WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2009; **9**:557–571
Published online 29 October 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.701

Channel assignment for multicast in multi-channel multi-radio wireless mesh networks

Hoang Lan Nguyen*,† and Uyen Trang Nguyen
Department of Computer Science and Engineering, York University, Toronto, Canada M3J 1P3

Summary

One of the most effective approaches to enhance the throughput capacity of wireless mesh networks (WMN) is to use systems with multiple channels and multiple radios per node. Multi-channel multi-radio (MCMR) networks require efficient *channel assignment* (CA) algorithms to determine which channel a link should use for data transmission in order to maximize network throughput. The problem of CA has been studied extensively for unicast communications, but addressed only recently for multicast. We propose a CA algorithm named Minimum interference Multi-channel Multi-radio Multicast (M4) that minimizes interference among nodes in a multicast routing tree and uses both orthogonal and overlapping channels such as those in IEEE 802.11b/g systems. Simulation results show that M4 outperforms the Multi Channel Multicast algorithm proposed. in various scenarios with respect to average packet delivery ratio, throughput and end-to-end delay. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: wireless mesh networks; multicast; channel assignment; interference minimization; multi-channel multi-radio systems

1. Introduction

Wireless mesh networking is an emerging technology that supports many important applications such as Internet access provisioning in rural areas, ad hoc networking for emergency and disaster recovery, security surveillance, and information services in public transportation systems. The technology enables networking capability where wiring or installing cables is difficult or expensive. In a wireless mesh network (WMN), wireless routers provide multi-hop wireless connectivity from a host to either other hosts in the same network or in the Internet. The wireless routers are often static and form a wireless mesh backbone. Our work in this paper focuses on this mesh backbone, and we will use the terms ‘routers’ and ‘nodes’ interchangeably.

Wireless mesh networking can be implemented using IEEE 802.11, IEEE 802.15, or more recently, IEEE 802.16 technologies. In this paper, we consider IEEE 802.11 with carrier sense multiple access with collision avoidance (CSMA/CA) medium access control (MAC) because this is currently the most commonly used radio technique for WMNs.

Until recently, research on wireless ad hoc networks considers mostly networks with a single channel. The theoretical upper limit of per node throughput capacity in such networks is limited by $O(1/\sqrt{n})$, where n is the number of nodes in the network [1]. The theoretical *achievable* throughput is even

*Correspondence to: Hoang Lan Nguyen, Department of Computer Science and Engineering, York University, Toronto, Canada M3J 1P3.
†E-mail: lan@cse.yorku.ca

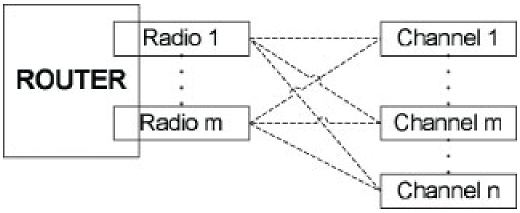

Fig. 1. Multi-radio multi-channel model.

lower, estimated as $\theta(1/\sqrt{n \log n})$ in a random ad hoc network with ideal global scheduling and routing [1]. It has also been shown through experiments that on a string topology using CSMA/CA MAC such as IEEE 802.11, the throughput degrades approximately to $1/n$ of the raw channel bandwidth [2]. The above results indicate that the throughput capacity of a single-channel WMN becomes unacceptably low as the network size increases.

Several factors contribute to such a rapid degradation of throughput such as the behavior of the MAC protocols, greediness of the initial nodes, and subsequent flow starvation of the latter hops. However, the single most important factor is the exposed terminal problem, worsened by the use of a single-radio single-channel network. One of the most effective approaches to enhance the aggregate network throughput is to use systems with multiple channels and multiple radios per node [3–7].

Figure 1 illustrates the multi-channel multi-radio (MCMR) model. The network has n channels, which may either overlap, such that a channel partially shares its spectrum with the adjacent channels, or may be completely separated (orthogonal). For example, IEEE 802.11b/g networks have 11 channels, numbered from 1 to 11. Orthogonal channels are separated by at least four other channels; for instance, channels 2 and 7 are orthogonal. A host in a MCMR network has m radios (interfaces), and typically $1 < m < n$ (e.g., $m = 3$, $n = 11$). A MCMR node can transmit on one channel and receive on another at the same time using two different radios. As a result, a MCMR wireless network *at least* doubles the throughput, since each node is now in full-duplex mode, being able to transmit and receive simultaneously. In return, MCMR networks require efficient algorithms for *channel assignment* (CA) [4–6,8,9], the task of determining which channel a link should use for data transmission in order to minimize interference for maximum throughput.

Multicast is a form of communication that delivers information from a source to a set of destinations simultaneously in an efficient manner. Important applications of multicast include distribution of financial data, billing records, software, and newspapers; audio/video conferencing; distance education; IP television; and distributed interactive games. Although multicast is required to support many important applications, research on multicasting in MCMR WMNs is still in its infancy.

The problem of CA has been studied extensively in the context of unicast communications [4–6,8,9], and most assumes orthogonal channels [10]. CA for multicast, however, has only been addressed recently [11,12]. Zeng *et al.* [11] proposed a CA algorithm for multicast in MCMR WMNs called Multi-Channel Multicast MCM. This algorithm suffers from low performance caused by the *hidden channel problem* (HCP), and from the inconvenient use of interference factors. In this paper, we propose a CA algorithm named Minimum interference Multi-channel Multi-radio Multicast (M4) that eliminates the HCP and the use of interference factors. The algorithm enables the nodes in a multicast tree to operate with minimum interference. Like MCM, we consider both orthogonal and overlapping channels such as those in IEEE 802.11b/g systems. Our experimental results show that M4 outperforms MCM in various scenarios with respect to average packet delivery ratio (PDR), throughput and end-to-end delay.

The remainder of the paper is organized as follows. We briefly describe the MCM algorithm and analyze its drawbacks in Section 3. We present our proposed CA algorithm and its performance evaluation in Sections 4 and 5, respectively. Related work is discussed in Section 6, and Section 7 concludes the paper.

2. Definitions and Assumptions

We consider WMNs with stationary wireless routers. Two nodes are directly connected if they are within the radio range of each other and referred to as *one-hop neighbors*. Two nodes that communicate with each other via *an* intermediate node are called *two-hop neighbors*.

The HCP occurs when two nodes that are two-hops away from each other select the same channel and thus interfere with each other's transmission. For example, in Figure 2(b), if S and E select the same channel, say, channel 1, and transmit at the same time, their signals will collide at node C. If C is the intended recipient of either transmitter, C will not receive the correct packet.

DOI: 10.1002/wcm

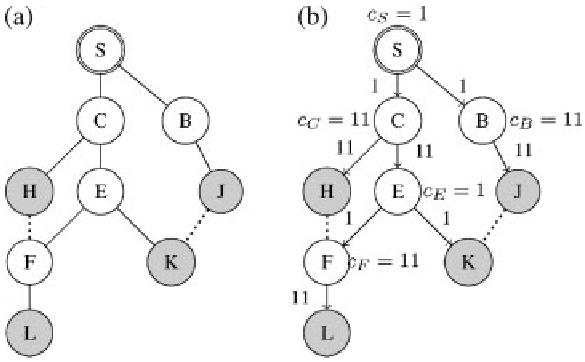

Fig. 2. An example of channel assignment using MCM. Dotted lines are not part of the multicast tree but shown to represent direct connectivity between nodes. (a) A multicast tree. (b) MCM channel assignment.

We assume that nodes that are three or more hops away from each other do not interfere. In WMNs (unlike MANETs or many types of sensor networks), we have control over the placement of routers in the network.

