Joint Multicast Routing and Channel Assignment in Multiradio Multichannel Wireless Mesh Networks Using Simulated Annealing

Hui Cheng and Shengxiang Yang

Department of Computer Science, University of Leicester
University Road, Leicester LE1 7RH, United Kingdom
{hc118,s.yang}@mcs.le.ac.uk

Abstract. This paper proposes a simulated annealing (SA) algorithm based optimization approach to search a minimum-interference multicast tree which satisfies the end-to-end delay constraint and optimizes the usage of the scarce radio network resource in wireless mesh networks. In the proposed SA multicast algorithm, the path-oriented encoding method is adopted and each candidate solution is represented by a tree data structure (i.e., a set of paths). Since we anticipate the multicast trees on which the minimum-interference channel assignment can be produced, a fitness function that returns the total channel conflict is devised. The techniques for controlling the annealing process are well developed. A simple yet effective channel assignment algorithm is proposed to reduce the channel conflict. Simulation results show that the proposed SA based multicast algorithm can produce the multicast trees which have better performance in terms of both the total channel conflict and the tree cost than that of a well known multicast algorithm in wireless mesh networks.

1 Introduction

Wireless mesh networks (WMNs) [1] have emerged as a new paradigm of static multi-hop wireless networks. Multicast [2] is an important network service, which is the delivery of information from a source to multiple destinations simultaneously. Quality of Service (QoS) requirements [2] proposed by different multicast applications are often versatile. Among them, end-to-end delay [3] is a pretty important QoS metric since real-time delivery of multimedia data is often required. The multicast tree cost, used to evaluate the utilization of network resource, is also an important QoS metric especially in wireless networks where limited radios and channels are available. So far, few work has addressed QoS multicast in WMNs. However, it is believed that efficient multicast, which cannot be readily achieved through combined unicast or simplified broadcast, is essential to WMNs and deserves a thorough investigation [4].

In WMNs, the wireless interference occurs when two links whose distance is less than 2 hops away are assigned to the same channel to support the concurrent communications, which is termed as channel conflict [5]. Therefore, for

X. Li et al. (Eds.): SEAL 2008, LNCS 5361, pp. 370–380, 2008.

multicast routing, each link on the multicast tree requires to be assigned to one channel and the assignment should lead to minimum interference. In fact, the minimum-interference channel assignment problem itself is basically the Max K-cut problem [6], which is known to be NP-hard. Therefore, our problem, i.e., the routing tree construction plus minimum-interference channel assignment, is also NP-hard. In this paper, we propose an efficient QoS multicast routing algorithm in WMNs, which utilizes the powerful simulated annealing (SA) technique to search a low cost routing tree on which the channel assignment can produce the minimum interference. Intuitively by exploring the strong search capability of SA, more candidate routing trees can be examined to help find the one with the minimum channel conflict.

2 Related Work

In [4], two multicast algorithms were proposed, which first build a multicast tree between the source and receivers, and use dedicated strategies to assign channels to the tree aiming to reduce interference. However, since both algorithms separate the construction of the multicast tree and the channel assignment, they will bear a potential drawback, that is, channel assignment cannot work well with the determined multicast tree. Furthermore, they do not consider the delay constraint which is a common issue for multicast problems.

Simulated annealing algorithm simulates the annealing process in the physics of solids. It is observed that a metal body heated to high temperature cools slowly and tends to a state with the least internal energy. SA regards the optimization problem as a physical system and the value of the objective function as its internal energy. With this analogy, annealing is the process of determining a solution with the least value of the objective function [7]. Simulated annealing algorithm is a powerful tool to solve the combinatorial optimization problems. It has been applied to the QoS multicast routing in the wired networks such as the multimedia communication networks [2,8].

We are not aware of any other work that jointly considers multicast routing, which further consists of channel assignment as well as QoS in multiradio multichannel WMNs, although there are quite a few works that are related to some relevant aspects. Since SA has shown good performance in the wired networks, we believe its strong search capability can also help find a low cost low interference routing tree in wireless mesh networks. However, to our best knowledge, SA has not been addressed to solve the multicast problems in WMNs.

