Algorithms for Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks

Hoang Lan Nguyen, Uyen Trang Nguyen
Department of Computer Science and Engineering, York University
4700 Keele Street, Toronto, Canada M3J 1P3
Email: {lan,utn}@cse.yorku.ca

***Abstract*—Traditional multicast routing algorithms such as shortest path tree (SPT) and Steiner tree (MST) do not consider the wireless broadcast advantage or the underlying channel assignments in a multi-channel multi-radio (MCMR) wireless mesh network (WMN). We propose multicast routing algorithms that take into account the above factors in order to minimize the amount of network bandwidth consumed by a routing tree. Experimental results show that routing trees constructed by the proposed algorithms outperform traditional trees such as SPTs, MSTs and minimum number of forwarders trees (MFTs) with respect to packet delivery ratio, throughput and end-to-end delay.**

I. Introduction

Traditional multicast routing algorithms such as SPT or Steiner tree do not consider the wireless broadcast advantage (WBA) or the channel assignments (CA) (i.e. channel diversity) in a MCMR WMN. The WBA refers to the fact that the delivery of a data packet from a given node to any number of its neighbors can be done with a single transmission. Two nodes are said to be neighbors if they can communicate directly with one another without going through intermediate nodes. We propose multicast routing algorithms that minimize the amount of network bandwidth consumed by a routing tree by taking into consideration the WBA and channel diversity. Given a MCMR network and a CA scheme, the algorithms construct multicast routing trees that minimize the total number of transmissions required to deliver a data packet from the source to all multicast destinations. Experimental results show that our routing trees outperform commonly used/cited trees such as SPTs, MSTs and MFTs [1] in terms of packet delivery ratio, throughput and end-to-end delay.

The remainder of the paper is organized as follows. We discuss related work in Section II, and define the problem to be solved in Section III. In Section IV, we first describe the proposed algorithm and then a distributed implementation of the algorithm. Experimental results comparing the performance of the proposed routing trees with that of SPTs, MSTs and MFTs are presented in Section V. Section VI concludes the paper and outlines our future work.

II. Related Work

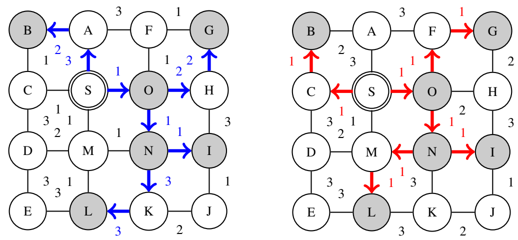
Research work on multicast in *single-channel* WMNs focuses mainly on multicast routing and performance study of routing approaches [1], [2], [3], [4]. The topic of channel assignment and routing for multicast in *multi-channel multi-radio* networks has been studied only recently [5], [6], [7], [8], [9], [10], [11]. The algorithms proposed in [5], [6], [7], [8] aim at minimizing the interference among multicast nodes and maximizing throughput using the "routing first, CA second" approach wherein a multicast routing tree is first constructed, and a CA scheme minimizing interference is then applied to the tree. Using this approach, a node may have more assigned channels than the number of available radios, which requires channel switching. However, currently no channel switching algorithm for multicast is available. Furthermore, channel switching adds considerable delay to data routing in MCMR networks [12].

In this paper, we consider the "CA first, routing second" approach. Given a MCMR network with a pre-determined CA scheme and a multicast group, we construct a multicast routing tree with a minimized number of transmissions, and thus minimize the bandwidth consumption of the tree. Ruiz et al. [1] propose algorithms that build multicast trees with minimized numbers of forwarding nodes, and hence minimized numbers of transmissions, in *single-channel* networks. Our proposed algorithms, on the other hand, minimize the number of transmissions incurred by a multicast tree in a *multi-channel multi-radio* network. Also using the "CA first, routing second" approach, Lim et al. [9] consider the existing CA to minimize the number of channel conflicts within two-hop distance, but their algorithm requires channel switching.

There exist also algorithms that consider both routing and CA simultaneously [10], [11]. Cheng et al.'s algorithm [10] constructs a multicast tree and a CA scheme with minimized channel conflict and minimal tree cost (defined as the total number of radios used by the nodes in the tree). Chiu et al. [11] propose a CA and tree construction scheme that satisfies a bandwidth constraint. The main limitation of the scheme is the assumption of a *perfect*, no-collision MAC scheduler.

