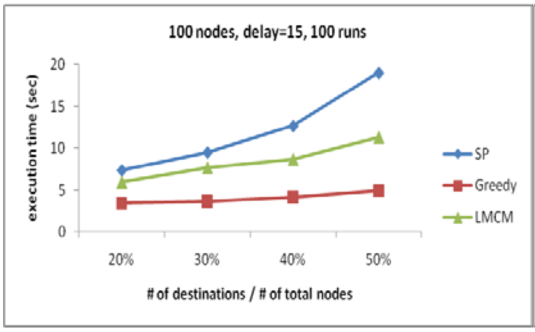

Fig. 16. Efficiency comparisons for 100-node networks.

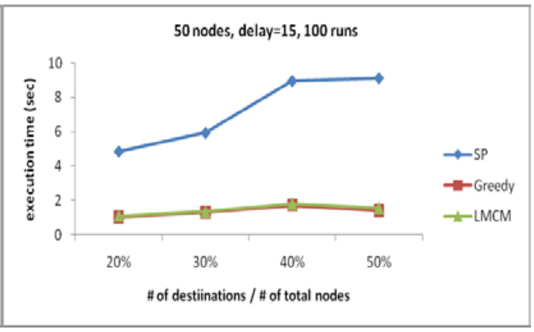

Fig. 17. Efficiency comparisons for 50-node networks.

5.3 Efficiency

According to the computation complexity analysis shown in the section 4, the running time of the LMCM and load-based greedy algorithms is $O(|V|^2 \times \log|V| \times L)$ and $O(|V|^2 L)$ respectively. As for the SP method, its time complexity is $O(|V|^3)$ since Dijkstra's algorithm takes $O(|V|^2)$ steps to find the shortest path for each destination. Based on these analytical results, the load-based greedy should be the most efficient method among these three approaches. In general, the tree height L is bounded by $O(\log|V|)$ for random large-size networks. Consequently, the executing efficiency of LMCM is better than that of SP. A set of experiments designed in this subsection is used to verify these analyses.

For each set of simulations, the summation of 100 runs CPU time is used for comparing executing efficiency for SP, load-based greedy and LMCM algorithms. As the results shown in **Fig. 16**, the execution time required by the LMCM is the second largest and slightly increases along with the multicast group size; even so, the LMCM is still an efficient and practical approach for large-size networks. For 100-node networks with 50% nodes being the destinations, for instance, the execution time required by LMCM is 0.1sec, where it is measured on a portable computer with Intel Core2 P7350 CPU. Whereas for small-size networks, the execution time required by the LMCM and load-based greedy methods is almost the same as the results shown in **Fig. 17** for the 50-node networks.

6. Conclusion

In this paper, we address the *MRDCM* problem that concerns about the delivery of real-time streams to a group of users based on multicast trees embedded in multi-radio WMNs. According to the results from the prior studies on interference factor, two load-based heuristic algorithms and CA procedures are developed for constructing an interference-free and delay-constrained multicast tree with all the available channels so that our objective function, the number of serviced clients, can be maximized.

Based on the experimental results, the LMCM would be the best heuristic method for solving the *MRDCM* problem when both the number of serviced subscribers and delay are considered at the same time. Although our load-based heuristics are centralized, they are still practical for our *MRDCM* problem since the number of clients serviced by each MR may remain a period of time. In fact, centralized approaches have been widely accepted in the industry [19]. However, we still need a distributed method to support MRs join or leave the multicast tree. We assume that a common channel is shared by one reserved radio installed on each MR. Based on the common channel, a new MR may join the tree by broadcasting "join" messages to it neighbors. The nearest MRs in the tree may send the acknowledges back to the original MR along the reverse path. A MR in the tree may send "leave" messages to inform the gateway when all the mesh cleints under it are left.

References

[1] I.F. Akyildiz, X. Wang and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445-487, 2005. Article(CrossRef Link)

[2] A.P. Subramanian, H. Gupta, S.R. Das and J. Cao, "Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 12, pp. 1459-1473, 2008. Article(CrossRef Link)

[3] A. Mishra, E. Rozner, S. Banerjee and W. Arbaugh, "Exploiting Partially Overlapping Channels in Wireless Networks: Turning a Peril into an Advantage," in *Proc. of Internet Measurement Conference*, pp. 311-316, 2005. Article(CrossRef Link)

[4] A. Kumar K.S and S. Hegde, "Multicasting in Wireless Mesh Networks: Challenges and Opportunities," in *Proc. of International Conf. on Information Management and Engineering*, pp. 514~518, 2009. Article(CrossRef Link)