3 Problem Formulation

We consider a wireless mesh network with stationary mesh routers where each router is equipped with a certain number of radio network interface cards (NICs). We model a wireless mesh network by a undirected and connected topology graph $G(V, E)$, where V represents the set of mesh routers and E represents the set of communication links connecting two neighboring mesh routers falling into the radio transmission range. A communication link (i, j) can not be used for packet transmission until both node i and node j have a radio interface each with a common channel. In addition, message transmission on a wireless communication link will experience a remarkable delay.

For clarity of presentation, we assume the *binary interference model*, i.e., two communication links either interfere or do not interfere. Given the binary interference model, the set of pairs of communication links that interfere with each other over the same channel can be represented by a conflict graph [5]. A communication link in the topology graph corresponds to a vertex in the conflict graph. With the binary interference model, the conflict graph $G_c(V_c, E_c)$ can be easily derived from the topology graph $G(V, E)$. We assume the communication links (a, b) and (c, d) in the topology graph $G(V, E)$ are represented by the node i_c and node j_c in the conflict graph $G_c(V_c, E_c)$, respectively. Then if the minimum distance between (a, b) and (c, d) is less than 2 hops, we have $(i_c, j_c) \in E_c$. Here, we summarize some notations that we use throughout this paper.

- $G(V, E)$, the WMN topology graph.
- $G_c(V_c, E_c)$, the conflict graph derived from the WMN topology graph.
- $K = \{0,1,2,...,k\}$, the set of available orthogonal channels.
- s, the source node of the multicast communication.
- $R = \{r_0,r_1,...r_m\}$, the set of receivers of the multicast communication.
- $T(V_T, E_T)$, a multicast tree with nodes V_T and links E_T.
- V_T^{Leaf}, the set of leaf nodes on the tree T.
- $P_T(s, r_i)$, a path from s to r_i on the tree T.
- d_l, the delay on the communication link l.
- $I_T(f)$, the total channel conflict on the tree T.
- C_T, the cost of the tree T.

The problem of joint QoS multicast routing and channel assignment in a multiradio multichannel wireless mesh network can be informally described as follows. Given a network of mesh routers with multiple radio interfaces, a delay upper bound, a source node and a set of receivers, we wish to find a delay-bounded multicast tree and assign a unique channel to each communication link on the tree. We define the *total channel conflict* as the number of pairs of communication links on the tree that are interfering (i.e., are assigned the same channel and are connected by an edge in the conflict graph). The objective of our problem is to minimize the above defined total channel conflict, as it results in improving the system throughput [4].

We also want to optimize the usage of the scarce network resources in the multicast tree. So we define the *tree cost* as the number of the radio interfaces involved in the multicast communications. We aim to find a multicast tree with low cost. There are two factors related to the tree cost. One is the number of communication links on the tree. Each communication link has one sender and one receiver, thereby occupying two radio interfaces. So we should reduce the number of links on the multicast tree, which also helps reduce the multicast end-to-end delay. The other factor is the number of broadcast nodes generated from the channel assignment. We make all the branch nodes become broadcast nodes by exploiting wireless multicast advantage (WMA) [9] and the detail is described in Section 4.2. If there are several multicast trees which have the same channel conflict value, we will choose the one with the minimum tree cost.

More informally, consider a wireless mesh network $G(V, E)$ and a multicast communication request from the source node s to a set of receivers R with the delay upper bound Δ. The *joint QoS multicast routing and channel assignment problem* is to find a multicast tree $T(V_T, E_T)$ satisfying the delay constraint as shown in (1) and compute a function $f : E_T \rightarrow K$ defined in (2) to minimize the *total channel conflict* $I_T(f)$ defined in (3).

$$\max_{r_i \in R} \left\{ \sum_{l \in P_T(s, r_i)} d_l \right\} \leq \Delta \,. \tag{1}$$

$$f(i_c \in E_T) = \{j | j \in K\} \tag{2}$$

$$I_T(f) = |\{(i_c, j_c) \in E_c | f(i_c) = f(j_c), i_c \in E_T, j_c \in E_T\}| \,. \tag{3}$$

Since the source only transmits packets and all the leaf nodes only receive packets, each of them occupies one radio interface only. All the other nodes are branch nodes which need to do both the transmission and reception. So each branch node occupies two radio interfaces. As a result, the tree cost C_T is calculated as follows:

$$C_T = |\{s\}| + |V_T^{Leaf}| + 2 * (|V_T| - |\{s\}| - |V_T^{Leaf}|) \,. \tag{4}$$