Our proposed CA algorithm considers both overlapping and orthogonal channels. Our experiments are based on IEEE 802.11b/g standards [13] in which there is a total of 11 channels numbered from 1 to 11. The channel separation between two channels c_1 and c_2 is defined as $|c_2 - c_1|$. For instance, the channel separation between channel 1 and channel 5 is four. In IEEE 802.11b/g standards, the separation between orthogonal channels is at least five (i.e., separated by at least four other channels, e.g., channels 3 and 8).

To measure the level of interference between neighboring nodes, the MCM algorithm uses a metric named *interference factors*. The interference factor is defined as the ratio of the interference range over the transmission range. Zeng *et al.* [11] described a method for measuring interference factors. Using four wireless routers equipped with Netgear WAG511 PC cards to establish two wireless links (one between each pair of routers), the authors moved the two wireless links far away from each other gradually until they did not interfere with each other. This gave the interference range of the two links in order to calculate the interference factor as defined above. Table I lists the interference factors versus channel separation for 2 Mbits/s, 5.5 Mbits/s, and 11 Mbits/s data rates obtained from this experiment [11]. The results show that the larger the channel separation, the less the interference and the lower the interference factor. When the channel separation is five or more, the interference factors approach zero.

Table I. An example of interference factors in an IEEE 802.11b network [11].

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

By definition, interference factors depend on the transmission rate at the physical layer, channel separation, distances between nodes and environmental factors such as signal reflections and multi-path fading.

We assume that a multicast tree has been constructed before the CA algorithm is applied, as in MCM. The goal of the CA algorithm is to minimize the interference between nodes in the given tree. There exist several approaches/algorithms for building multicast trees [14], such as shortest path trees [15–22], minimum Steiner tree [23–27], minimum data overhead tree [28]. A performance comparison of these types of trees for single-channel WMNs can be found in [29,30].

The parent-child and sibling relationships between nodes in a multicast tree is the same as those defined for the traditional rooted tree data structure [31], where the root of the tree is the multicast source.

When applying the M4 algorithm, each node in the multicast tree uses only two radios (interfaces): one for receiving multicast data from its parent (uplink interface), and the other for sending multicast data to its children (downlink interface). Other remaining radios, if any, can be used for other flows. Note that the multicast source has no uplink interface in the tree, and multicast destinations have no downlink interface.

We assume that the network topology as well as multicast membership are static. In practice, routers may be added to, removed from or moved inside the network; members may join or leave the multicast group freely at will. These events require the reconstruction of the multicast tree and re-computation of the CA. These issues are to be addressed in our future work.

Finally, multicast data flows are assumed to be one-way, from the source towards the destinations (although the routing protocol and the CA algorithm may require children to sent control messages to their parents. However, this is done before data transmission starts and does not interfere with data packets.)

DOI: 10.1002/wcm

3. Multi-Channel Multicast (MCM)

In this section, we briefly describe the MCM algorithm and then analyze its drawbacks.

3.1. The MCM Algorithm

The MCM algorithm considers both overlapping and orthogonal channels. It uses interference factors to minimize interference among one-hop neighboring nodes. The MCM CA algorithm works as follows. It starts with the source node by assigning a channel to the downlink interface of the source. All multicast children of the source node then tune into this channel for receiving multicast data from the source. The algorithm then processes the source's children, following a breadth-first search traversal [32] of the multicast tree. For each child, MCM assigns a channel to the downlink interface of the node so that the assignment minimizes the interference factor between this node and all of its one-hop neighbors who have already been assigned a channel. Specifically, let $N(v)$ denote the set of one-hop neighbors of node v that have already been assigned a channel; c_v is the channel that is assigned to node v; $\delta_{(c_v,c_w)}$ is the interference factor between two channels c_v and c_w. For each forwarding node v in the multicast tree (including the source), MCM selects a channel c_v for v so that it minimizes the following function:

$$\sum_{\forall w \in N(v)} \delta^2{}_{(c_v,c_w)} \quad (1)$$

If there is more than one channel that satisfies the optimization function, MCM will randomly choose one channel from the multiple solutions. The CA procedure repeats until it covers all forwarding nodes of the routing tree in the order of a breadth-first search traversal. An example of the CA produced by the MCM algorithm is shown in Figure 2. In this example, S is the multicast source; H, J, K, L are multicast destinations; and C, B, E, F are multicast forwarding nodes.

3.2. Drawbacks of MCM

The CA algorithm of MCM suffers from the HCP, and the use of interference factors is not convenient and not flexible. Moreover, random selection of a channel from multiple choices may not give the best solution. In this section, we discuss these drawbacks of the MCM algorithm.

(1) *The hidden channel problem.* When computing the optimization function (1), MCM considers only the interference caused by *one-hop* neighbors of a node v. For example, in Figure 2(a), if $v = E$ then the set of neighbors used in function (1) is $N(v) = \{C\}$. This, however, causes the HCP, as illustrated in Figure 2(b). In this example, node C receives data from node S on channel 1 and is within the transmission range of node E, who transmits on channel 1 as well. If two signals transmitted by S and E arrive at C at the same time, they will collide at node C. The reason is that node E considers only the channel assigned to node C, its one-hop neighbor, and ignores the two-hop neighbor S. A similar scenario happens between nodes C, H, and F on channel 11. Our proposed algorithm improves the MCM algorithm by including a two-hop interference component in the optimization function, making it aware of this HCP.

(2) *Interference factors.* As discussed earlier, interference factors depend on the transmission rates at the physical layer, distances between nodes and physical properties of the operating area. Therefore, before applying the MCM algorithm, one needs to acquire the interference factor values of a given network and supported data rates. In addition, the interference factors obtained in a network area may not be applicable to others as the interference characteristics may not be the same. Moreover, varying interference factors (due to varying environmental conditions) are likely to generate fluctuating CA solutions that are not optimal. Therefore, the M4 algorithm uses channel numbers (in combination with its own optimization function) instead of interference factors.

(3) *Random selection from multiple choices.* Assuming IEEE 802.11b/g, when the channel separation is equal or greater than five channels, the interference factor approaches zero [11]. This means that there can exist multiple channels that satisfy the objective function. Consider the example illustrated by Figure 3(a), in which there are three nodes that are within each other's transmission

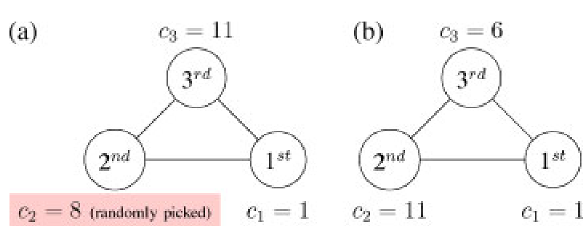

Fig. 3. Channel assignment comparison between MCM and M4. (a) MCM: non-optimal channel assignment. (b) M4: optimal channel assignment.

DOI: 10.1002/wcm

range. Since the first node has selected channel 1, the possible solutions for the second node are channels 6, 7, 8, 9, 10 or 11. In this example, the second node chooses (randomly) channel 8.

Randomly selecting one of the possible channels, as MCM does, may not give the best performance. Using the above example and optimization function (1), we see that the channel to be chosen for the third node is 11. The solution is thus $\{1, 8, 11\}$, resulting in one pair of overlapping channels (8 and 11). On the other hand, the optimal solution is $\{1, 6, 11\}$, which gives us three orthogonal channels and no overlapping channels (Figure 3(b)). The optimization function in our proposed M4 algorithm supports optimal channel selection when there are multiple choices.

4. The M4 Channel Assignment Algorithm

Considering the drawbacks of MCM, we propose a CA algorithm for multicast called M4 that solves the HCP and does not use interference factors in the optimization function.