III. Problem Definition

We consider MCMR WMNs with stationary wireless routers (nodes). Two nodes are directly connected and form a communication link if they are within the transmission radio range of each other and share a common channel. We assume that a CA scheme [12], [13], [14], [15], [16], [17], [18] is independently applied to the network prior to the construction of the multicast tree. We also make the following assumptions,



(a) Multicast tree T_1 (b) Multicast tree T_2

Fig. 1. A network with three channels and two radios per node. Each link is labeled with the assigned channel.

which are common constraints imposed by MCMR wireless mesh networking.

- The channels are orthogonal (non-overlapping).
- The channel assigned to a link is used for transmissions in both directions of the link.
- The radio antennas are omnidirectional.
- For any node, the number of distinct channels assigned to the node is less than or equal to the number of radios the node possesses. As a result, each radio is bound to a specific channel and no channel switching is needed.

Following is an informal definition of the problem by means of an example. A formal definition can be found in [19].

Consider the MCMR network shown in Fig. 1. Assume that the network has three orthogonal channels and each node has two radios. Given a multicast group with source S and six destinations B, G, I, L, N and O, we show two possible routing trees for this multicast group in Fig. 1.

The problem focuses on the number of transmissions a forwarding node requires to multicast a packet to its one-hop neighbors in the routing tree. For example, in both trees, node N is used as a forwarding node to deliver multicast data to destinations I and L. In tree T_1 (Fig. 1(a)), N has to transmit two copies of every packet (i.e., two transmissions), one on channel 1 to I and the other on channel 3 to K, which will forward the packet to L. However, in tree T_2 (Fig. 1(b)), N needs to perform only one transmission on channel 1 to reach both I and M, which will forward the packet to L. This shows that the choice of route affects the number of transmissions a node has to perform to forward a data packet.

If we sum up the numbers of transmissions that all forwarding nodes in a routing tree T need to perform to deliver a packet to their multicast neighbors in T, the result is the total number of transmissions the tree incurs to deliver a packet from the source to all the destinations, denoted by $S(T)$. (We do not consider retransmissions caused by packet loss or errors.) For the example trees T_1 and T_2 in Fig. 1, $S(T_1) = 9$ while $S(T_2) = 6$. Tree T_2 is preferred because it requires less transmissions per packet and thus consumes less network bandwidth. Among the possible trees connecting the source to the destinations, our goal is to find a tree with the minimum $S(T)$. Finding such a tree is an NP-hard problem [19]. We thus propose heuristic algorithms to find approximate solutions, which constructs Multi-Channel Minimal Number of transmissions Trees (MCMNTs).

IV. The MCMNT Algorithms

We present the MCMNT algorithm in Section IV-B, followed by its distributed implementation in Section IV-C. We first define link and path costs.

A. Definitions of Link Cost and Path Cost

Consider a connected graph $G = (V, E)$, where V is the set of stationary mesh routers (nodes), and E is the set of communication links (edges) with pre-assigned channels. For each node $u \in V$, $\mu_u(c)$ denotes the number of links that are incident on u and assigned channel c. For example, for node A in the graph shown in Fig. 1, $\mu_A(1) = 0$, $\mu_A(2) = 1$ and $\mu_A(3) = 2$. Value $\mu_u(c)$ can be considered as the *channel utilization* of channel c by node u: the higher the value, the more neighbors u can reach with a single transmission on channel c.

Since a *high* channel utilization value is desirable while the heuristic selects paths based on *minimum* costs, we convert channel utilization to a metric whose smaller values are more favorable than higher values in order to perform least cost path selection. In our heuristic, we take the inverse of the channel utilization value $\mu_u(c)$ and assign it to a new channel metric denoted by $\delta_u(c) = 1/\mu_u(c)$. For any link (u,v) $\in E$, both $\mu_u(c)$ and $\mu_v(c)$ are greater or equal to 1, making the inverse function always defined.

Each directional link (u, v) is associated with a link cost $w(u, v)$ defined as:

$$w(u, v) = \delta_u(c)/\delta_v(c), \tag{1}$$

where c is the channel used by link (u, v). The originating node u of the directional link (u, v) is termed the *transmitter*, while the ending node v, the *receiver*. We favor a transmitter with a channel *highly utilized* so that the channel can be used for as many receivers as possible in the final tree. This explains the term $\delta_u(c)$ in the link cost.

