11-2009

Minimum Latency Broadcasting in Multiradio, Multichannel, Multirate Wireless Meshes

Junaid QADIR
National University of Sciences and Technology, Pakistan

ChunTung CHOU
University of New South Wales

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Joo Ghee LIM
University of New South Wales

DOI: https://doi.org/10.1109/SAHCN.2006.288412

Citation

QADIR, Junaid; CHOU, ChunTung; MISRA, Archan; and LIM, Joo Ghee. Minimum Latency Broadcasting in Multiradio, Multichannel, Multirate Wireless Meshes. (2009). *IEEE Transactions on Mobile Computing*. 8, (11), 1510-1523. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/654

Published in IEEE Transactions on Mobile Computing, Volume 8, Issue 11, November 2009, Pages 1510-1523.
http://doi.org/10.1109/TMC.2009.68

Minimum Latency Broadcasting in Multiradio, Multichannel, Multirate Wireless Meshes

Junaid Qadir, Chun Tung Chou, Archan Misra, and Joo Ghee Lim

Abstract—This paper addresses the problem of "efficient" broadcast in a multiradio, multichannel, multirate wireless mesh network (MR^2-MC WMN). In such an MR^2-MC WMN, nodes are equipped with multiple radio interfaces, tuned to orthogonal channels, that can dynamically adjust their transmission rate by choosing a modulation scheme appropriate for the channel conditions. We choose "broadcast latency," defined as the maximum delay between a packet's network-wide broadcast at the source and its eventual reception at all network nodes, as the "efficiency" metric of broadcast performance. We study in this paper how the availability of multirate transmission capability and multiple radio interfaces tuned to orthogonal channels in MR^2-MC WMN nodes can be exploited, in addition to the medium's "wireless broadcast advantage" (WBA), to improve the "broadcast latency" performance. In this paper, we present four heuristic solutions to our considered problem. We present detailed simulation results for these algorithms for an idealized scheduler, as well as for a practical 802.11-based scheduler. We also study the effect of channel assignment on broadcast performance and show that channel assignment can affect the broadcast performance substantially. More importantly, we show that a channel assignment that performs well for unicast does not necessarily perform well for broadcast/multicast.

Index Terms—Routing, broadcasting, wireless mesh networks, multiradio multichannel, multirate.

1 INTRODUCTION

Wireless mesh networks (WMN) [1], where potentially-mobile mesh clients connect over a relatively-static multihop wireless network of mesh routers, are viewed as a promising broadband access infrastructure in both urban and rural environments [2]. However, the relatively low spatial reuse of a single radio channel in multihop wireless environments (due to wireless interference) poses an impediment to the widespread adoption of WMN as a viable access technology. It has been shown that network capacity drops off as the number of nodes is increased in single-channel wireless networks [3]. With recent advancements in wireless technology rendering the usage of multiple radios affordable, a popular current trend is to equip mesh nodes with multiple radios, each tuned to a distinct orthogonal channel. The usage of multiple radios can significantly improve the capacity of the network by permitting an increased number of concurrent transmissions in the network [4], [5], [6]. Another feature widely available in commodity wireless cards, which are envisioned to connect the wireless mesh nodes, is the ability to transmit at multiple transmission rates. WMN nodes can utilize the flexibility of multirate transmissions to make appropriate range and throughput/latency tradeoff choices across a wide range of channel conditions. While this flexibility has traditionally been used only for unicast, it has recently been proposed for use in broadcasting scenarios as well [7], [8]. In the near future, multiradio, multichannel, multirate (MR^2-MC) WMNs are expected to gain a niche in the wireless market due to adoption and support from leading industry vendors [9] and active research by the research community.

In this paper, we address an important open question related to MR^2-MC WMNs: how can we support "*efficient*" broadcast in such networks? We gauge this efficiency in terms of "*broadcast latency*," which we define as the maximum delay between the transmission of a packet by the source node and its eventual reception by all receivers. The minimum latency broadcasting (*MLB*) problem is particularly challenging in MR^2-MC meshes due to a myriad of complex, intertwined decisions that need to be made. Kyasanur and Vaidya have hinted in their work [6] about some of the potential problems related to routing of broadcast traffic, vis-a-vis channel assignment, in multiradio meshes.

The MLB problem, apart from its theoretical significance, is an important practical problem in WMN. There are several emerging multicast-oriented multiparty applications—such as managed software updates, local content distribution (e.g., video feeds) within communities, and multimedia gaming—that require stringent latency bounds for effective operation. Studying the impact of channel and rate diversity for network-wide broadcasts serves as the first step for devising practical network-layer multicasting in wireless meshes, as both problems require the use of an underlying link-layer multicast capability. The MLB problem has been studied for "*single-radio single-channel*" (SR-SC) wireless

---

- *J. Qadir is with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), H-12 Sector, Islamabad, Pakistan. E-mail: junaid.qadir@niit.edu.pk.*
- *C.T. Chou is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia. E-mail: ctchou@cse.unsw.edu.au.*
- *A. Misra is with the Applied Research, