2009

Minimum interference channel assignment for multicast in multi-channel multi-radio wireless mesh networks

Sangil Choi
Iowa State University

Minimum interference channel assignment for multicast in multi-channel multi-radio wireless mesh networks

by

Sangil Choi

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Lu Ruan, Major Professor
Leslie Miller
Wensheng Zhang

Iowa State University

Ames, Iowa

2009

TABLE OF CONTENTS

LIST OF FIGURES iii

LIST OF TABLES iv

ACKNOWLEDGEMENTS v

ABSTRACT vi

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. LITERATURE REVIEW 3
- 2. 1 BFVC Algorithm 3
- 2. 2 MCM Algorithm 5
- 2. 3 M4 Algorithm 9

CHAPTER 3. MINIMUM INTERFERENCE CHANNEL ASSIGNMENT 11
- 3. 1 Step 1: Calculating channel separation of all pairs of nodes 11
- 3. 2 Step 2: Finding channel separation of zero 14
- 3. 3 Step 3: Assigning channels to all nodes 18

CHAPTER 4. SIMULATIONS 22
- 4. 1 Performance metrics 22
- 4. 2 Simulation Parameters 23
- 4. 3 Simulation results 24

CHAPTER 5. CONCLUSTIONS 30

BIBLIOGRAPHY 31

LIST OF FIGURES

Figure 1. An example of channel assignment by MCM 8
Figure 2. Calculating channel separation of all pairs of nodes 12
Figure 3. Multicast tree and children of *S* and *A* 14
Figure 4. Finding channel separation of zero 17
Figure 5. Assigning channels to all nodes 20
Figure 6. An example of channel assignment by MICA 21
Figure 7. Average packets received by multi-receivers 27
Figure 8. Standard deviation of all packets received by multi-receivers 28
Figure 9. Impact of number of multi-receivers 29
Figure 10. Delay comparison 29

LIST OF TABLES

Table 1. Interference factors in an IEEE 802.11b network 8
Table 2. Simulation parameters 24

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, I thank Dr. Lu Ruan for her guidance, patience and support throughout this research and the writing of this thesis. Her insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Leslie Miller and Dr. Wensheng Zhang.

ABSTRACT

Wireless mesh networks (WMNs) have emerged as a key technology for next-generation wireless networking. In a WMN, wireless routers provide multi-hop wireless connectivity between hosts in the network and also allow hosts to access the Internet via the gateway nodes. Wireless routers are typically equipped with multiple radios operating on different channels to increase network throughput. Multicast is a form of communication that delivers data from a source to a set of destinations simultaneously. It is used in a number of applications such as distributed games, distance education, and video conferencing. In this work, we address the channel assignment problem for multicast in multi-radio multi-channel WMNs. In a multi-radio multi-channel WMN, when two nearby nodes transmit on the same channel, they will interfere with each other and cause throughput decrease. Thus, an important goal for multicast channel assignment is to reduce the interference among the tree nodes. We have developed a Minimum Interference Channel Assignment (MICA) algorithm for multicast that accurately models the interference relationship between pairs of multicast tree nodes using the concept of interference factor and assigns channels to tree nodes to minimize interference within the multicast tree. Simulation results show that MICA achieves higher throughout and lower end-to-end packet delay compared with an existing channel assignment algorithm named MCM. In addition, MICA achieves much lower throughput variation among the destination nodes than MCM.

CHAPTER 1. INTRODUCTION

Wireless network is common environment for us in the world. Wireless devices are widely spread out and we can use those everyday. Especially, wireless mesh networks (WMNs) [1] are the most recently emerging technology that provides more reliability than mobile ad hoc networks (MANET). WMNs can still work when a node breaks down or a connection failed, so this network is very reliable. Typical WMNs usually consist of mesh clients, mesh routers and gateways. The mesh clients are often laptops, cell phones and other wireless devices while the mesh routers forward data packets to and from the gateway nodes which usually connect to the Internet. In WMNs, most of nodes are generally not mobile and fully connected. WMNs were originally designed for military applications but have experienced significant evolution in the past decade. WMNs can be implemented with various existing wireless technology including 802.11, 802.16, or cellular technologies.

Wireless mesh routers in WMNs are typically equipped with multiple interfaces operating on different channels to increase network throughput. In a multi-radio multi-channel WMN, when two nearby nodes transmit on the same channel, they may interfere with each other and cause throughput decrease. Therefore, we need an efficient solution to reduce network interference to improve network throughput significantly. One of the best ways to decrease interference in WMNs is always to assign a different channel to each wireless node instead of the same channel. However, we have no enough available number of channels to use at the same time. As a result, we need a very efficient and careful channel assignment mechanism to diminish the overall network interference in the network and improve network throughput. In this work, we have proposed a MICA algorithm for multicast that accurately models the interference relationship between pairs of multicast tree nodes using the concept of interference factor and assign channels to tree nodes to minimize interference within the multicast tree.

There are many studies on how to assign channels to nodes in WMNs [2-11]. All of these researches focus on unicast communications. On the other hand, channel assignment for multicast has only been addressed recently [12-14]. The channel assignment algorithm named Multi-Channel Multicast (MCM) [14] suffers from low throughput caused by the hidden channel problem (HCP) [12]. Our proposed algorithm in this thesis can get rid of HCP by considering every pair of nodes in the network. This algorithm allows the nodes in a multicast tree to work with minimum interference. Our simulation results show that MICA achieves higher throughput and lower end-to-end packet delay compared with MCM. In addition, MICA accomplishes much lower throughput variation among the destination nodes than MCM.

The rest of this thesis is organized as follows. In chapter 2 we review relevant literature in the area. We describe our proposed algorithm MICA in detail in chapter 3. In chapter 4, we show our simulation results. It focuses on comparing the performance of MICA with that of MCM. We end the thesis by providing conclusions from our work in chapter 5.

CHAPTER 2. LITERATURE REVIEW

In this section, we review some of the related literature in the area of channel assignment for multicast in multi-channel multi-radio WMNs. Recent studies on multicast in WMNs concentrate on multicast routing and performance study of routing algorithms in single-channel networks [15-19]. On the other hand, The channel assignment problem in multi-channel multi-radio WMNs has been studied substantially for unicast communications [2-11].

In particular, we focus on the problem of channel assignment for multicast in multi-channel multi-radio WMNs. Channel assignment problem for multicast has only been studied lately [12-14]. Yin et al. [13] proposed a novel channel assignment strategy called Unidirectional Channel Assignment Strategy (UCAS) based on unidirectional link model, and an efficient greedy vertex coloring algorithm called Breadth First Vertex Coloring (BFVC). In [14], authors presented the MCM algorithm and Nguyen et al. [12] provided a channel assignment algorithm named Minimum interference Multi-channel Multi-radio Multicast (M4).