4.1. Algorithm

We eliminate the HCP by adding to our optimization function the channel information of the *two-hop* neighbors of a node v. In the example of multicast tree in Figure 2, node E is informed of the channel already selected by node S so that it can avoid that channel. Similarly, F should know about the channel information of node C. How to collect channel information of two-hop neighbors is discussed in Section 4.2.

We solve the interference factor problem by developing an optimization function which uses only channel numbers (e.g, 1, 2, ..., 11 in IEEE 802.11b) while maximizing the channel separation among one-hop and two-hop transmitting neighbors.

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. We define optimization function $F(c)$ as follows:

$$F(c) = \frac{\prod_{\forall w \in N^{\diamond}(v)} |c - c_w|}{\max_{\forall i \in N^{\diamond}(v)} \{|c - c_i|\} \div \min_{\forall j \in N^{\diamond}(v)} \{|c - c_j|\}} \quad (2)$$

Then 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 $F(c_v)$.

The right hand side of Equation (2) is a fraction whose numerator is the product of the absolute values of the channel separations between v and w, $w \in N^{\diamond}(v)$. The goal is to maximize the channel separation between v and all its neighbors.

To obtain the denominator, M4 finds the maximum and minimum among all channel separations of the currently available (c_v, c_w) neighbor pairs, and then divides the maximum by the minimum. The objective here is to balance the channel separation among all channel pairs considered. This helps avoid situations such as the example in Figure 3(a) where one channel pair is 'over-separated' (channels 1 and 8), while the other pair is overlapping (8 and 11). M4 offers the optimal solution, which is three orthogonal channel pairs from the set $\{1, 6, 11\}$ (Figure 3(b)).

Thanks to the above optimizations, M4 is able to find CAs with less interference than MCM, which resorts to random selections when there are several channel choices (compare the CAs in Figure s 3(a) and 3(b) for an example).

To further demonstrate the improvement of M4 over MCM, we use M4 to assign channels to nodes in the multicast tree shown in Figure 2(a). The resulting CAs, after applying Equation (2) to all multicast nodes in the multicast tree, are shown in Figure 4. Compared to the

Channel	C, $N^{\diamond}(C) = \{S\}$	B, $N^{\diamond}(B)=\{S,C\}$	E, $N^{\diamond}(E)=\{S,C\}$	F, $N^{\diamond}(F)=\{C,E\}$
c=1	$F(c) = 0$	$F(c) = 0$	$F(c) = 0$	$F(c) = 25$
c=2	$F(c) = 1$	$F(c) = 1$	$F(c) = 1$	$F(c) = 16$
c=3	$F(c) = 2$	$F(c) = 4$	$F(c) = 4$	$F(c) = 9$
c=4	$F(c) = 3$	$F(c) = 9$	$F(c) = 9$	$F(c) = 4$
c=5	$F(c) = 4$	$F(c) = 16$	$F(c) = 16$	$F(c) = 1$
c=6	$F(c) = 5$	$F(c) = 25$	$F(c) = 25$	$F(c) = 0$
c=7	$F(c) = 6$	$F(c) = 16$	$F(c) = 16$	$F(c) = 1$
c=8	$F(c) = 7$	$F(c) = 9$	$F(c) = 9$	$F(c) = 4$
c=9	$F(c) = 8$	$F(c) = 4$	$F(c) = 4$	$F(c) = 9$
c=10	$F(c) = 9$	$F(c) = 1$	$F(c) = 1$	$F(c) = 16$
c=11	$F(c) = 10$	$F(c) = 0$	$F(c) = 0$	$F(c) = 0$
$c_S = 1$	$c_C = 11$	$c_B = 6$	$c_E = 6$	$c_F = 1$

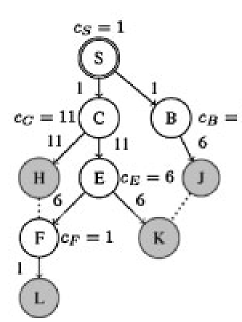

Fig. 4. Channel assignments in M4. The hidden channel problem in MCM (Figure 2) no longer exists in M4.

DOI: 10.1002/wcm

CAs obtained from MCM (Figure 2(b)), the CAs given by M4 show no hidden channels and no interference among multicast nodes.

4.2. Implementation

We discuss two major issues, namely the collection of one-hop and two-hop neighbor information and the computation of function $F(c)$.

The channel information of two-hop neighbors can be obtained in a distributed manner as follows. After a node v obtains the channel assignment it is assigned, it broadcasts a message containing its node ID and the assigned channel number to its one-hop neighbors. When a neighbor u of node v receives the broadcast message, u adds its own ID and channel information to the message (if the channel information is available at that time). Node u then broadcasts the updated message to its neighbors. Besides channel information, these broadcast messages also provide us with the set of one-hop and two-hop neighbors of a node v that have been assigned channels, $N^{\diamond}(v)$.

In our implementation, we include a hop count value with each channel information entry so that the information is not propagated beyond two-hop neighbors. Each message is broadcast three times to increase delivery reliability. To minimize collision, when a node receives a channel information message, it does not re-broadcast a message immediately, but waits for some random amount of time whose value is drawn from a distribution. Since we assume static membership, the broadcast messages are transmitted before data transmission starts, and thus do not interfere with the data traffic. When we consider dynamic joins/leaves and traffic loads, channels will need to be re-assigned based on the changes. In that case, we will need more efficient algorithms for nodes to exchange channel and neighbor information. This, along with support for dynamic membership and traffic loads, will be addressed in our future work (see also 4.3).

With respect to the computation of function $F(c)$, when Equation (2) gives more than one solution with the same optimal value F_c, we break the tie as follows. The solution with the most number of orthogonal channel pairs is selected because orthogonal channels are more favorable than overlapping channels. If the solutions have the same number of orthogonal channel pairs, the one with the node having the least number of one-hop and two-hop neighbors is chosen to further minimize the interference.

4.3. Discussion

When designing the M4 CA algorithm, we recognized two major types of interference among nodes: intra-flow and inter-flow. We discuss how the proposed M4 algorithm handles these types of interferences.

(1) *Intra-flow interference.* For a unicast flow, this is the interference among nodes on the path from the source to the destination. For a multicast group, we consider the whole tree/group as a flow. Multicast intra-flow interference is thus among nodes in the routing *tree*. When we include all one-hop and two-hop neighbors of a multicast node v (i.e., set $N^{\diamond}(v)$) in Equation (2) of the M4 algorithm, we take into account all possible relatives of v within two-hop transmission of v in the routing tree, including siblings (i.e., multicast nodes at the same level in the tree).
(2) *Inter-flow interference.* This is the interference among nodes belonging to different flows. In the current implementation, we consider only intra-flow interference (i.e., among nodes in a multicast tree). To account for inter-flow interference, we can incorporate the channel information of other flows into Equation (2) when performing CA for the multicast group. We can also apply the M4 algorithm to unicast flows by collecting the channel information of one-hop and two-hop neighbors of the nodes on the source-to-destination path and then apply Equation (2).

Figure 5 illustrates an example which re-uses the multicast group shown in Figure 4. Assume that the unicast flow with source M and destination N has been active on channel 2 when the multicast session starts. When we solve Equation (2) for multicast forwarding node F, node M of the unicast flow is also included in the computation; that is, $N^{\diamond}(F) = \{C, E, M\}$, resulting in $c_F = 9$ (recall that $c_F = 1$ in Figure 4 when there are no other flows).