4 Algorithm Design

We adapt SA to the joint multicast routing and channel assignment problem, and the objective function is just the fitness function, which returns the total channel conflict of the multicast tree. The fitness value just simulates the internal energy. First, the initial solution is generated randomly to explore the diversity. Then we start the annealing process at a high temperature. As the temperature decreases, the annealing process tries to converge to the optimal solution. At each temperature, the algorithm searches a number of solutions in the solution space so that the current optimal solution stabilizes at a fitness value. When the temperature decreasing number reaches a specified upper bound and the current optimal solution keeps unchanged, the algorithm terminates and outputs the current optimal solution as the final solution.

4.1 Design of the Simulated Annealing Algorithm

Solution Representation and Initial Solution. A routing path is encoded by a string of positive integers that represent the IDs of nodes through which the path passes. Each locus of the string represents an order of a node. The first locus is for the source and the last one is for the receiver. The length of a routing path should not exceed the maximum length $|V|$, where V is the set of nodes in the WMN.

For a multicast tree T spanning the source s and the set of receivers R, there are $|R|$ routing paths all originating from s. Therefore, we encode a tree by an integer array in which each row encodes a routing path along the tree. For example, for T spanning s and R, row i in the corresponding array A lists up node IDs on the routing path from s to r_i along T. Therefore, A is an array of $|R|$ rows. All the solutions are encoded under the delay constraint. In case it is violated, the encoding process is usually repeated so as to satisfy the delay constraint.

To explore the solution diversity, in the initial solution Q, all the routing paths are randomly generated. We start to search a random path from s to $r_i \in R$ by randomly selecting a node v_1 from $N(s)$, the neighborhood of s. Then we randomly select a node v_2 from $N(v_1)$. This process is repeated until r_i is reached. Thus, we get a random path $P_T(s, r_i)=\{s, v_1, v_2, ..., r_i\}$. Since no loop is allowed on the multicast tree, the nodes that are already included in the current tree are excluded, thereby avoiding reentry of the same node. The initial solution is generated as follows.

Step 1: Start(j=0, $V_T=\emptyset$, $E_T=\emptyset$);

Step 2: Search a random path $P_T(s, r_i)$ which can guarantee $T \cup P_T$ be an acyclic graph;

Step 3: Add all the nodes and links in P_T into V_T and E_T, respectively;

Step 4: $j = j+1$. If $j < |R|$, go to *Step 2*, otherwise, stop.

Fitness Function. Given a solution, we should accurately evaluate its quality (i.e., fitness value), which is determined by the fitness function. In our algorithm, we aim to find a low cost multicast tree on which the minimum interference channel assignment can also be achieved. Our primary criterion of solution quality is the total channel conflict and the subsidiary one is the tree cost. Therefore, among a set of candidate solutions (i.e., multicast trees) with the same minimum channel conflict value, we choose the one with the lowest tree cost. The fitness value of chromosome Ch_i (representing multicast tree T), denoted as $F(Ch_i)$, is given by:

$$F(Ch_i) = [I_T(f) + 1.0]^{-1} . \qquad (5)$$

The proposed fitness function only involves the total channel conflict. As mentioned in Section 3, The tree cost is used in the course of selecting the elitism for keeping the searched optimal solution.

Neighborhood Structure. Since SA performs searching from one solution to one of its neighbors in the solution space, we need to determine the neighborhood structure of each solution. In accordance with the solution representation, we propose two methods to construct the neighborhood.

(a) First, randomly select one receiver r_i from R, and randomly select another node v_i on the path $(s \longrightarrow r_i)$. Then replace the subpath $(v_i \longrightarrow r_i)$ by a new random subpath.

(b) First, randomly select two receivers r_i and r_j from R, and randomly select another two nodes v_i and v_j on the paths $(s \longrightarrow r_i)$ and $(s \longrightarrow r_j)$, respectively. Then replace the subpaths $(v_i \longrightarrow r_i)$ and $(v_j \longrightarrow r_j)$ by new random subpaths, respectively.