If a link (u, v) using channel c has been added to the tree, the next link (v, z) to be added should avoid using channel c so that transmissions from u and v do not interfere because u and v are one-hop neighbors of each other. Therefore, given a transmitter u and an assigned channel c, we should choose a receiver v whose channel c is *lowly utilized* so that node v will have less chance of being selected next as a transmitter *on* channel c. Hence the term $1/\delta_v(c)$ in the link cost.

Finally, let $P(s, d)$ denote a path connecting a source s to a destination d. The path cost $W(P(s, d))$ of path $P(s, d)$ is the sum of the costs of the (directional) links on the path. Let $\Phi(s, d)$ be the set of all possible paths connecting s to d. The least cost path $P_{min}(s, d)$ is defined as the path whose cost is the lowest among all paths in set $\Phi(s, d)$.

B. The MCMNT Algorithm

Given a MCMR network with pre-assigned channels, the MCMNT algorithm operates by increasing the initial solution tree using least cost paths based on link costs. The heuristic works in a similar manner to the Dijkstra's [20] and Prim's

Algorithm 1 The MCMNT Algorithm

1: **Input**: $G = (V, E)$; source $s \in V$; destination set $\Delta = \{d_1, ..., d_m\} \subset V$;
2: **Output**: tree T connecting s to Δ with minimized $S(T)$; set of forwarding nodes F.
3: **Other global variables**: current set of unconnected destinations Δ_{cur}; current set of forwarding nodes F_{cur}; current tree T_{cur}.
4: **Initialization**: $\Delta_{cur} = \Delta$; $F_{cur} = \{s\}$; $T_{cur} = \{s, \emptyset\}$; compute the costs w of all directional links in E;
5: **START**
6: **while** $\Delta_{cur} \neq \emptyset$ **do**
7: $\quad P_{min}$ = NULL; {least cost path (LCP) in this round}
8: $\quad W(P_{min}) = \infty$; {cost of this path}
9: $\quad d_{min}$ = NULL; {destination of this LCP}
$\quad$ {Find an unconnected destination that can be connected to the current tree with the minimum cost.}
10: $\quad$ **for all** nodes $v \in T_{cur}$ **do**
11: $\quad\quad$ Compute the LCP connecting v to each node in Δ_{cur} using Dijkstra's algorithm.
12: $\quad\quad$ Among these LCPs, select the path $P(v, d)$ with the smallest cost, where d is some node in Δ_{cur}.
$\quad\quad$ {Keep $P(v, d)$ if it is better than current P_{min}}
13: $\quad\quad$ **if** $W(P(v, d)) < W(P_{min})$ **then**
14: $\quad\quad\quad P_{min} = P(v, d)$; $d_{min} = d$;
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
$\quad$ {d_{min} can be connected to the current tree with the minimum cost among the unconnected destinations. Add d_{min} and P_{min} to tree.}
17: $\quad T_{cur} = T_{cur} \cup$ {nodes and links on P_{min}};
18: $\quad F_{cur} = F_{cur} \cup$ {intermediate nodes on P_{min}};
19: $\quad \Delta_{cur} = \Delta_{cur} \backslash d_{min}$;
$\quad$ {Update applicable link costs to take advantage of the WBA in the next round.}
20: $\quad$ **for all** link (u, v) in P_{min} **do**
21: $\quad\quad$ {N_u denotes the set of one-hop neighbors of a node u.}
22: $\quad\quad$ **for all** $z \in N_u$ **do**
23: $\quad\quad\quad$ **if** $z \notin T_{cur}$ and $channel(u, z) = channel(u, v)$ **then**
24: $\quad\quad\quad\quad w(u, z) = 0$ {link cost set to zero}
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **end for**
27: $\quad$ **end for**
28: **end while** {terminates when all destinations are connected to T_{cur}}
29: $T = T_{cur}$; $F = F_{cur}$; **return** $[T, F]$;
30: **END**

algorithms [21] to some extent. The algorithm is summarized in **Algorithm 1** shown below. Initially, the initial solution tree consists only the source, s. Multicast destinations are then added to the tree one by one using the least cost path from each destination to the current tree (the **while** loop on line 6). In particular, for each node v in the current tree T_{cur}, we find the least cost path connecting v to each node d in the current set of unconnected destinations Δ_{cur} using the Dijkstra's algorithm (line 11). We then consider all the computed least cost paths $P(v, d)$, $\forall d \in \Delta_{cur}$, $\forall v \in T_{cur}$, and select the path P_{min} with the minimum cost (lines 13-14). This path P_{min} and the corresponding destination d_{min} are then added to the solution tree (lines 17-19).