2. 1 BFVC Algorithm

In this algorithm, channel assignment is based on unidirectional wireless link model. There exist two kinds of groups in this model. One is one-to-one link and the other is one-to-multiple one which correspond to unicast and multicast. If wireless node *u* wants to send a unicast packet to the destination node *v*, network interfaces of two nodes, *u* and *v*, should have the same channel c. For multicast communications, the same rule is applied to each interface of nodes participating in multicast communications. This strategy is UCAS. The BFVC algorithm uses a directed graph $G (V, E)$ to represent the connectivity between nodes in WMNs, where V represents the set of all nodes. For u, $v \in V$, $r (u, v)$ represents the communication range from radio interface u_i on a node u to interface v_j on a node v. There exists a direct link $e (u_i, v_j) \in E$ between a node u and v if and only if $0 < distance (u, v) \leq r (u_i, v_j)$. Therefore, for a link $e (u_i, v_j)$, if there exists a radio interface w_k sending packets at the same channel with u_i, then we say that there is an interference node w that conflict with a node u. In [13], authors show that the problem of channel assignment for multicast with minimum interference is NP-complete by translating it into a problem of k-vertex-multi-coloring [20]. It is known that the problem of vertex coloring with equal weight is NP-complete, so the problem of channel assignment for multicast with minimum interference is also NP-complete.

BFVC operates as follows. It visits all the nodes from the mesh gateway with the sequence of breadth-first. Whenever visiting each node, it calculates its interference values at different channels, and then assigns one channel to its radio interfaces with minimum interference. The following process illustrates how to compute the interference values at different channels for a node to be colored. Node u is the node to be colored, and the set $Ng(u)$ is the set of the neighbors of node u. For $v \in Ng(u)$, let $IF(v, c)$ represent the set of nodes which has an interface assigned to the channel c within the interference range of node v, and $P_t(w)$ is the probability that the node w sends packets with the radio interface assigned to the channel c. The interference value of node u at channel c $itf(u, c)$ can be represented as follow:

$$itf(u, c) = \sum_{v \in Ng(u)} \sum_{w \in IF(v,c)} p_t(w)$$

Although *itf* function depends on the use of the probability $P_t(w)$, this paper did not mention how to compute this probability. Also, collecting and maintaining this information causes high overheads.

2. 2 MCM Algorithm

The MCM algorithm is a channel assignment algorithm for multicast in multi-channel multi-radio wireless mesh network environments. This algorithm first constructs a multicast tree for multicast communications from a source to multi-receivers. In the multicast tree, there are three kinds of nodes: a source, relay nodes, and multi-receivers. A source node usually generates data packets and a relay node receives packets from its relay node at the upper layer and forwards them to its children at the lower layer. Lastly, there are several receivers that just receive data from its parent node. In this algorithm, they assumed each wireless node has two network interfaces: one for sending packets to its children and the other for receiving data from its relay one. So each node needs two channel information for these network interfaces. We called the network interface for receiving Receive-Interface (RI) and one for sending Send-Interface (SI). The main goal of MCM is to assign channel number to SI and RI respectively without network interference among wireless nodes in the multicast tree.

The MCM channel assignment algorithm works as follows. The first step is that the source node uses channel number 0 for its SI and its children use the same channel for their

RIs. Each node's RI is related to the SI of its upper layer node in order to guarantee that the relay node can communicate with its children. Therefore the RI of relay nodes should be the same channel as the SI of their parent's node. Next, this algorithm tries to assign the channel to the SI of the source's children. In this step, each node considers its one-hop neighboring nodes in order to set up the channel of its SI. If the neighboring nodes have been assigned channels for their SIs, those channel information is used to assign a new channel to the SI of the relay nodes at the same level in the multicast tree. MCM attempts to find out the available channels that minimize the following function:

$$\sum_{v \in S(u)} \delta^2_{|i_u - i_v|}$$

Here, u is the relay node that wants to receive a new channel for its SI and v is one of the neighboring nodes in the set $S(u)$. $S(u)$ is the set of one-hop neighbors of node u that have already been assigned a channel and i_u is the channel that is assigned to node u. $\delta_{|i_u - i_v|}$ is the interference factor [14] between two channels i_u and i_v. Interference factor is defined as the ratio of the interference range by the transmission range. The MCM algorithm uses this metric to estimate the level of interference between two nodes. The interference factor is closely connected with the channel separation [21] between two nodes. In the previous research, they conducted real experiments to measure the interference factor between two wireless peer-to-peer links. Table 1 shows the interference factors in an IEEE 802.11b network [14]. As seen from this table, we can know that interference range decreases as the channel separation increases. For instance, if the channel separation is grater than equal to 5, interference factor is zero, which means there is no interference between two nodes. Therefore, the MCM algorithm tries to find out the optimal channel number with minimum interference using the above objective function. If the best channel number has been selected, *u* uses this channel for its SI and *u*'s children also use this one for their RIs. This channel assignment process continues until it covers all relay nodes in the multicast tree using a breadth-first search method. An example of channel assignment by MCM is shown in Figure 1. In this example, *S* is a source node, *C*, *D*, and *F* are multi-receivers, and *A*, *B*, and *E* are relay nodes. The black-colored number means the channel information for each node's SI and the gray-colored one represents the channel number for each node's RI.

The MCM channel assignment algorithm considers only one-hop neighboring nodes to decide the influence of interference. This approach may yield the HCP. The HCP takes place when two different nodes which are away from each other with two-hop distance use the same channel, so these two nodes can interfere with themselves. For example, node *A* receives packets from node *S* on channel 1, which means node *A* is located in the source node's transmission range. Also, node *A* is within node *B*'s transmission range in Figure 1. Unfortunately, node *B* uses the same channel as the source node's SI. If node *S* and *B* transmit their packets at the same time, there will be a collision at node *A*. This situation occurs because node *B* examines only the channel assigned to its one-hop neighbor *A* and does not consider the one assigned to two-hop neighbor *S*. A similar problem exists among nodes *A*, *B*, and *E* on channel 6. Our new channel assignment algorithm thinks about every pair of nodes within the given network in order to minimize network interference and improve overall network throughput.