Given the current solution, a new neighbor solution will be produced using either of the above two methods. The first method only changes one path on the tree while the second method changes two paths at the same time. Intuitively, the adjustment to the tree is relatively smaller in (a) than in (b). So we name the first method as the fine-grain adjustment and the second method as the coarse-grain adjustment. In the proposed algorithm, we apply the fine-grain adjustment in the first half of the temperature decreasing procedure, and then the coarse-grain adjustment in the second half of the temperature decreasing procedure. Therefore, we can ont only guarantee the algorithm converges to the optimal solution theoretically, but also accelerate the procedure to improve the efficiency.

Initial Temperature. We start the SA algorithm from a high temperature (T_0) in order to allow acceptance of any new neighbor solution. A reasonable setting of the initial temperature will reduce the waste of the search time and still allow virtually all proposed uphill or downhill moves to be accepted [8]. In this algorithm, we set $T_0 = 100$.

Temperature Decreasing. We employ the following method:

$$T_{k+1} = \alpha * T_k \ (0 \leq k, 0 < \alpha < 1) \ . \tag{6}$$

This method is widely used, simple but effective. By this method, the temperature decreases at the same ratio.

Iterative Length at Each Temperature. In our algorithm, the iterative length at one temperature is proportional to the number of temperature decreasing counted so far. We use L_i to denote the maximum iteration number allowed at temperature T_i, and M_i to denote the maximum number of continuous iterations without improving the present optimal solution allowed at T_i. As the temperature gradually decreases to T_i, both L_i and M_i should become larger simultaneously to explore more candidate solutions in the solution space.

We employ the method of linear increasing, that is, the maximum iteration number allowed at temperature T_i is in direct proportion to the up-to-now times of temperature decreasing, and the maximum number of continuous iterations without improving the present optimal solution allowed at T_i is in direct proportion to the maximum iteration number allowed at the same temperature. The method is formulated as follows:

$$L_i = (i + 1) * \delta * \tau \ . \tag{7}$$

$$M_i = \omega * L_i \ . \tag{8}$$

Here, τ is the size of the receiver set, serving as the cardinal number. Since in each iteration, we need to change the path to one receiver. Ideally, we hope the paths to all the receivers will undergo the change at the same temperature. L_i limits the iteration number at the same temperature to speed up the convergence, and M_i helps stop the iteration at T_i since the search may be stuck in the local optimum.

Termination Rule. The termination rule employed in this algorithm is to control the maximum number of continuous temperature decreasing without improving the present optimal solution. Let the maximum number of temperature decreasing be I, and the upper bound of the continuous temperature decreasing without improving the present optimal solution be U. They have the following relationship:

$$U = \lambda * I \ (0 < \lambda < 1) . \tag{9}$$

In the proposed algorithm, during the first half period of temperature decreasing, i.e., from T_0 to $T_{\lfloor I/2 \rfloor}$, we generate a neighbor solution by the fine-grain method; during the second half period of temperature decreasing, i.e., from $T_{\lfloor I/2 \rfloor+1}$ to T_I, we generate a neighbor solution by the coarse-grain method. During the first half period, it is more likely that the difference between the current solution and the global optimal solution is relatively large. So we change two paths to two receivers at each iteration. During the second half period, the difference may become smaller. So we change only one path at each iteration. This design philosophy can help reduce the overhead of the fitness function calculation. Moreover, the algorithm can be theoretically assured to find the global optimal solution as the iteration approaches infinity.

4.2 Channel Assignment Algorithm

In a wireless mesh network, a link cannot be used for data transmission until it has been assigned a wireless communication channel. To support the multicast communication over the routing tree, an appropriate channel should be assigned to each link on the tree so as to achieve the minimum interference (i.e., channel conflict). In addition, the number of available channels is limited in the current network protocols. For example, in IEEE 802.11-based wireless networks, there are 11 available channels. However, at most 3 of them are orthogonal (non-interfering). The number of radio interfaces is also limited as a type of scarce radio network resource. Hence the channel assignment should use as small number of channels and radio interfaces as possible.

Since the minimum-interference channel assignment problem is NP-hard, we propose a heuristic algorithm which aims to reduce both the channel conflict and resource utilization. Given the set of orthogonal channels K=$\{0,1,...,k\}$ ($k \geq 2$), the algorithm works on the multicast tree T as follows.

Step 1) Start(i=0).

Step 2) Assign channels to the routing path $P_T(s, r_i)$=$(s, v_1, v_2, ..., v_{j-1}, r_i)$. Here v_0 represents the source s and v_j represents the receiver r_i, respectively.

a) Start(n=0);

b) If link (v_n, v_{n+1}) has not been assigned a channel, assign channel n%3 to it;

c) n=n+1. If $n < j$, go to *b*).