We then update the applicable link costs to take advantage of the WBA in the next round of inserting a new destination to the tree (lines 20-27). Specifically, for each directional link (u, v) on path P_{min} just selected, and for each one-hop neighbor z of u that currently resides *outside* the tree, if link (u, z) is assigned the same channel as link (u, v), we update the cost of link (u, z) to zero (lines 23-24). By doing this, we increase the chance of link (u, z) being selected in the next round. If (u, z) is later added to the solution tree, u will be able to reach z without requiring one more transmission. These link cost updates aim at exploiting the WBA, as suggested by Wieselthier et al. [22]. The above procedure is repeated until all the destinations are added to the solution tree. The time complexity of **Algorithm 1** is $O(|\Delta||V|^2)$ as proved in [19].

C. The Distributed MCMNT Algorithm

Like the centralized algorithm, the distributed algorithm builds a multicast tree such that the source is connected to multicast destinations via the least cost paths (using link costs $w(u, v)$ defined by Eq. (1)) and takes into account both the WBA and the underlying channel assignments. Unlike the centralized algorithm, it uses local information only, i.e., a node uses only the information received from its neighbors. We assume that prior to the start of the algorithm, each node has computed the costs of its outgoing links using Eq. (1) based on the channel information exchanged with its neighbors.

The algorithm works in two phases. In the first phase, a broadcast tree containing the least cost paths from s to all the other nodes in the network is constructed. In the second phase, the resulting broadcast tree is pruned to create a multicast tree.

1) Phase I: The broadcast tree is constructed in a similar manner to the least cost path algorithm by Chandy et al. [23], which is based on distance-vector routing [24]. It uses the the link costs $w(u, v)$ defined by Eq. (1), which considers both the WBA property and channel diversity in the computations of path costs. Phase I requires two types of messages: EXPLORE and ACK. An EXPLORE message is a two-tuple (u, e_c), where u is the node sending the message and e_c is the cost of the current best path, as seen by node u, from the source to u's neighbor(s) that is (are) using channel c. After a node u broadcasts an EXPLORE message to its neighbors, it waits for ACK messages from all of its neighbors, then sends its own ACK to the node(s) from which it received an EXPLORE message. The EXPLORE and ACK messages, besides propagating path costs, ensure that all nodes in the network are covered in the process of constructing the broadcast tree.

Each node v maintains the following local variables:

- W_{best}: the cost of the current best path P_{best} from the source to v as currently seen by v.
- p: the parent node of v on the best path P_{best} from which v received the cost W_{best}.
- K_v: the set of channels that are assigned to v. For example, in Fig. 1(a), node M has $K_M = \{1, 2\}$.
- $N_v(c)$: the set of v's neighbors that communicate with v via channel c. Node M above has $N_M(1) = \{S, N, L\}$.

Phase I starts when the multicast source s broadcasts an EXPLORE message $(s, \min_{\forall u \in N_s(c)} \{w(s, u)\})$ on every channel $c \in K_s$. When a node v receives an EXPLORE message (u, e_c) from a node u via channel c, if $e_c < W_{best}$, this indicates that the path $\{s, ..., u, v\}$ is better than the path v currently buffers. Thus node v sets W_{best} to e, and p to u, and then broadcasts an EXPLORE message $(v, W_{best} + \min_{\forall z \in N_v(c)} \{w(v, z)\})$ on each channel $c \in K_v$. The min function is used to exploit the WBA property. For example, suppose there are two link costs $w(v, z)$ and $w(v, z')$ where $w(v, z) < w(v, z')$, then it is sufficient for node v to reach both neighbors z and z' at a cost of $w(v, z)$, provided that both links (v, z) and (v, z') are assigned the same channel. This is similar to the maximum power transmission technique introduced in [22]. If $e_c \geq W_{best}$, then v simply sends an ACK message to u to indicate that v has already got a path better than the path $\{s, ..., u, v\}$.

Phase I terminates when the source s has received ACK messages from all of its neighbors. Variable p of a node then indicates the parent of that node in the broadcast tree. Note that loops do not exist in the resulting broadcast tree since our link costs are positive [24]. Phase I is summarized in **Algorithm 2** below.