When inter-flow interference is included in the CA algorithm, traffic loads must be considered in order to obtain the optimal solution. For instance, when a new flow starts, the CA should be recomputed to account for the interference caused by the new flow. Similarly, when a flow terminates, the CA must be updated to exclude the interference of this flow. Furthermore, a flow with very light load incurs less interference than one with heavy load. In other words, the traffic load of a flow determines its level of interference, and thus must

DOI: 10.1002/wcm

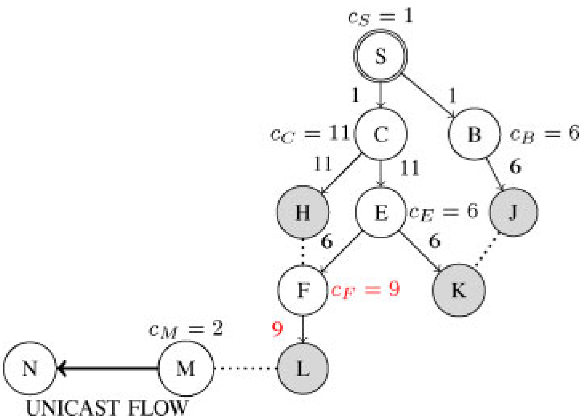

Fig. 5. Inter-flow interference: co-existence of multicast and unicast flows.

be considered in a CA algorithm, as done in several CA algorithms (for unicast communications) [4,5,8,33].

The CA could be updated when the traffic load changes. However, network load conditions can be very dynamic; it could be very expensive to keep track of every change and re-compute the CA accordingly. Therefore, most CA algorithms use the simple method of periodic updates [4,8,33], i.e., the CA is updated periodically, e.g., every one minute [8].

Our future work is to evaluate the proposed algorithm with dynamic network traffic loads.

5. Experimental Results

We evaluate the performance of M4 and compare it with that of MCM using QualNet simulator version 4.0 [34]. M4 offers two advantages over MCM: (1) eliminating the HCP, and (2) avoiding the use of interference factors and random channel selections when there exist multiple choices. Therefore, we evaluated two versions of MCM:

- Original MCM (denoted by MCM).
- Improved MCM (denoted by i-MCM). We modified MCM to eliminate the HCP by considering two-hop neighbors in the CA procedure, but kept the original form of the MCM optimization function (Section 3.1). By comparing M4 with i-MCM, we show that we can avoid using interference factors in the CA algorithm without performance degradation, and that our channel selection strategy performs better than that of MCM.

The multicast ad-hoc on-demand distance vector (MAODV) protocol [22] is used to build multicast routing trees (although the CA algorithms of M4 and MCM can be applied to any types of tree). MAODV is provided by QualNet.

Following are our performance metrics, simulation parameters, and results.

5.1. Performance Metrics

We use the following metrics to measure the performance of the M4 and MCM algorithms:

- *Average multicast packet delivery ratio*. The PDR of a receiver is the number of data packets actually delivered to the receiver versus the number of data packets supposed to be received. The average PDR of a multicast group is the average of the PDRs of all the receivers in the group.
- *Average end-to-end delay*. The end-to-end delay of every packet received at every receiver is recorded; the average over all the packets received is then computed.
- *Average throughput*. The throughput is defined as the total amount of data a receiver actually receives divided by the time between receiving the first packet and the last packet. The average taken over all the receivers is the average throughput of the multicast group, assuming that each group has one sender.

5.2. Simulation Parameters

We simulated a small network of 50 wireless routers distributed over a 1000 m × 1000 m area, and a medium-size network of 100 wireless routers, over a 1700 m × 1700 m area. The nodes are distributed uniformly over the sub-areas within a terrain, and the nodes within a sub-area are randomly placed in that space. There are no network partitions throughout the simulation.

The transmission power of the routers is set constant at 20 dBm; the transmission range of the wireless routers is 315 m. We use PHY802.11b at the physical layer with a transmission rate of 11 Mbits/s. A two-ray propagation model [35] is used when the distance between two routers is 250 m or more; otherwise, a free space model is used to avoid the oscillation caused by the constructive and destructive combination of the two rays over short distances. The above distance threshold for switching between the two models is calculated by the QualNet software.

DOI: 10.1002/wcm

Table II. Common simulation parameters.

Parameter	Value
Network size	50 nodes over a 1000 m × 1000 m area 100 nodes over a 1700 m × 1700 m area
Path loss model	Free space for distances below 200 m two-ray for distances of 200 m or more
Router transmission power	20 dBm
Transmission rate at physical layer	11 Mbits/s
Physical layer protocol	PHY802.11b
Medium access control	MAC802.11 with DCF
MAC for multicast flows	CSMA/CA
Packet size (excluding header size)	512 bytes
Queue size at routers	50 Kbytes
Queuing policy at routers	First-in-first-out
Traffic model of sources	Constant bit rate (CBR)
Duration of each experiment	400 s of simulated time
Number of runs per data point	10

The MAC802.11 protocol with DCF (Distributed Coordination Function) is chosen as the MAC protocol.We implemented only CSMA/CA without RTS (right to send), CTS (clear to send) or ACK (acknowledgment) for multicast MAC. There currently does not exist an effective algorithm for implementing RTS/CTS/DATA/ACK exchanges at the branch points of a multicast tree for the following two reasons. First, CTS packets sent by the multicast neighbors of a transmitter have a very high probability of colliding at the transmitter. More importantly, it may not be possible for all the multicast neighbors to agree on a common time slot for the transmission of a packet, or the delay would be very long to reach such an agreement. Therefore, all multicast implementations in 802.11-based wireless networks so far have used only CSMA/CA without RTS/CTS/DATA/ACK exchanges.

The data packet size excluding the header size is 512 bytes. The size of the queue at every node is 50 Kbytes. The packets in a queue are scheduled on a first-in-first-out basis. We did not implement flow or congestion control in order to test the network performance under very high loads.

Each multicast group has one sender. The sender of a multicast group transmits at a *constant bit rate* properly set for each experiment. The number of receivers (the group size) is also specified for each scenario. We assume that each sender or receiver is connected to a *different* wireless router since our work focuses on the mesh backbone. (In practice, there can be many hosts communicating with a wireless router, e.g., to form a wireless local area network.) The sender and the receivers of a multicast group were selected randomly, and the same sender and receivers and the same network configuration were used for all CA algorithms in order to obtain a fair comparison. All receivers joined a multicast group at the beginning and stayed until the whole group terminated.

In each experiment, the source sent data for 300 s of simulated time, at a constant bit rate specified for each experiment. After the source finished sending, the simulation continued to run for 100 s of simulated time to give the last packets time to be processed and routed, for a total of 400 s. This 400 s duration did not include the time needed for constructing the routing tree at the beginning. Each data point in the graphs was obtained from 10 runs using different randomly generated seed numbers, and the collected data were averaged over the 10 runs.

The above parameters are summarized in Table II.

To confirm the results reported in this paper, we also created two more configurations for each data point by changing the node placement in the network and multicast senders and receivers, and repeated the experiments. The results from these configurations are consistent with those presented in this paper.

5.3. Scenarios

We considered a small network of 50 nodes in a 1000 m × 1000 m area and a medium-size network of 100 nodes in a 1700 m × 1700 m area. For each network size, we measure the average PDR, throughout and end-to-end delay as functions of

- *the sender's sending rate* at the application layer (i.e., traffic load). The rate varies from 10 to 100 packets/s. The multicast group sizes are 20 and 35 receivers in the small and medium networks, respectively. We implemented the IEEE 802.11b 11-channel system.

DOI: 10.1002/wcm

Table III. Simulation scenarios.