Step 3) i=i+1. If $i < |R|$, go to *Step 2*, otherwise, stop.

For each routing path, the algorithm uses 3 channels to do the assignment. Since the minimum distance between two links to avoid channel conflict is 2 hops, 3 is the least number of channels to achieve conflict free assignment on each routing path of the multicast tree. By our assignment strategy, all the links originating from the same branch node are assigned the same channel as utilizes the so-called WMA [9]. WMA refers to that a single transmission can be received by all the nodes that are within the transmission range of a transmitting node. Hence, using one radio interface only, the branch node transmits packets to all its children. This also saves the number of used radio interfaces.

5 Performance Evaluation

In this section, the proposed SA-based joint multicast routing and channel assignment algorithm is compared with *Zeng'* Level Channel Assignment (LCA) multicast algorithm [4] through simulation experiments. The LCA multicast algorithm is composed of two components. First, it constructs a multicast tree based on breadth first search (BFS) aiming to minimize the hop count distances between the source and receivers. Second, it uses a dedicated strategy to assign channels to the tree aiming to reduce the interference. Hence, this algorithm separates the multicast tree construction and channel assignment. If the channel assignment strategy cannot work well on the generated multicast tree, the algorithm can do nothing while our algorithm can search other trees.

In our algorithm, we have the following SA parameters: the initial temperature (T_0), the coefficient of temperature decreasing (α), the coefficient of the maximum iteration number allowed at one temperature (δ), the coefficient of the maximum number of continuous iterations without improving the present optimal solution allowed at one temperature (ω), and the coefficient of the maximum number of continuous temperature decreasing without improving the present optimal solution (λ). Their suggested values are set in Table 1. In addition, the delay upper bound Δ is set to 20.

Without loss of generality, we assume each mesh router has two radio network interface cards: one for transmission and the other for reception. We assume there are 3 orthogonal channels as the case in 802.11 wireless network. We evaluate both algorithms on two different network topologies: one consists of 11 nodes and 20 links and the other consists of 23 nodes and 34 links. The metrics that we have evaluated include the total channel conflict and the tree cost.

We have compared the SA multicast algorithm with the LCA multicast algorithm over various size of multicast groups. In the WMN of 11 nodes, the size ranges from 3 to 7 whilst in the WMN of 23 nodes it ranges from 3 to 11. Fig. 1

Table 1. SA parameters and the suggested values

Parameter variable	Parameter description	Suggested value
T_0	the initial temperature	100
α	the coefficient of temperature decreasing	0.95
δ	the coefficient of the maximum iteration number allowed at one temperature	1
ω	the coefficient of the maximum number of continuous iterations without improving the present optimal solution allowed at one temperature	0.50
λ	the coefficient of the maximum number of continuous temperature decreasing without improving the present optimal solution	0.30

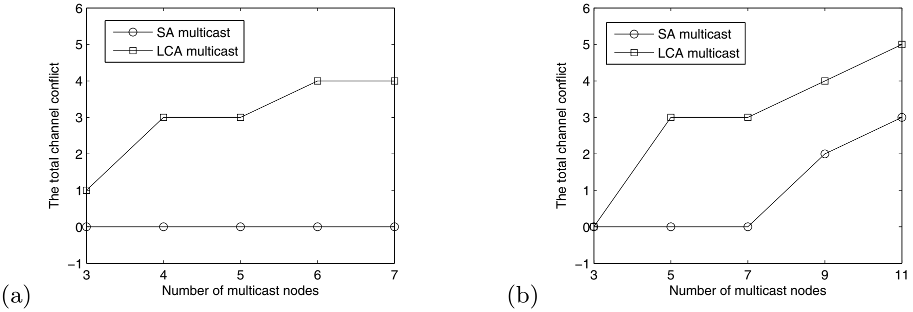

Fig. 1. Comparison of SA multicast and LCA multicast in terms of the total channel conflict in: (a) a WMN of 11 nodes; (b) a WMN of 23 nodes