Algorithm 2 The Distributed MCMNT Algorithm: Phase I

if v **receives an EXPLORE message** (u, e_c) **then**
    if $e_c < W_{best}$ **then**
        sends an ACK to the current p, if it has not done so;
        $p = u$; $W_{best} = e_c$; {updates new p and W_{best};}
        {broadcasts EXPLORE messages to neighbors;}
        for all channel $c \in K_v$ **do**
            broadcasts EXPLORE $(v, W_{best} + \min_{\forall z \in N_v(c)} \{w(v, z)\})$;
        end for
    else
        sends an ACK to u;
    end if
end if
if v **receives an ACK message then**
    saves this ACK;
    if v has received ACK from all neighbors **then**
        sends an ACK to p;
    end if
end if

2) Phase II: The source uses the broadcast tree to notify all nodes to start the pruning process. Each multicast destination sends a KEEP message to its parent p. A node u, upon receiving a KEEP message from a node v, records v as its child, and then forwards the message to its parent. The forwarding process continues until the KEEP message reaches the source. Non-leaf nodes that do not receive a KEEP message after a pre-determined period, and non-destination leaf nodes will remove themselves from the tree.

V. Experimental Results

Using the QualNet simulator version 4.0 [25], we compare the performance of the centralized and distributed MCMNTs (C-MCMNT and D-MCMNT, respectively) with that of SPTs, MSTs [26] and MFTs. We simulate a medium-size MCMR network of 100 wireless routers uniformly distributed over a 1700m × 1700m area with random channel assignments. Each wireless router (node) has a transmission range of 350m. We use PHY802.11b at the physical layer with a data rate of 11 Mbits/s. The IEEE802.11 CSMA/CA protocol without RTS/CTS exchange is chosen as the multicast medium access control protocol. At the transport layer, we do not use any flow or congestion control mechanisms in order to test the network performance under very high loads. The packet size excluding the headers is 512 bytes. The multicast group has one source placed at the center sending data at a constant bit rate (CBR), while the destinations are randomly scattered around the network. (Destinations are wireless routers of the WMN backbone.) In each experiment, the source transmits at a specified CBR for 600 seconds of simulated time. Each data point in the graphs is averaged from five runs using different random seeds and plotted with a confidence interval of 95%.

For each type of tree, we measure the total number of transmissions per packet $S(T)$ (as defined in Section III), average packet delivery ratio (PDR), average throughput, and average packet end-to-end delay (averaged over all destinations) as functions of

- *multicast group size*. The number of multicast destinations varies from 20 to 80 nodes. The number of radios per node and the number of available channels are set to 3. The source transmits at a rate of 200 packets/s.
- *multicast traffic load*. The multicast source rate at the application layer varies from 100 to 300 packets/s . The number of channels is 3 and the group size is 40.
- *number of channels*. The number of channels is set to 1, 3, 5, and 7. The group consists of 40 destinations and the multicast rate is set to 200 packets/s.

The graphs in Figs. 2, 3, and 4 show that in most cases the C-MCMNT and D-MCMNT trees give similar results. In the following discussion, we treat both types of MCMNTs as one.

A. Function of Group Size

The graph in Fig. 2(a) confirms that the MCMNTs require the least numbers of transmissions, followed by the MFT, MST and SPT in that order. For example, the number of transmissions incurred by the MCMNTs in the 80-destination trees are about 22%, 42% and 42% less than by the MFT, MST and SPT, respectively. For all types of trees, as the group size increases, more forwarding nodes are added and the number of transmissions per packet goes up, as expected.

The graph in Fig. 2(b) shows that the MCMNTs offer the highest PDRs in all cases. For instance, the MCMNT PDRs are 12%, 17%, and 40% higher than those of the SPT, MST and MFT, respectively, when there are 20 destinations. The performance gap between the MCMNTs and the MST/MFT narrows down as the group size increases. However, the MCMNT PDR is still significantly higher than the PDRs of the other trees. In Fig. 2(c), a similar trend is observed for the average throughput. For example, for a group of size 20, the MCMNTs offer 15%, 19% and 85% higher throughputs than the SPT, MST and MFT, respectively. When a forwarding node n uses less channels (i.e., less transmissions) to multicast a data packet, that reduces the probability of packet collision with the packets its neighboring nodes transmitting on n's unused channels. That explains the higher PDRs of the MCMNTs.

The MCMNTs also incur the lowest end-to-end delay in most cases, as shown in Fig. 2(d). When a forwarding node uses more channels to multicast a data packet, more neighboring nodes may have to defer their transmissions if they also use the same channels, resulting in higher packet end-to-end delay. The MCMNT low end-to-end delays are a result of less channels being used by a forwarding node.