Table 1. Interference factors in an IEEE 802.11b network

Channel separation	2 Mbits/s	5.5 Mbits/s	11 Mbits/s
0	2.5	2.2	2.0
1	1.6	1.5	1.2
2	1.2	1.0	0.7
3	0.9	0.8	0.5
4	0.5	0.3	0.2
>= 5	0.0	0.0	0.0

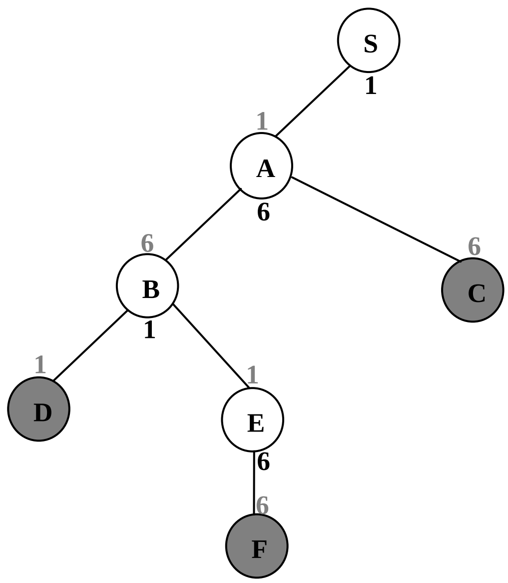

Figure 1. An example of channel assignment by MCM

2. 3 M4 Algorithm

Nguyen et al. [12] proposed M4 algorithm in order to solve HCP. M4 also does not use the interference factor in the optimization function. This algorithm eliminates HCP by adding to the optimization function the channel information of the two-hop neighbors of a node *v*. M4 obtains this information by allowing nodes to broadcast to their neighbors a message containing the channel they are using. Whenever every node receives the broadcast message, it adds its own channel information and re-broadcasts the updated message to its neighbors.

In M4, authors developed an optimization function which uses only channel numbers. Let $N^{\Diamond}(v)$ denote the set of one-hop and two-hop neighbors of node v that have already been assigned a channel, and c_v be the channel used by node v. They defined function $F(c)$ as follows:

$$F(c) = \frac{\prod_{\forall w \in N^{\Diamond}(v)} |c - c_w|}{\max_{\forall i \in N^{\Diamond}(v)} \{|c - c_i|\} \div \min_{\forall j \in N^{\Diamond}(v)} \{|c - c_j|\}}$$

For each multicast forwarding node v in the multicast tree including the source, the M4 algorithm assigns to v a channel c_v that maximizes the value $F(c_v)$. Node v chooses the channel that maximizes the channel separation from all of its one-hop and two-hop neighbors whose transmission channels have already been assigned using the above optimization function. Because of this feature, M4 is able to find the optimal channel with less interference than MCM.

The M4 algorithm has definitely further performance improvement compared to MCM. It considers not only one-hop neighbors but also two-hop ones for channel assignment to solve HCP. However, if the physical distance of one node and its two hop neighbors is very close, there may be network interference between two nodes. Therefore, M4 still has a network interference problem. Because of this reason, our proposed algorithm investigates every pair of nodes in the multicast tree to minimize interference between nodes.

CHAPTER 3. MINIMUM INTERFERENCE CHANNEL ASSIGNMENT

The ultimate goal of the minimum interference channel assignment (MICA) algorithm is to assign channels to wireless nodes with minimum network interference. Finally, it achieves maximum throughput for multicast in wireless mesh networks. The MICA accepts a multicast tree data structure as an input and produces the channel assignment for network interfaces of wireless nodes in that tree. In this chapter, we describe how the MICA algorithm operates.

There are three steps in the MICA. The first step is to compute channel separation of all pairs of nodes in the network in order to avoid network interference. The second one is that it figures out whether there exists the channel separation of zero among nodes because we can assign the same channel into those nodes as many as possible. Finally, the MICA assigns an optimal channel number to all nodes that have not been assigned channels for their SIs with minimum interference.

3. 1 Step 1: Calculating channel separation of all pairs of nodes

The MICA algorithm first calculates the channel separation of all pairs of nodes in the multicast tree. In this step, this algorithm considers only a pair of nodes which are not a leaf node in the tree. Channel separation is defined as the difference in the channel numbers used by the two pairs [21]. When computing the channel separation of two pairs, the MICA examines every pair of nodes in the network to minimize network interference between two nodes. Let me further explain the first step shown in Figure 2.

Input: T: a multicast tree
Output: The channel separation of all pairs of nodes

$CS<i, j>$: the channel separation between i and j

For each pair of nodes $(u, v) \in T$ ***do***
    C_u = the set of u's children
    C_v = the set of v's children
    $MAX_u = 0$
    $MAX_v = 0$

    For each node $i_u \in C_u$ ***do***
        $CS<v, i_u> \leftarrow$ the channel separation between v and i_u
        $MAX_v \leftarrow$ the maximum value between MAX_v and $CS<v, i_u>$
    End

    For each node $i_v \in C_v$ ***do***
        $CS<u, i_v> \leftarrow$ channel separation between u and i_v
        $MAX_u \leftarrow$ the maximum value between MAX_u and $CS<u, i_v>$
    End

    $CS<u, v> \leftarrow$ the maximum value between MAX_u and MAX_v
End

Figure 2. Calculating channel separation of all pairs of nodes

The purpose of the first step is to compute the channel separation of all pairs of nodes in the multicast tree. First, the MICA considers all pairs of nodes (u, v) in the tree. Next, it tries to get the channel separation between u and v's children and vice versa. To calculate the channel separation of two nodes, our algorithm uses interference factors.

As we mentioned before, interference factor is defined as the ratio of the interference range by the transmission range. From this definition, we can obtain the interference range by multiplying an interference factor by the transmission range. If the physical distance of two comparing nodes is within this interference range, those nodes may interfere with each other.

However, we can eliminate network interference between two nodes by increasing channel separation. According to table 1, as channel separation increases, an interference factor decreases. For example, if channel separation is grater than or equal to 5, there is no interference between two nodes in 802.11b network. Consequently, we can get channel separation with no interference between two nodes if we acquire channel separation in which the physical distance of two nodes is grater than the interference range of those. Finally, we decide the final channel separation between *u* and *v* by picking up the maximum value among channel separations between *u* and *v*'s children and between *v* and *u*'s children.

The example of a multicast tree is shown in Figure 3. We assume that all wireless nodes have the same transmission range 250m in this multicast tree. As we mentioned before, we do not consider leaf nodes in this step. So we think about the combinations of all nodes except *F*, *G*, and *I* in this graph. For example, we want to calculate the channel separation of *S* and *A*. The MICA always looks at the opponent's children to totally eliminate network interference between two nodes. In the case of *S* and *A* pair, *S*'s children are *A* and *B* and *A*'s child is *C*. Accordingly, *S*-*C* and *A*-*B* pairs will be considered to calculate the channel separation of *S* and *A*.