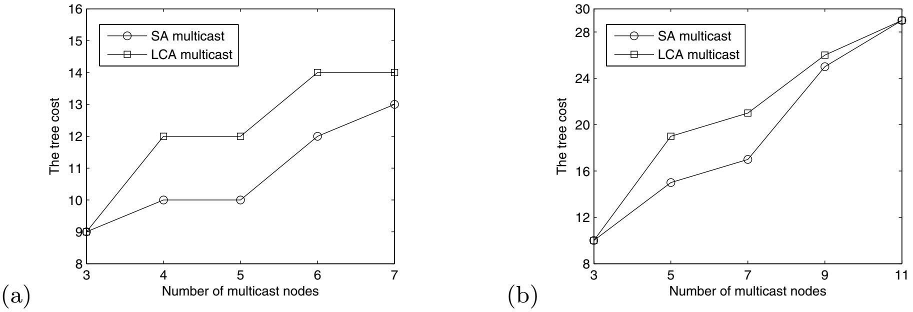

Fig. 2. Comparison of GA multicast and LCA multicast in terms of the tree cost in: (a) a WMN of 11 nodes; (b) a WMN of 23 nodes

and Fig. 2 show the comparison results in terms of the total channel conflict and the tree cost, respectively. From Fig. 1, we can see that in both networks, our SA multicast algorithm can find the multicast trees with less channel conflict than the trees obtained by the LCA multicast algorithm. In the network of 11 nodes, the SA multicast algorithm can always find the conflict-free multicast trees. Furthermore, with the increase of the multicast group size, the LCA multicast algorithm produces the multicast trees with more and more channel conflict. Fig. 2 shows that the cost of our SA multicast trees is also lower than the cost of the LCA multicast trees when the multicast group size exceeds 3. It means that the SA multicast trees occupy less radio network resources. To sum up, our SA multicast algorithm can find the multicast tree which logs less channel conflict and lower cost while satisfying the delay constraint.

6 Conclusions

A routing tree with orthogonal channels appropriately assigned is preferred to support the multicast service in WMNs. However, the optimal multicast routing and channel assignment problem is proved to be NP-hard. This paper presents a simulated annealing algorithm based joint multicast routing and channel assignment algorithm to discover a delay-bounded minimum-interference low cost multicast tree. We believe that the synergy achieved by combining the strong search capability of SA and the effective channel assignment results in the improved quality of solution. We compare the performance of the proposed algorithm with the prestigious LCA multicast algorithm. Experimental results demonstrated that our SA multicast algorithm is capable of finding the multicast trees which have both less channel conflict and lower cost (i.e., occupying less radio network interfaces) than the multicast trees produced by the LCA multicast algorithm.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/E060722/1.

References

1. Akyildiz, I., Wang, X., Wang, W.: Wireless Mesh Networks: a Survy. Comput. Netw. 47, 445–487 (2005)
2. Wang, X., Cao, J., Cheng, H., Huang, M.: QoS Multicast Routing for Multimedia Group Communications Using Intelligent Computational Methods. Comput. Comm. 29, 2217–2229 (2006)
3. Parsa, M., Zhu, Q., Garcia-Luna-Aceves, J.: An Iterative Algorithm for Delay-constrained Minimum-cost Multicasting. IEEE/ACM Trans. Netw. 6, 461–474 (1998)
4. Zeng, G., Wang, B., Ding, Y., Xiao, L., Mutka, M.: Multicast Algorithms for Multi-Channel Wireless Mesh Networks. In: Proc. ICNP 2007, pp. 1–10 (2007)

5. Jain, K., Padhye, J., Padmanabhan, V., Qiu, L.: Impact of Interference on Multi-hop Wireless Network Performance. In: Proc. MobiCom 2003, pp. 66–80 (2003)
6. Subramanian, A., Gupta, H., Das, S.: Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks. In: Proc. SECON 2007, pp. 481–490 (2007)
7. Levanova, T.V., Loresh, M.A.: Algorithms of Ant System and Simulated Annealing for the p-median Problem. Autom. Rem. Contr. 65, 431–438 (2004)
8. Zhang, K., Qi, Y., Zhang, H.: Dynamic Multicast Routing Algorithm for Delay and Delay Variation-bounded Steiner Tree Problem. Knowl.-Based Syst. 19, 554–564 (2006)
9. Thai, M., Li, Y., Du, D.: A Combination of Wireless Multicast Advantage and Hitch-hiking. IEEE Commun. Lett. 9, 1037–1039 (2005)