B. Function of Traffic Load

The graph in Fig. 3(a) confirms that the C-MCMNT and D-MCMNT require the least numbers of transmissions. The PDR and throughput of the MCMNTs are higher than those of the other trees in most cases, especially under high traffic loads (Fig. 3(b) and 3(c)). A lower number of transmissions enables an MCMNT to achieve better performance, as discussed earlier. The MCMNTs also have the lowest average end-to-end delays thanks to the least numbers of transmissions resulting in less contention among nodes in the routing tree (Fig. 3(d)).

C. Function of Number of Channels

In this set of experiments, we vary the number of orthogonal channels from one to seven. In general, increasing the number of channels improves the average PDRs, throughputs and end-to-end delays of all trees, as illustrated by the graphs in Fig. 4. Note that as the number of channels increases, the number of transmissions also goes up in many cases (Fig. 4(a)). More transmissions in this case, however, do not necessarily imply performance degradation, because the loads are distributed over more channels and parallel transmissions can be used with less interference. That explains the improved performance as the number of channels increases from one to seven.

When only one channel is available, the MFT has the least number of transmissions since the MFT algorithm is optimized for single-channel networks. As a result, the MFT provides the best PDR, throughput and end-to-end delay in this special case (Fig. 4(b), 4(c) and 4(d)). However, when multiple channels are used, the MCMNT algorithms produce trees with the least numbers of transmissions and, consequently, the highest PDRs and throughputs, as well as the lowest end-to-end delays. We also observe that the performance gap between the MCMNTs and the other trees narrows as the number of channels goes up. Increasing the number of channels naturally reduces interference, making the performance of the MCMNT trees less dominant. Nevertheless, the MCMNTs still offer noticeably better performance than the other trees, especially with respect to PDR and throughput.

To confirm the above results, we created several configurations by varying the node placement, multicast members, and network sizes (50 and 200 nodes). The results from these experiments are consistent with the above.

VI. Conclusion

We study the problem of building multicast routing trees with minimum numbers of transmissions in WMNs where multiple channels and multiple radios are used. The objective is to minimize interference among multicast nodes for improved performance. Our experimental results show that the proposed MCMNT algorithms perform significantly better than commonly used/cited trees such as SPTs, MSTs and MFTs in terms of PDR, end-to-end throughput and delay. Our future work on this problem is to incorporate traffic load at each node into the link and path costs for better load balancing and performance under dynamic network conditions.

References

[1] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating Optimal Multicast Trees in Wireless Multihop Networks," in *Proc. IEEE ISCC'05*, 2005, pp. 686–691.

[2] U. Nguyen and J. Xu, "Multicast Routing in Wireless Mesh Networks: Minimum Cost Trees or Shortest Path Trees?" *IEEE Commun. Mag.*, vol. 45, pp. 72–77, Nov. 2007.

[3] U. T. Nguyen, "On Multicast Routing in Wireless Mesh Networks," *Elsevier Computer Communications*, vol. 31, pp. 1385–1399, May 2008.

[4] S. Roy, D. Koutsonokolas, S. Das, and Y. C. Hu, "High-Throughput Multicast Routing Metrics in Wireless Mesh Networks," in *Proc. IEEE ICDCS'06*, 2006, pp. 48–48.

[5] H. L. Nguyen and U. T. Nguyen, "Channel Assignment for Multicast in Multi-channel Multi-radio Wireless Mesh Networks," *Wiley Journal of WCMC*, vol. 9, pp. 557–571, Apr. 2009.

[6] G. W. Zeng, B. Ding, Y. L. Xiao, and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," in *Proc. IEEE ICNP'07*, 2007, pp. 1–10.

[7] Z. Yin, Z. Li, and M. Chen, "A Novel Channel Assignment Algorithm for Multicast in Multi-radio Wireless Mesh Networks," in *Proc. IEEE ISCC'07*, 2007, pp. 283–288.

[8] C.-S. Chou, C. Chen, Y.-Y. Chen, R.-H. Jan, and C.-C. Hsieh, "Multicast Routing and Channel Assignment in Wireless Mesh Networks," in *Proc. IEEE MASS'09*, Oct. 2009, pp. 676–681.

[9] S. H. Lim, C. Kim, Y. B. Ko, and N. H. Vaidya, "An Efficient Multicasting for Multi-Channel Multi-Interface Wireless Mesh Networks," in *Proc. IEEE MILCOM'09*, 2009.