Let us assume that the physical distance of *S* and *C* is 350m and the physical one of *A* and *B* is 420m. If the channel separation of two nodes is 1 then an interference factor is 1.6 at 2 Mbit/s. Now, we can calculate the interference range by multiplying 1.6 by 250m. So the interference range between two nodes is 400m. This means that if the distance of two nodes is within this amount of value, they can interfere with each other. Consequently, we have to increase the channel separation of two nodes to avoid network interference. Therefore, the channel separation of *S* and *C* is 2 and the channel separation of *A* and *B* is 1 to remove network interference. Among these two values, the MICA finally picks up the maximum as the channel separation of *S* and *A*, so the final channel separation of *S* and *A* is 2. This procedure is applied to calculate the channel separation of all pairs of nodes except *F*, *G*, and *I* in the multicast tree shown in Figure 3.

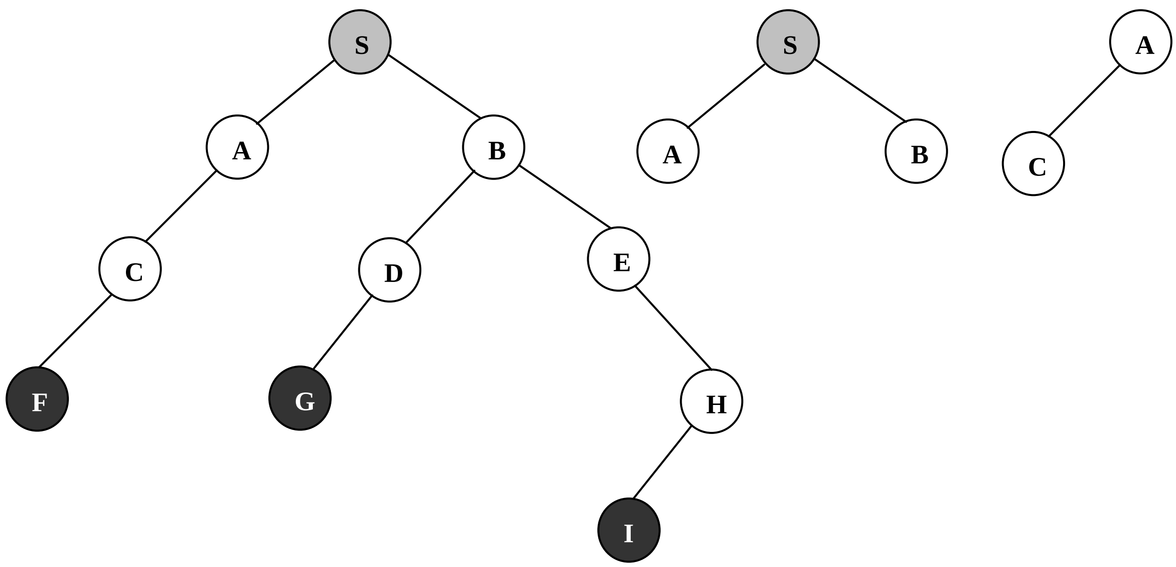

Figure 3. Multicast tree and children of *S* and *A*

3. 2 Step 2: Finding channel separation of zero

There are two situations in the second step. One is that there exists the channel separation of zero between two nodes and the other is that there is no pair of nodes with the channel separation of zero. In this step, we first find the channel separation of zero. If there is no channel separation of zero then we try to discover the channel separation of a maximum value. Let us further talk about this procedure represented in Figure 4.

The channel separation of zero means that there is no difference in the channel numbers between two nodes. In other words, although two nodes use the same channel number at the same time, there is no interference between these nodes. It is very important to find channel separation of zero because available channel numbers we can use are not sufficient.

The second step takes a multicast tree and the channel separation obtained from the first step as an input. We define two sets of nodes, S_D and S_N in this phase. We use S_D to put nodes that have determined the channel number for their SIs and S_N to store nodes that have not determined the channels. These sets are the result of the second step. For each pair of nodes (u, v) in the multicast tree, the MICA investigates whether there is the channel separation of zero. If it exists, we assign channel number 6 to the SI of u and v and put two nodes to S_D and the rest of nodes except u and v to S_N. Next, if S_D is not empty, we keep searching the channel separation of zero between a node x in S_N and every y in S_D. Whenever getting the channel separation of zero, we remove x from S_N and put it to S_D. This work continues until there is no channel separation of zero. Otherwise, for every (u, v), we just pick up the channel separation of a maximum value between u and v. After that, we give the channel number 6 to one of these nodes and the summation of channel 6 and the maximum channel separation value to the other node as its channel number. Finally, we put u and v to S_D and all nodes except u and v to S_N.

We think about the following example to illustrate this process. Let us assume that there exists the channel separation of zero between node A and D in Figure 3. We can assign the same channel to A and D. In this case, the MICA assigns the channel number 6 to these nodes for their SIs and put two nodes into S_D. After that, it keeps searching a pair of nodes with the channel separation of zero. We already have two nodes assigned the same channel in S_D. So we have to compare all other nodes which are in S_N with two nodes (A and D) in S_D. For example, we check the channel separation of one pair of A and H and the other pair of D and H. If the channel separation of two pairs is all zero, we can give node H the same channel assigned A and D and put it into S_D. As a result, $S_D = \{A, D, H\}$ and $S_N = \{S, B, C, E\}$. If there is no further node with the channel separation of zero compared to all nodes in S_D, we put those nodes into S_N.

The following example is the other situation in the second step. There is no pair of nodes with the channel separation of zero in the multicast tree. If this situation happens, the MICA just picks up two nodes with the channel separation of a maximum value. Suppose that the channel separation of A and H has the maximum value 5. We can assign the channel 6 to A and the channel 11 to H and put these two nodes into S_D. Finally, all remaining nodes - S, B, C, D, and E - are located in S_N. Therefore, $S_D = \{A, H\}$ and $S_N = \{S, B, C, D, E\}$.

Input:
T: a multicast tree
$CS<u, v>$: channel separation of $(u, v) \in T$
Output:
S_D: the set of nodes that have been assigned channels for their SIs
S_N: the set of nodes that have not been assigned channels for their SIs

For each pair of nodes $(u, v) \in T$ ***do***
If $CS<u, v> = 0$ ***then***
The sending interface of $u \leftarrow$ channel 6
The sending interface of $v \leftarrow$ channel 6
$S_D \leftarrow u, v$
$S_N \leftarrow$ all nodes except u and v in T
Stop searching
End
End

If S_D is not empty ***then***
For each node $x \in S_N$ ***do***
If $CS<x, y> = 0$ for all $y \in S_D$ ***then***
The sending interface of $x \leftarrow$ channel 6
Remove x from S_N
$S_D \leftarrow x$
End
End
Else
Find $u, v \in T$ such that $CS<u, v>$ is a maximum
The sending interface of $u \leftarrow$ channel 6
The sending interface of $v \leftarrow$ channel 6 + $CS<u, v>$
$S_D \leftarrow u, v$
$S_N \leftarrow$ all nodes except u and v in T
End

Figure 4. Finding channel separation of zero

3. 3 Step 3: Assigning channels to all nodes

In the final round, we have to assign the channel number to all nodes in S_N based on the channel separation. Let us look at Figure 5 to explain the final step in detail.