Function of	Parameters	50-node network	100-node network
Traffic load	Sending rate	from 10 to 100 packets/s	
	Number of channels	11	
	Number of receivers	20	35
Multicast group size	Sending rate	60 packets/s	40 packets/s
	Number of channels	11	
	Number of receivers	from 1 to 30	from 1 to 55
Number of channels	Sending rate	60 packets/s	40 packets/s
	Number of channels	from 1 to 20	
	Number of receivers	20	35

- *the multicast group size*. The number of receivers ranges from 1 to 30 in the small network and from 1 to 55 in the medium network. The number of channels is 11. The source sending rates are 60 packets/s and 40 packets/s in the small and medium networks, respectively. These rates generate a moderate load in the given networks and group sizes.
- *the number of channels*. The values range from 1 to 20. Any two channels are orthogonal if they are separated by at least four channels (e.g., channels 2 and 7). In the small network, there are 20 receivers, and the traffic load is set at 60 packets/s. In the larger network, the multicast group contains 35 receivers and the traffic load is 40 packets/s.

A summary of these above parameters is shown in Table III.

5.4. Function of Traffic Load

In this set of experiments, the sender's rate varies from 10 to 100 packets/s. The multicast group in the 50-node network, has 20 receivers, and the results are given in Figure 6.

When the traffic load is light (10–20 packets/s), the three algorithms perform similarly with respect to PDR and throughput. When the load is light, there is less medium contention and usage; the multicast group did not take advantage of MCMR. A single channel would have been adequate in this case. Therefore, the three algorithms perform similarly.

When the traffic load is moderate to heavy (above 40 packets/s), the advantage of MCMR clearly demonstrates, which leads to M4 outperforming i-MCM and MCM. For instance, under heavy load, 80 packets/s, the PDRs of M4, i-MCM, and MCM are 85, 79, and 64%, respectively, a difference of 21% between M4 and MCM.

The performance gap between M4 and MCM is larger than that between M4 and i-MCM. This indicates that the HCP is the main factor weighing down the performance of MCM.

M4 performs better than i-MCM, thanks to a channel selection strategy better than random selection when there exist multiple choices. The results also show that using simple channel numbers as a measure of channel separation in M4 is just as effective as using interference factors in MCM.

M4 offers the lowest average end-to-end delay, about 26 and 19% lower, than MCM and i-MCM, respectively. Better CA resulted in lower contention for medium, and thus lower end-to-end delay.

We now examine the performance of the three algorithm as function of traffic loads in the larger network of 100 nodes with 35 multicast receivers. The graphs are shown in Figure 7. As above, M4 performs similarly to MCM and i-MCM under light loads (10–20 packets/s), and significantly better under heavier loads with respect to all metrics.

The performance gap between M4 and i-MCM in the 100-node network is more pronounced than that in the smaller network. For instance, when the number of receivers is 20 and the traffic load is 60 packets/s, the PDRs of M4 and i-MCM in the network of 50 nodes are 89 and 86%, respectively (Figure 6(a)), while the PDRs in the network of 100 nodes are 85 and 73%, respectively (Figure 7(a)). In the same scenario, the average end-to-end delay given by i-MCM is about 19% higher than that of M4 in the smaller network (42 ms vs. 33 ms in Figure 6(c)), and about 32% higher in the larger network (65 ms vs. 44 ms in Figure 7(c)). The reason is that longer source-to-destination paths in the larger network take more advantage of the better channel selection algorithm of M4. Similarly, the performance gap between M4 and MCM also widens in the 100-node network. The HCP in MCM caused

DOI: 10.1002/wcm

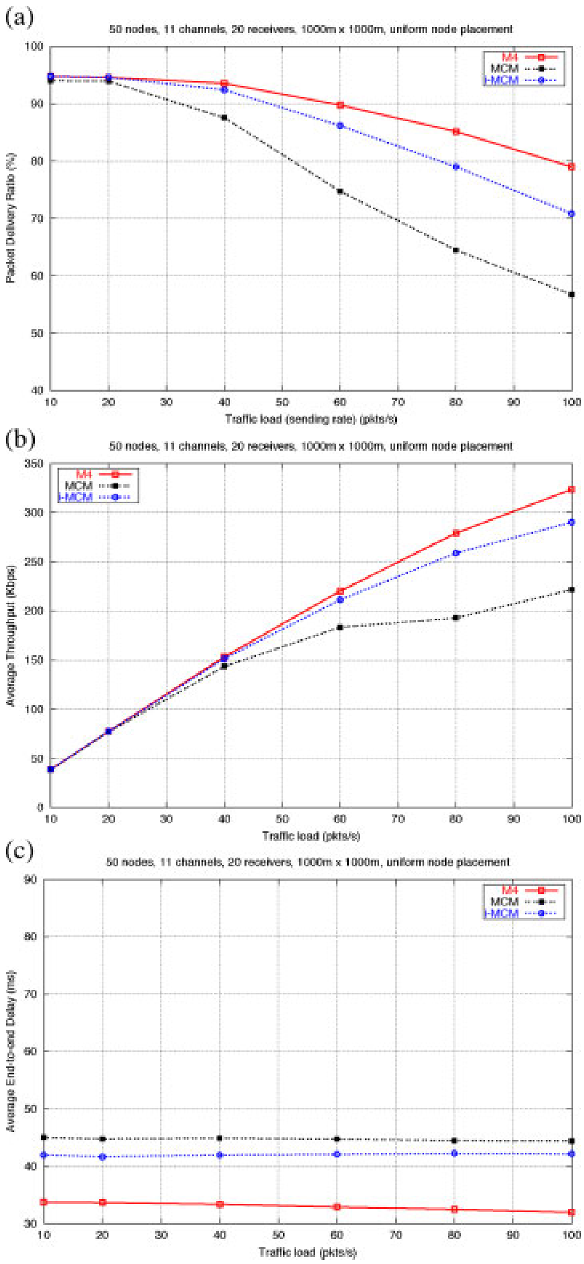

Fig. 6. Functions of traffic load—50-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

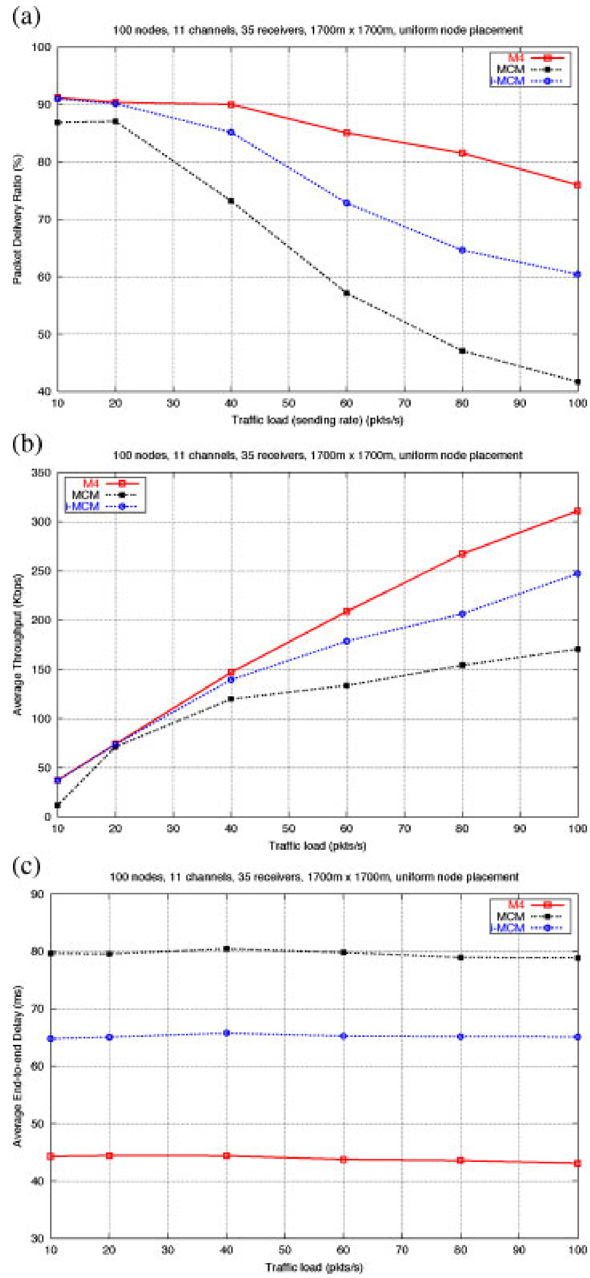

Fig. 7. Functions of traffic load—100-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

more collision and congestion when there were more nodes on a source-to-destination path.