[10] H. Cheng and S. Yang, "Joint Multicast Routing and Channel Assignment in Multiradio Multichannel Wireless Mesh Networks Using Simulated Annealing," in *Springer LNCS*, 2008, pp. 370–380.

[11] H. S. Chiu, K. Yeung, and K. S. Lui, "Bandwidth-Guaranteed Multicast in Multi-Channel Multi-Interface Wireless Mesh Networks," in *Proc. IEEE ICC'09*, 2009.

[12] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Annual IEEE SECON'07*, 2007, pp. 481–490.

[13] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Networks," in *Proc. IEEE InfoCom'05*, 2005, pp. 2223–2234.

[14] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks," in *Proc. ACM SIGMOBILE'05*, 2005, pp. 58–72.

[15] J. Tang, G. Xue, and W. Zhang, "Interference-aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks," in *Proc. ACM SIGMOBILE'05*, 2005, pp. 68–77.

[16] 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 *Proc. IEEE SECON'05*, 2005, pp. 463–474.

[17] M. Marina and S. Das, "A topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks," in *Proc. IEEE Broadnets'05*, 2005, pp. 381–390.

[18] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Proc. IEEE InfoCom'06*, 2006, pp. 1–12.

[19] H. L. Nguyen and U. T. Nguyen, "Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks," in *Proc. IEEE ICUMT'09*, Oct. 2009.

[20] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerical Mathematics*, vol. 1, pp. 269–271, Dec. 1959.

[21] R. C. Prim, "Shortest Connection Networks and Some Generalisations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[22] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," in *Proc. IEEE InfoCom'00*, 2000, pp. 585–594.

[23] K. M. Chandy and J. Misra, "Distributed Computation on Graphs: Shortest Path Algorithms," *Communications of the ACM*, vol. 25, 1982.

[24] D. Bertsekas and R. Gallagher, *Data Networks*. Prentice Hall, 1991.

[25] Qualnet Simulator. [Online]. Available: http://www.qualnet.com

[26] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Mathematica Japonica*, vol. 24, 1980.

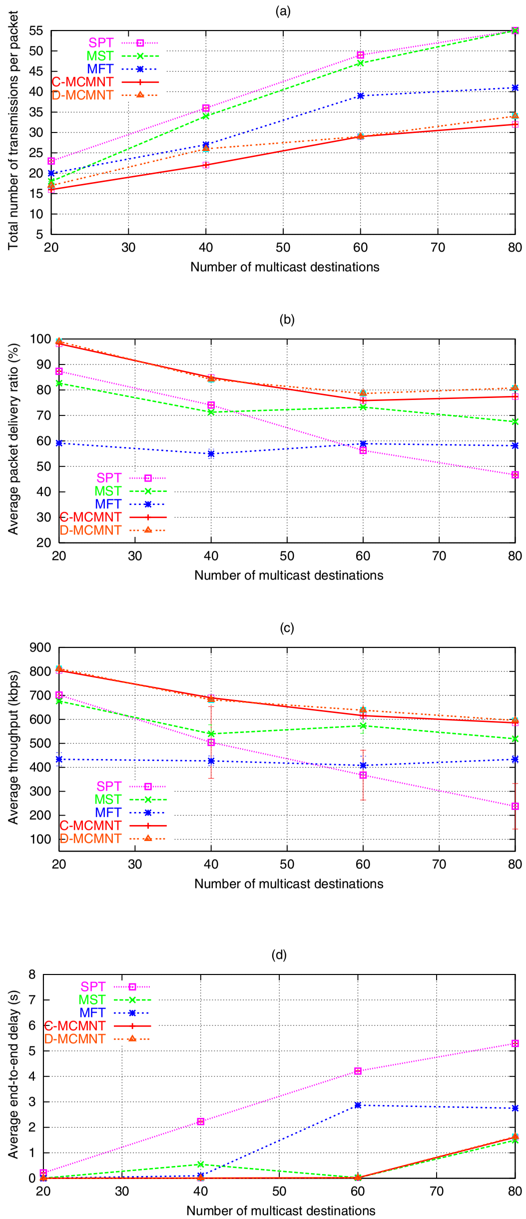


Fig. 2. Group size (3 channels; source rate 200 pkts/s)