The MICA takes S_D and S_N as an input and produces channel assignment for all network interfaces of every node in the multicast tree. First, the MICA compares the channel separation of each node x in S_N with all nodes in S_D. After all comparison, it selects one node which has the channel separation of a maximum value. Next, for each node k in S_D, we can generate several conditions for the channel of node x. We already have the channel separation of (k, x) and the channel number of k, so a possible channel number of x can be greater than or equal to the summation of the channel separation of (k, x) and the channel number of k or less than or equal to the difference between the channel separation of (k, x) and the channel number of k. According to the number of elements of set S_D, the number of conditions will be decided. After producing all possible conditions, the MICA picks up a channel number for node x that satisfies all conditions. Eventually, that channel number will be used for the SI of node x. Now we remove x from set S_N and put it to S_D. This process keeps going until there is no element in S_N, which means the SI of all nodes in the multicast tree has its own channel number. Finally, we can easily assign the channel to each node's RI because each node's RI should be the same channel as its parent's SI to communicate each other.

The following scenario is a specific example to demonstrate this procedure. In Figure 3, assume that $S_D = \{A, H\}$, $S_N = \{S, B, C, D, E\}$, and node A and D have the channel separation of 3 and H and D have 2. This value is the maximum channel separation among all nodes in S_N. Therefore, we conclude the channel separation between *A* and *D* has the maximum value 3. Node *D* can have the channel number based on these channel separations. Let us assume that the channel of *A* is 6 and the channel difference between *A* and *D* is 3, so there is the following condition: the channel of *D* <= 3 or the channel of *D* >= 9, so it can have the channel number 3 or 9. Also, suppose that the channel of *H* is 11 and the channel separation between *H* and *D* is 2, therefore, there exists the possible condition: the channel of *D* <= 9 or the channel of *D* >= 13. However we cannot use channel 13 because our assumption is that we use 11 channels from 1 to 11. Channel 13 is an invalid number, so we have to eliminate this channel number. As a result, the possible channel of *D* is 9 in this case. From these three conditions, we assign the channel number 9 to node *D* and put it into S_D to satisfy these conditions. We cannot assign channel 3 to *D* because one of the possible conditions is that the channel of *D* should be grater than of equal to 9. This process continues until it covers all nodes in S_N. Finally, all nodes except *F*, *G*, and *I* in the multicast tree are placed in S_D. This means every node except leaf nodes has its own channel number for its SI.

Figure 6 represents the multicast tree where every node has its own channel number for its SI and RI. Leaf nodes *F*, *G*, and *I* has only a receiving interface because they do not need to forward packets and the source node *S* has simply a sending interface because it always sends packets to its children.

Input:

S_D: the set of nodes that have been assigned channels for their SIs
S_N: the set of nodes that have not been assigned channels for their SIs
T: a multicast tree

Output: Channel assignment for all network interfaces

CS<u, v>: channel separation between *u* and *v*
CH_i: channel number for SI of node *i*
$COND_j$: the set of conditions in which there is the range of CH_j

While S_N is not empty ***do***
Find $x \in S_N$, $y \in S_D$ such that *CS<x, y>* is a maximum
For each node $k \in S_D$ ***do***
$COND_x \leftarrow [CH_x >= CH_k + CS<x, k>$ or $CH_x <= CH_k - CS<x, k>]$
Eliminate all invalid channel numbers ($CH_x < 1$ or $CH_x > 11$) and conflict conditions
Pick up CH_x that satisfies all conditions $\in COND_x$
End
The sending interface of $x \leftarrow CH_x$
Remove *x* from S_N
$S_D \leftarrow x$
End

For each node $u \in T$ ***do***
The channel for RI of *u*'s children ← the channel for SI of *u*
End

Figure 5. Assigning channels to all nodes

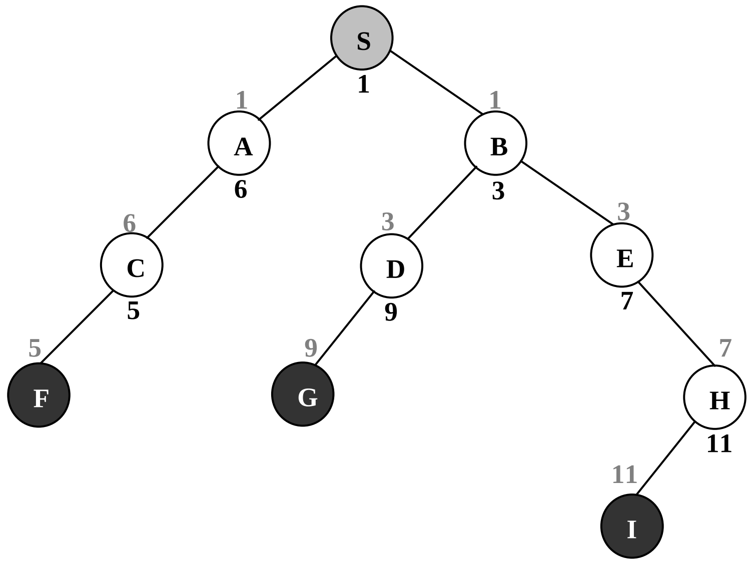

Figure 6. An example of channel assignment by MICA

CHAPTER 4. SIMULATIONS

We evaluate the MICA algorithm by comparing it with the MCM algorithm. Our simulation tool is the QualNet 4.5 [22]. First, we measure average packets received by multi-receivers and standard deviation of average packets in different network topologies. In this experiment, we fix the number of multi-receivers and use a totally different multicast tree whenever we perform each experiment. Next, we compute average packets received by multi-receivers and average end-to-end packet delay in different number of multi-receivers. In this simulation, we use the same network topology throughout the entire experiment. Whenever we carry out each simulation, we only change the number of multi-receivers.

4. 1 Performance metrics

We use the following metrics to measure the performance of the MICA and MCM algorithm.

- *Average packet*: average packet is defined as the average number of packets each multi-receiver receives successfully during a simulation time.
- *Average delay*: average delay is the average time taken for a packet to be transmitted across a network from source to destination.
- *Standard deviation*: the standard deviation is the variability or dispersion of packets received by all multi-receivers.