For all three algorithms in both networks, as the sender's rate increases, the throughput increases as expected; the PDR decreases because higher loads cause more congestion and collisions, resulting more packets dropped or damaged.

5.5. Function of Number of Channels

The number of channels in this set of experiments is varied from 1 to 20. The multicast group in the 50-node network has 20 receivers, and its source sends at a rate of 60 packets/s. This rate yields a moderate load for the given group size in this network.

DOI: 10.1002/wcm

The results in Figure 8 show that M4 and i-MCM outperform MCM in all cases, thanks to the elimination of the HCP. When the number of channels is 20, the PDRs of M4 and MCM are 94.1 and 89.6%, respectively. The average end-to-end delay of M4 is 25.8% lower than that of MCM. Note that in the network with only one channel the results from the three algorithms are almost the same as we would expect.

The performance of M4 is only slightly better than than of i-MCM in this set of experiments. Note, however, that our intention was to replace interference factors with a metric that is simpler, more convenient, and more flexible. To that end, our optimization

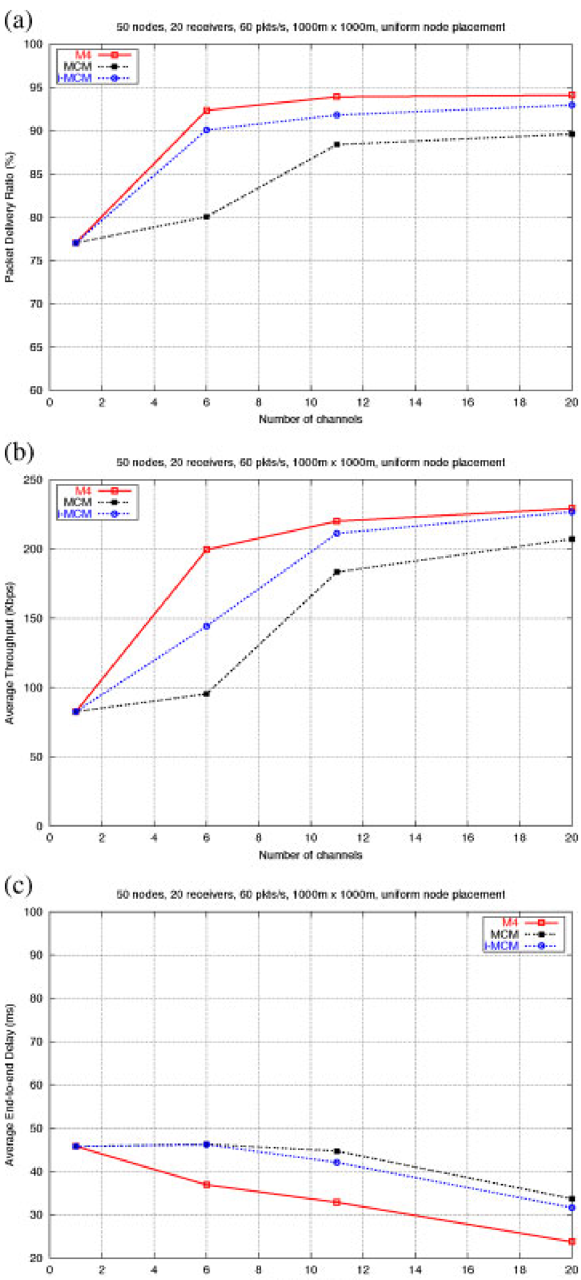

Fig. 8. Functions of number of channels—50-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

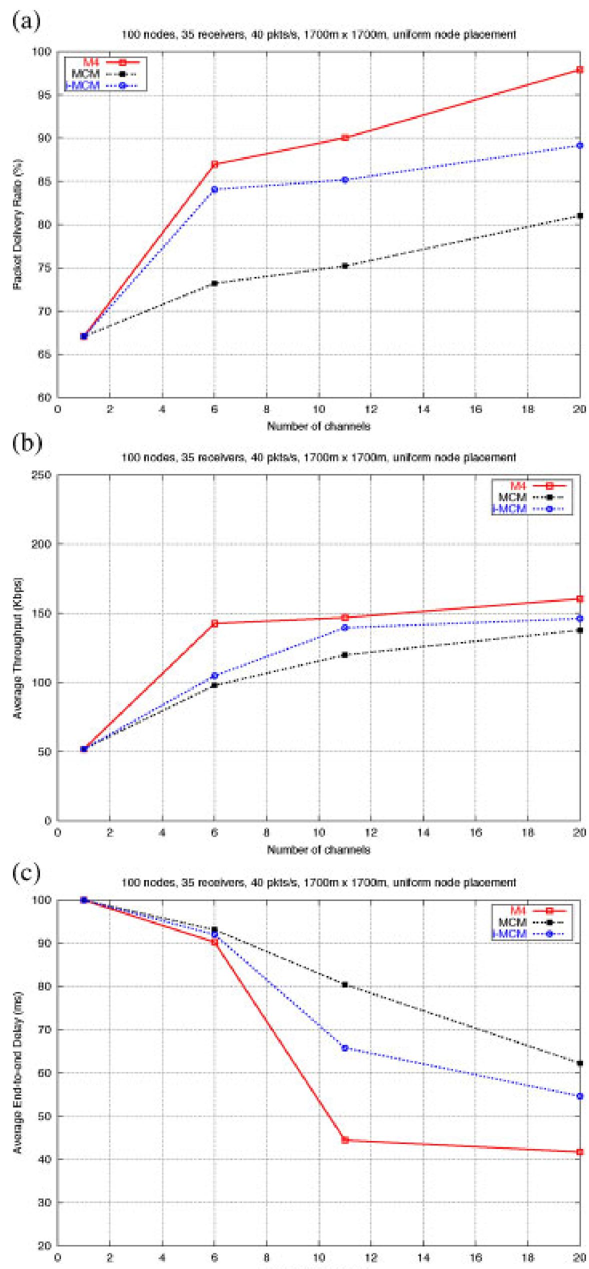

Fig. 9. Functions of number of channels—100-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

function using simple channel numbers to measure the degree of channel separation proves to be as effective as interference factors, because M4 performs similarly to or better than i-MCM in all cases.

In the 100-node network, we simulated a multicast group having 35 receivers and a source rate of 40 packets/s (Figure 9). Again, M4 performs better than MCM and i-MCM. The performance gap between M4 and MCM/i-MCM magnifies as the network size increases, for the same reason as explained above (longer source-to-destination paths).