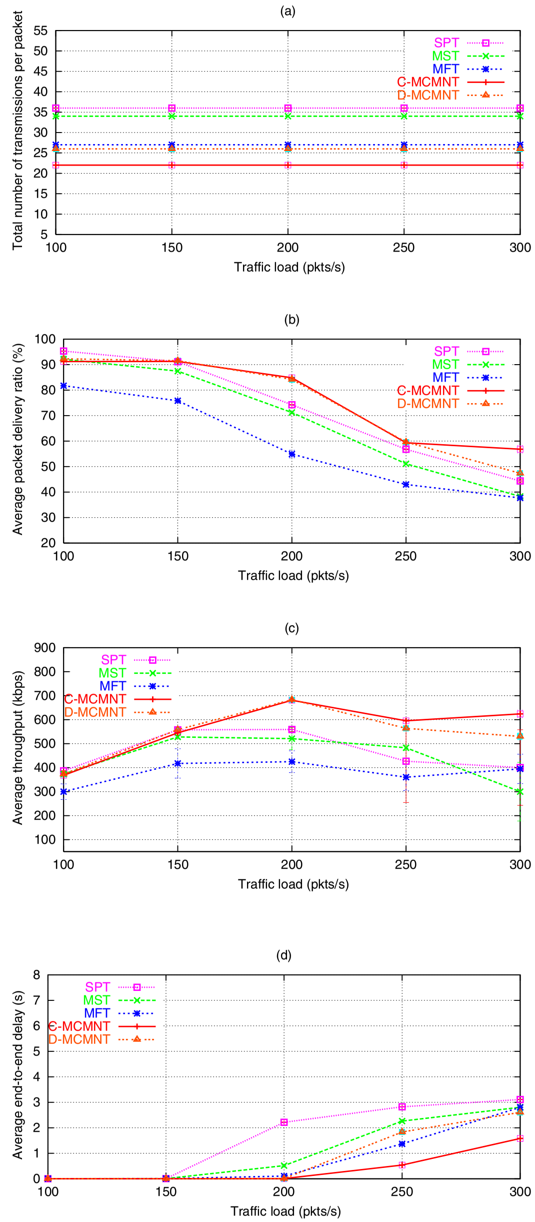


Fig. 3. Traffic load (3 channels; 40 destinations)

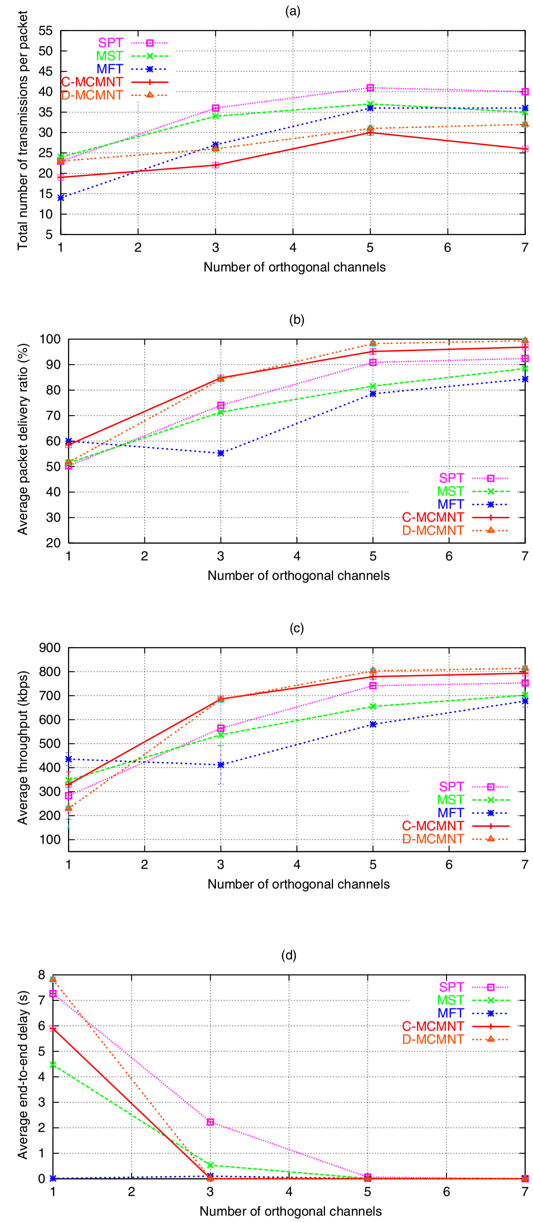


Fig. 4. Number of channels (source rate 200 pkts/s; 40 destinations; 1 radio/node for 1 channel; 3 radios/node for 3, 5, 7 channels)