4. 2 Simulation Parameters

We first perform 10 experiments with different network topologies. Whenever conducting each experiment, we randomly place 30 different wireless nodes in a flat area of 900 m by 900 m. In this scenario, there are one source node and 10 multi-receivers in the multicast tree where these nodes are randomly selected. For the other experiment, we changed the number of multi-receivers with the same topology. There are one source and different number of receivers and 5 experiments in this simulation. In entire experiments, each node has two network interfaces for sending and receiving data packets, so they use two different channels for its radios. We generate 11 channels for channel assignment in QualNet environment. The transmission range of wireless nodes is 250m. We use PHY802.11b at the physical layer with a transmission rate of 11Mbits/s. The data packet size for all traffic is set to be 512 bytes and transmission rate at the source node is 100 packets/s. The traffic model we chose is the multicast constant bit rate (MCBR) traffic generator [23] to evaluate the multicast performance. MCBR operates identically to constant bit rate (CBR), but the destination must be a multicast address and a multicast routing protocol must be configured. The QualNet software provides the following multicast routing protocols: on-demand multicast routing protocol (ODMRP) [24], distance vector multicast routing protocol (DVMRP) [25], multicast open shortest path first (MOSPF) [26], and multicast ad hoc on-demand distance vector (MAODV) [27]. We decide to use MOSPF because this protocol basically supports multiple network interfaces. Finally, we set the total simulation time to 300 seconds. The above simulation parameters are summarized in Table 2.

Table 2. Simulation parameters

Parameter	Value
Number of channels used	11
Network size	30 nodes over 900 m × 900 m
Transmission range	250 m
Transmission rate at physical layer	11 Mbits/s
Physical layer protocol	PHY802.11b
Multicast routing protocol	MOSPF
Packet size	512 bytes
Transmission rate at application layer	100 packets/s
Traffic model	MCBR
Simulation time	300 seconds

4. 3 Simulation results

We use different network topologies when we perform the first scenario. Whenever executing each simulation, a source node generates 100 packets per second and the simulation time is 300 seconds, so a sender produces 30,000 packets during each simulation. For each experiment, we measure total packets received by each multi-receiver and compute the average value for these packets. This simulation results are represented in Figure 7. This graph shows average packets received by multi-receivers during simulation time. As seen from this graph, the performance of the MICA algorithm is much better than that of MCM. The average number of packets received by multi-receivers using MICA is approximately between 25,000 and 29,000. This value is very close to the number of packets the source generates. On the other hand, the majority of average packet number using MCM is below 20,000 and the worst case is under 5,000.

We also measure the standard deviation of all packets received by multi-receivers. The result is shown in Figure 8. This picture shows the standard deviation of the packets received by receivers. The standard deviation of MICA is much lower than that of MCM. For 10 experiments, the standard deviation of MICA has only below 2000, which means all receivers receive packets evenly. However, MCM has much higher value than MICA, which means some destinations receive some amount of packets but other nodes cannot receive anything at all in the worst case. The standard deviation in our simulation shows the variability of packets received by all multi-receivers. If the value of standard deviation is small, all multi-receivers fairly receive data packets, but a large amount of value means some receivers get enough packets and others did not receive anything at all in some cases. Consequently, the variance of packets received by multi-receives is extremely large, which indicates overall network throughput is not outstanding.

Finally, we evaluate average packets received in different number of multi-receivers by assigning the number of receivers with 2, 4, 6, 8, and 10. We measure the average number of packets using the MICA and MCM algorithm. The results are shown in Figure 9. This graph shows the trend of average packets received when the number of receivers is increased. Although the number of receivers is increased, the performance of MICA is very stable. All receivers receive almost all of packets from source regardless of the number of receivers. However, the performance of MCM is affected by the number of multi-receivers because the chance of network interference is also increased. We also compute the average end-to-end packet delay of the MICA and MCM algorithm by comparing the average time each packet takes to arrive at multi-receivers. Figure 10 shows the trend of average end-to-end packet delay. The average delay of MICA is almost same regardless of the number of receivers.

However, the delay of MCM is getting higher as the number of receivers is increased because of network interference. If network interference exists then it causes packet collision and retransmit, so the end-to-end delay is also increased.

The most important reason why we get these simulation results is that the MCM algorithm has the HCP which we discuss in chapter 2. We conclude that the HCP yields network interference among wireless nodes. As a result, it causes a poor network throughput. By eliminating this problem, the MICA algorithm significantly improves the network performance for multicasting.