For all three algorithms in both networks, as the number of channels increases, the PDR and throughput

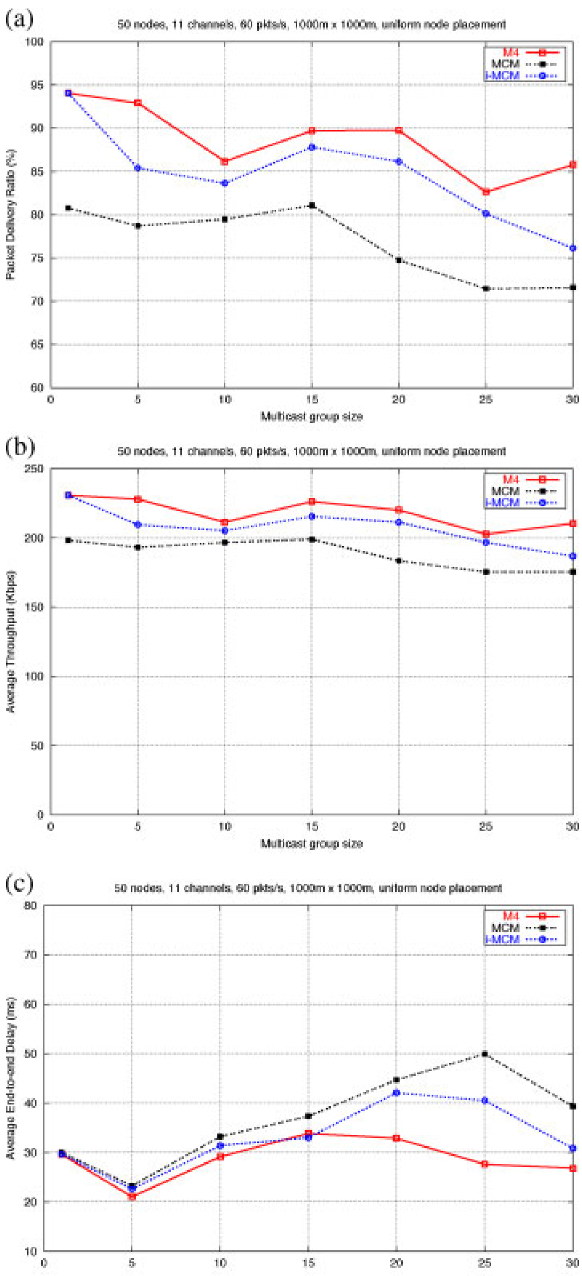

Fig. 10. Functions of group size—50-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

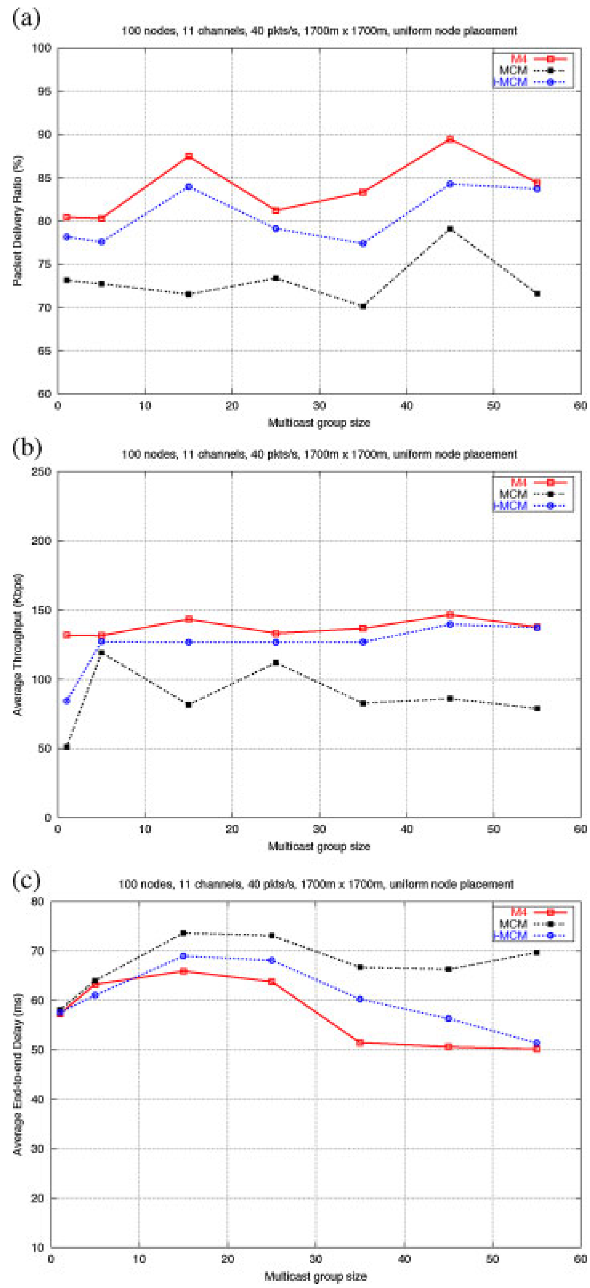

Fig. 11. Functions of group size—100-node network. (a) Average PDR. (b) Average throughput. (c) Average end-to-end delay.

increases, and the average end-to-end delay decreases. The higher the number of channels, the less time spent contending for the medium. However, the performance of M4 increases at a faster rate than MCM, thanks to the elimination of the HCP and better optimization function.

5.6. Function of Group Size

In the 50-node network, the multicast group size is increased from 1 to 30, while the number of channels and sender's rate are set to 11 and 60 packets/s, respectively. The results are shown in Figure 10, which indicate that M4 performs better than MCM and i-MCM in almost all cases, especially when the group size is large. For instance, when the number of receivers is 30, the PDR of M4 is 85%, i.e., 9 and 14% higher than that of i-MCM and MCM, respectively (Figure 10(a)). Similarly, the average end-to-end delay of M4 is 12.9% lower than that of i-MCM (27 ms vs. 31 ms) and 30.7% lower than that of MCM (27 ms vs. 39 ms), as shown in Figure 10(c).

As the group size increases, the more traffic is created in the network. Therefore, the PDR and throughput of the three algorithms go down slightly. Similarly, the average end-to-end delays tend to go up as the group size increases.

We conducted the same experiment with 100-node network with the group size varying from 1 to 55. The results of this experiment, shown Figure 11, provide a similar comparison between M4, MCM, and i-MCM. Specifically, when the multicast group size was 45, M4 achieved a PDR of 89%, while the PDRs of i-MCM and MCM were lower, at 84 and 79%, respectively (Figure 11(a)). The average end-to-end delay of M4 was 50 ms, 10.7 and 24% lower than that of i-MCM (56 ms) and MCM (66 ms), respectively (Figure 11(c)). Again, although i-MCM performs better than MCM, its results were not as good as M4 due to M4's better optimization function.

6. Related Work

Recent work on multicast in WMNs focuses on multicast routing and performance study of routing approaches in single-channel networks [28–30,36–39]. On the other hand, the problem of CA in multi-radio multi-channel WMNs has been studied extensively for unicast communications [5,6,8,9,40–42].

In the context of unicast communications, the CA problem can be classified into three approaches: (1) routing first, CA second [8,9]; (2) CA first, routing second [6,40,41]; and (3) joint CA and routing [5,42]. For instance, the protocol by Raniwala and Chiueh [8] performs routing first, followed by CA. The CA algorithm is called *load-aware CA*, because the traffic loads of the links are known at the time CA is performed. The protocol carries out the procedure of routing and CA periodically because link traffic loads may change over time. Tang *et al*. [6] use the second approach in their algorithm: CA is done first, followed by routing. Thanks to the CA result, the interference among links is given, and routing under this constraint is called *interference-aware routing*. Alicherry and Li [5], on the other hand, use linear programming to solve the problems of CA and routing simultaneously, taking into account the inter-dependence between routing and CA to maximize the network throughput. Most of existing work on CA for unicast communications assumes orthogonal channels.

The problem of CA for multicast has only been studied recently [11,12]. The MCM algorithm [11] suffers from the HCP as discussed earlier. The optimization function in the algorithm by Yin *et al*. [12] depends on the use of the probability that a channel is being busy. The paper did not mention how to compute this probability; furthermore, collecting and maintaining this information for all links in the network would incur high overheads. Both the above algorithms and ours in this paper assume that a multicast routing tree is first constructed, and CA is then applied (i.e., the first approach listed above).