[5] V. Bhandari and N. H. Vaidya, "Connectivity and Capacity of Multi-Channel Wireless Networks with Channel Switching Constraint," in *Proc. of INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 785- 793, 2007. Article(CrossRef Link)

[6] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network," in *Proc. of INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, pp. 2223-2234, 2005. Article(CrossRef Link)

[7] C. Chereddi, P. Kyasanur and N. H. Vaidya, "Net-X: A Multichannel Multi-Interface Wireless Mesh Implementation," *Mobile Computing and Communications Review*, vol. 11, no. 3, pp. 84-95, 2007. Article(CrossRef Link)

[8] L. Kou, G. Markowsky and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Infomatica*, Spring-Verlag, vol.15, pp. 141-145, 1981. Article(CrossRef Link)

[9] M.K. Marina and S.R. Das, "A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks," in *Proc. of Broadband Networks*, pp. 381-390, 2005. Article(CrossRef Link)

[10] R. Draves, J. Padhye and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *Proc. of MobiCom*, pp. 114-128, 2004. Article(CrossRef Link)

[11] Wen-Lin Yang, "A Maximum-Revenue Multicast Routing Problem on Wireless Mesh Networks," in *Proc. of the 24th International Conf. on Information Networking*, 2010.

[12] G. Zeng, B. Wang, Y. Ding, L. Xiao and M. Mutka, "Multicast Algorithms for Multi-Channel

Wireless Mesh Networks," in *Proc. of International Conf. on Network Protocols*, pp. 1-10, 2007. Article(CrossRef Link)

[13] P.M. Ruiz and A.F. Gomez-Skarmeta, "Approximating Optimal Multicast Trees in Wireless Multihop Networks," in *Proc. of IEEE Symposium on Computers and Communications*, pp. 686-691, 2005. Article(CrossRef Link)

[14] U.T. Nguyen and J. Xu, "Multicast Routing in Wireless Mesh Networks: Minimum Cost Trees or Shortest Path Trees," *IEEE Communications Magazine*, pp. 72-75, 2007. Article(CrossRef Link)

[15] H.L. Nguyen and U.T. Nguyen, "Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks," in *Proc. of International Conf. on Wireless Communications and Mobile Computing*, pp. 626-631, 2008. Article(CrossRef Link)

[16] A. Raniwala, K. Gopalan and T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks," *ACM Mobile Computing and Communications Review*, vol.8, no. 2, pp. 50-65, 2004. Article(CrossRef Link)

[17] Z. Yin, Z. Li and M. Chen, "A Novel Assignment Algorithm for Multicast in Multi-radio Wireless Mesh Networks," in *Proc. of IEEE Symposium on Computers and Communications (ISCC)*, pp. 283-288, 2007. Article(CrossRef Link)

[18] H.L. Nguyen and U.T. Nguyen, "Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks," in *Proc. of the International Conf. on Ultra Modern Telecommunications* (*ICUMT*), 2009. Article(CrossRef Link)

[19] *Tropos Networks*, http://www.tropos.com .

Wen-Lin Yang received his PhD degree and MS degree in Computer Science from the Pennsylvania State University and State University of New York, USA, in 1993 and 1988, respectively. Currently, he serves as the Dean of College of Science and Enginnering as well as Professor at the Department of Computer Science and Information Engineering in the National University of Tainan, Taiwan. His primary research interests include routing protocols, quality of service, wireless networks and distributed computing.

Chi-Chou Kao received a BS in engineering science from National Cheng-Kung University, Taiwan, in 1990, and MS and PhD degrees in electrical engineering from National Cheng-Kung University, Taiwan, in 1996 and 2002, respectively. Dr. Kao received the Long-Term Distinguished Paper Award, Acer Foundation, in 2003, and appears in the 23rd 2006 and 24rd 2007 edition of Who's Who in the World. His research interests include graph algorithms, combinatorial optimization, and circuit design. He is currently an Associate Professor in the Department of Computer Science and Information Engineering, National University of Tainan, Taiwan.

Cheng-Huang Tung was born on May 28, 1967, in Tainan, Taiwan, ROC. He received his BS and PhD from National Chiao Tung University in 1989 and 1994, respectively, both in Computer Science and Information Engineering. He has been with the Department of Computer Science and Information Engineering, National Pingtung Institute of Commerce, Taiwan, since August 2001. His research interests include natural language processing, pattern recognition, image processing, and computer network.