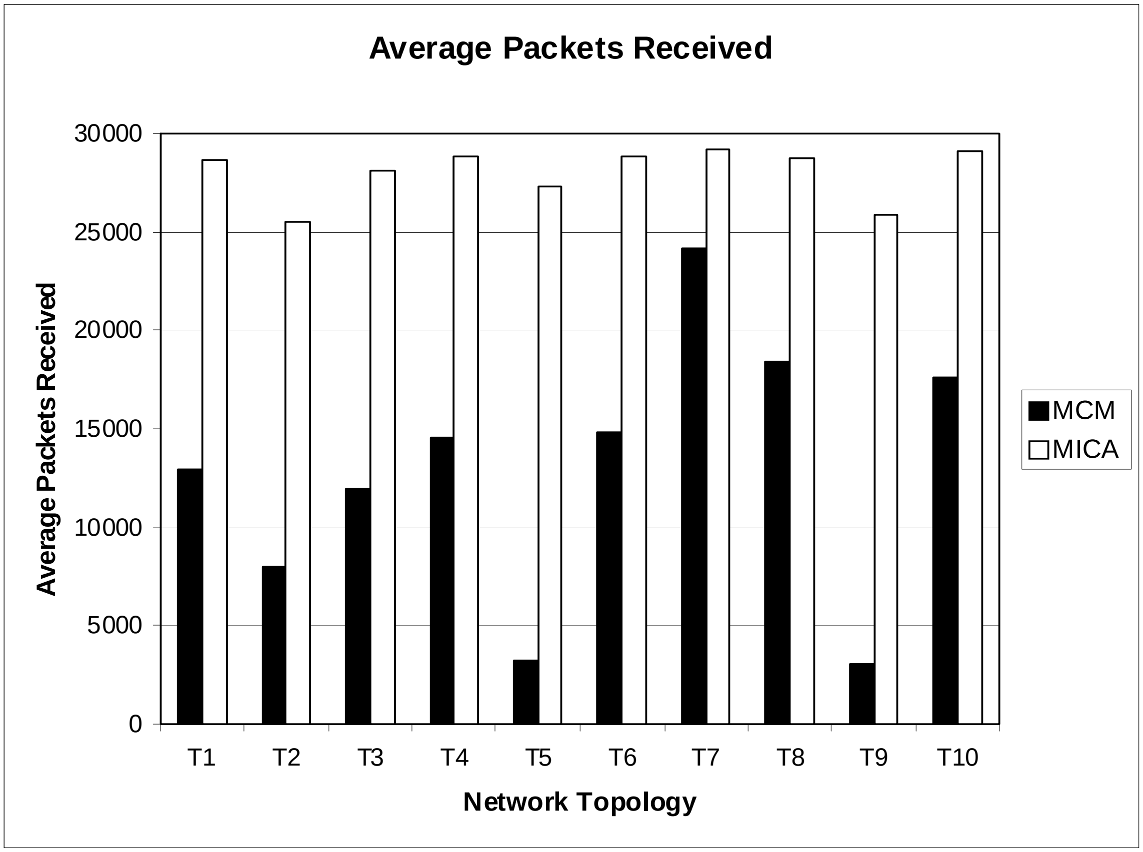

Figure 7. Average packets received by multi-receivers

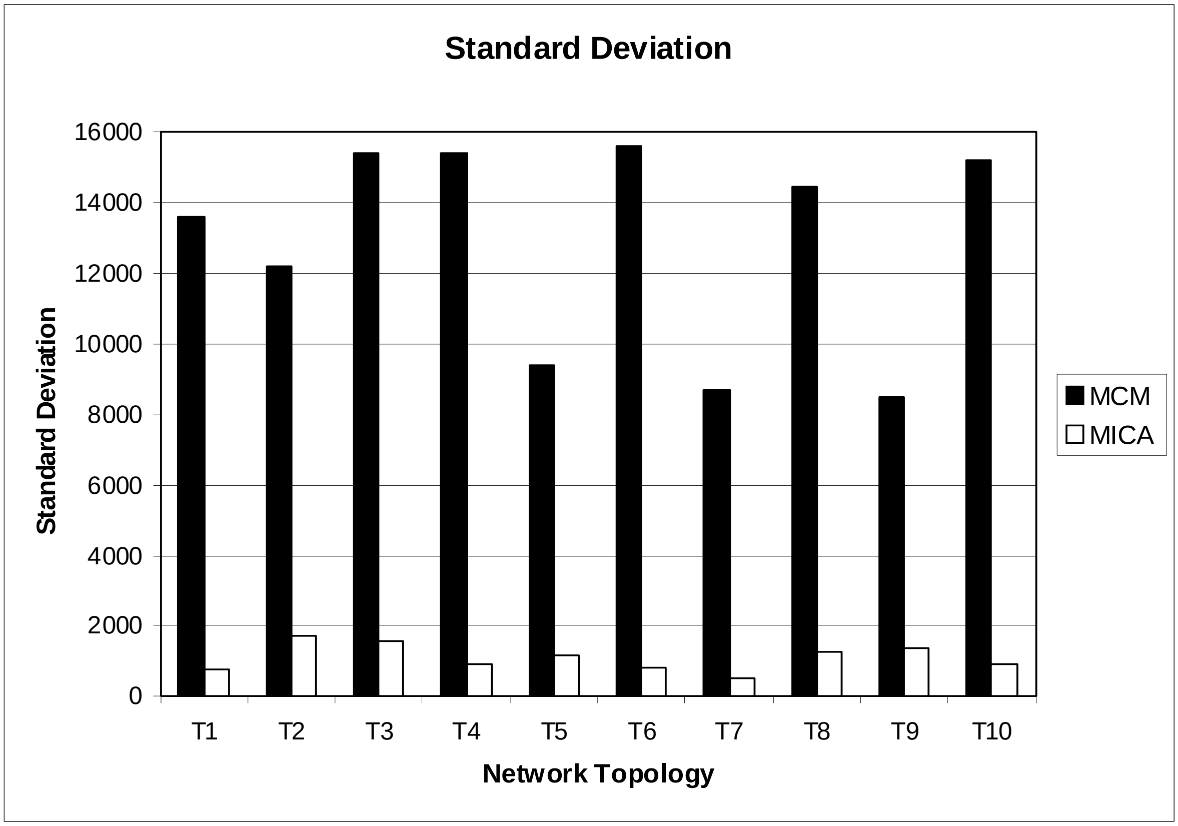

Figure 8. Standard deviation of all packets received by multi-receivers

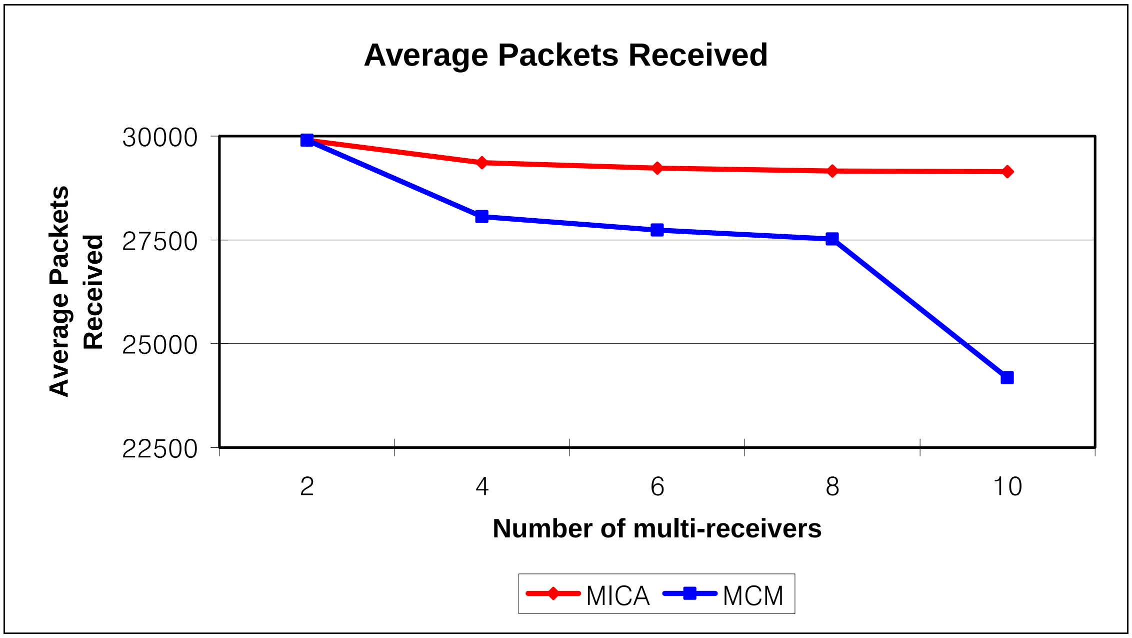

Figure 9. Impact of number of multi-receivers

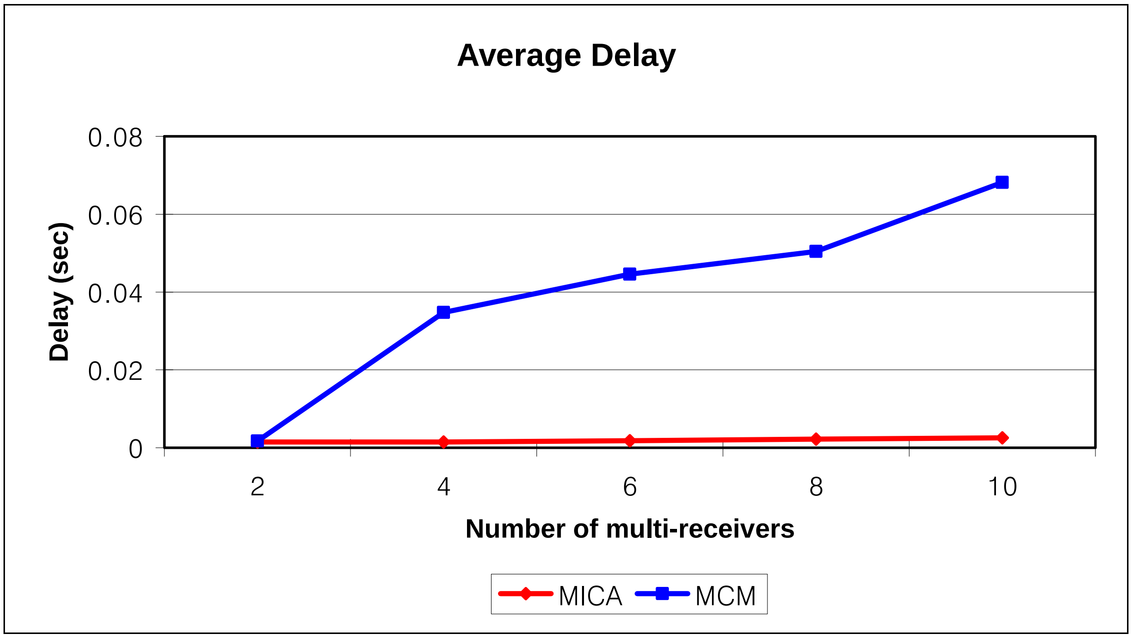

Figure 10. Delay comparison

CHAPTER 5. CONCLUSTIONS

In this work, we propose a channel assignment algorithm for multicast in multi-channel multi-radio wireless mesh networks. We investigate the drawback of the MCM algorithm and try to find the solution to minimize network interference and enhance network throughput. MCM only considers one-hop neighbors for channel assignment. This mechanism may yield the HCP, so there is network interference among wireless nodes when they communicate each other at the same time and this problem has influence on overall network throughput. Our simulation results show that the performance of MCM is much worse than that of MICA because of network interference. Accordingly, our approach focuses on reducing network interference by considering every pair of nodes in the multicast tree for channel assignment. By minimizing interference among wireless nodes in the wireless network, we can improve overall network throughput and reduce end-to-end packet delay. The performance evaluation shows that our algorithm outperforms the MCM algorithm in terms of average packets received by multi-receivers and average delay in the given network environment.

BIBLIOGRAPHY

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Elsevier Comp. Networks,* vol. 47, no. 4, pp. 445-87, March 2005.

[2] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of ACM SIGMOBILE'05*, 2005.

[3] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, "Optimization models for fixed channel assignment in wireless mesh networks with multiple radios," in *Proceedings of IEEE SECON'05*, 2005.

[4] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom'05)*, 2005.

[5] M. K. Marina and S. R. Das, "A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks," in *Proceedings of IEEE International Conference on Broadband Networks (Broadnets'05)*, 2005.

[6] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proceedings of Infocom 2006* Barcelona, Spain, 2006.

[7] B. Raman, "Channel allocation in 802.11-based mesh networks," in *Proceedings of IEEE INFOCOM'06* Barcelona, Spain, 2006.

[8] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," in *Proceedings of IEEE InfoCom'05*, 2005.

[9] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," in *Proceedings of ACM SIGMOBILE'04*, 2004.