7. Conclusion

We propose a CA algorithm for multicast in MCMR WMNs. We discuss the drawbacks of the MCM algorithm, and propose the solution to the HCP as well as an optimization function that does not rely on the computation of interference factors. Advantages of the proposed CA algorithm include its simple implementation and high performance. Our simulation results show that the M4 algorithm outperforms MCM in terms of average PDR, throughput, and end-to-end delay under various network conditions. Our future work will address the problem of dynamic group membership and incorporate traffic load information into the CA algorithm.

DOI: 10.1002/wcm

References

1. Gupta P, Kumar PR. The capacity of wireless networks. *IEEE Transactions on Information Theory* 2000; **46**: 388–404.
2. Li J, Blake C, Couto DSJD, Lee HI, Morris R. Capacity of ad hoc wireless networks. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom'01)*, 2001; pp. 61–69.
3. Draves R, Padhye J, Zill B. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM MobiCom'04*, 2004; pp. 114–128.
4. Raniwala A, Gopalan K, Chiueh T. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In *Proceedings of ACM SIGMOBILE'04*, 2004; pp. 50–65.
5. Alicherry M, Bhatia R, Li L. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proceedings of ACM SIGMOBILE'05*, 2005; pp. 58–72.
6. Tang J, Xue G, Zhang W. Interference-aware topology control and QoS routing in multi-channel wireless mesh networks. In *Proceedings of ACM SIGMOBILE'05*, 2005; pp. 68–77.
7. Kyasanur P, Vaidya NH. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. In *Proceedings of ACM SIGMOBILE'06*, 2006; pp. 31–43.
8. Raniwala A, Chiueh T. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh networks. In *Proceedings of IEEE InfoCom'05*, 2005; pp. 2223–2234.
9. Das AK, Alazemi HMK, Vijayakumar R, Roy S. Optimization models for fixed channel assignment in wireless mesh networks with multiple radios. In *Proceedings of IEEE SECON'05*, 2005; pp. 463–474.
10. Fuxjager P, Valerio D, Ricciato F. The myth of non-overlapping channels: interference measurements in IEEE 802.11. In *Proceedings of IEEE Wireless on Demand Network Systems and Services (WONS'07)*, 2007; pp. 1–8.
11. Zeng GW, Ding B, Xiao YL, Mutka M. Multicast algorithms for multi-channel wireless mesh networks. In *Proceedings of IEEE International Conference on Network Protocols (ICNP'07)*, 2007; pp. 1–10.
12. Yin Z, Li Z, Chen M. A novel channel assignment algorithm for multicast in multi-radio wireless mesh networks. In *Proceedings of IEEE ISCC'07*, 2007; pp. 283–288.
13. O'Hara B, Petrick A. *802.11 Handbook: A Designer's Companion*. IEEE Press: New York, USA, 2005.
14. Nguyen UT, Nguyen HL. Multicast routing algorithms. *Technical Report CSE-2008-08*, Department of Computer Science and Engineering, York University, 2008.
15. Dijkstra EW. A note on two problems in connexion with graphs. *Numerical Mathematics* 1959; **1**: 269–271.
16. Bellman R. *Dynamic Programming*. Princeton University Press: New Jersey, USA, 1957.
17. Cheng C, Riley R, Kumar S, Garcia-Luna-Aceves J. A Loop-free extended bellman-ford routing protocol without bouncing effect. *ACM SIGCOMM Computer Communications Review*, Vol. 19, September 1989; pp. 224–236.
18. Sloman MS, Andriopoulos X. Routing algorithm for interconnected local area networks. *Computer Networks and ISDN Systems* 1985; **9**: 109–130.
19. Waitzman D, Partridge C, Deering S. Distance Vector Multicast Routing Protocol. *RFC 1075*, November 1988.
20. Moy J. Multicast Extensions to OSPF. *RFC* 1584, March 1994.
21. Lee SJ, Gerla M, Chiang CC. On-demand multicast routing protocol. In *Proceedings of IEEE WCNC'99*, 1999; pp. 1298–1302.
22. Royer EM, Perkins CE. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of ACM/IEEE MobiCom'99*, 1999; pp. 207–218.
23. Kou L, Markowsky G, Berman L. A fast algorithm for steiner trees. *Acta Informatica* 1981; **2**: 141–145.
24. Takahashi H, Matsuyama A. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica* 1980; **24**: 573–577.
25. Zelikovsky A. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica* 1993; **9**: 463–470.
26. Winter P. Steiner problem in networks: a survey. *Networks* 1987; **17**: 129–167.
27. Winter P, Smith JM. Path-distance heuristics for the Steiner problem in undirected networks. *Algorithmica* 1992; **7**: 309–327.
28. Ruiz PM, Gomez-Skarmeta AF. Approximating optimal multicast trees in wireless multihop networks. In *Proceedings of IEEE ISCC'05*, 2005; pp. 686–691.
29. Nguyen U, Xu J. Multicast routing in wireless mesh networks: minimum cost trees or shortest path trees? *IEEE Communications Magazine* 2007; **45**: 72–77.
30. Nguyen HL, Nguyen UT. 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).*
31. Weiss MA. *Data Structures and Algorithm Analysis in Java*. Addison-Wesley: Boston, USA, 2007.
32. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. The MIT Press: Cambridge, MA, USA, 2001.
33. Lin C-J, Chou C-F. Route-aware load-balanced resource allocation for wireless mesh networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'07)*, 2007; pp. 3093–3511.
34. Qualnet Simulator. Available online at: http://www.qualnet.com
35. Schafer W, Lutz E. Propagation characteristics of short-range radio links at 60 GHz for mobile intervehicle communication. In *Proceedings of IEEE Telecommunications Symposium (ITS'90)*, 1990; pp. 212–216.
36. Roy S, Koutsonokolas D, Das S, Hu YC. High-throughput multicast routing metrics in wireless mesh networks. In *Proceedings of IEEE ICDCS'06*, 2006; pp. 48–48.
37. Zhao X, Chou CT, Guo J, Jha S. Protecting multicast sessions in wireless mesh networks. In *Proceedings of IEEE Conference on Local Computer Networks (LCN'06)*, 2006; pp. 467–474.
38. Yuan J, Li Z, Yu W, Li B. A cross-layer optimization framework for multihop multicast in wireless mesh networks. *IEEE Journal on Selected Areas in Communications* 2006; **24**: 2092–2103.
39. Zhao X, Chou CT, Guo J, Jha S. Protecting multicast sessions in wireless mesh networks. In *Proceedings of IEEE LCN'06*, 2006; pp. 467–474.
40. Marina M, Das S. 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; pp. 381–390.
41. Subramanian AP, Gupta H, Das SR. 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; pp. 481–490.
42. Kodialam M, Nandagopal T. 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; pp. 73–87.

DOI: 10.1002/wcm

Authors' Biographies

Hoang Lan Nguyen is currently a Ph.D. student at the Department of Computer Science and Engineering, York University (Toronto, Canada). He received his B.E degree with the highest honors in Telecommunications in 2003 from the University of Wollongong (NSW, Australia), and his M.Sc. degree in Computer Science from York University in 2006. Before joining York University, he worked for Australia Nortel Networks as a software engineer. His current research interests include wireless communications, multicast routing, and network security.

Uyen Trang Nguyen received her Bachelor of Computer Science and Master of Computer Science degrees in 1993 and 1997 from Concordia University, Montreal, Canada. She completed her Ph.D. degree at the University of Toronto, Canada, in 2003. From 1995 to 1997 she was a software engineer at Nortel Networks, Montreal, Canada. She joined the Department of Computer Science and Engineering at York University, Toronto, Canada, in 2002 and is currently an Associate Professor. Her research interests are in the areas of wireless ad-hoc networks, multipoint communications, and multimedia applications.

DOI: 10.1002/wcm