[10] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum interference channel assignment in multi-radio wireless mesh networks," in *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'07)*, 2007.

[11] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proceedings of ACM SIGMOBILE'05*, 2005.

[12] H. L. Nguyen and U. T. Nguyen, "Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks," in *Proceedings of IEEE International Wireless Communications and Mobile Computing Conference (IWCMC'08)*, 2008.

[13] Z. Yin, Z. Li, and M. Chen, "A Novel Channel Assignment Algorithm for Multicast in Multi-radio Wireless Mesh Networks," in *Proceedings of IEEE International Conference on Network Protocols (ICNP'07)*, 2007.

[14] G. Zeng, B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Multicast Algorithms for Multi-channel Wireless Mesh Networks," in *Proceedings of 15th IEEE International Conference on Network Protocols (ICNP 2007)* Beijing, China, 2007.

[15] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating Optimal Multicast Trees in Wireless Multihop Networks," in *Proceedings of IEEE ISCC'05*, 2005.

[16] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *IEEE Journal on Selected Areas in Communications 2006,* pp. 2092-2103, 2006.

[17] S. Roy, D. Koutsonikolas, S. Das, and Y. C. Hu, "High-throughput multicast routing metrics in wireless mesh networks," in *Proceedings of IEEE ICDCS'06*, 2006.

[18] U. T. Nguyen and J. Xu, "Multicast routing in wireless mesh networks: minimum cost trees or shortest path trees?," in *IEEE Communications Magazine*, 2007, pp. 72-77.

[19] X. Zhao, C. T. Chou, J. Guo, and S. Jha, "Protecting multicast sessions in wireless mesh networks," in *Proceedings of IEEE Conference on Local Computer Networks (LCN'06)*, 2006.

[20] http://mathworld.wolfram.com/VertexColoring.html, "Vertex Coloring."

[21] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: turning a peril into an advantage," in *IMC'05: Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference*, 2005, pp. 29-29.

[22] http://www.qualnet.com.

[23] http://www.scalable-networks.com/distributions/documentation/4.5/QualNet-4.5-Developer-ModelLibrary.pdf.

[24] S. J. Lee, M. Gerla, and C. C. Chiang, "On-Demand Multicast Routing Protocol," in *Proceedings of IEEE WCNC'99* New Orleans, LA, 1999.

[25] http://www.networksorcery.com/enp/rfc/rfc1075.txt, "Distance Vector Multicast Routing Protocol," 1988.

[26] http://www.networksorcery.com/enp/rfc/rfc1584.txt, "Multicast Extensions to OSPF," 1994.

[27] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," in *Proceedings of MobiCom '99* Seattle, WA, 1999.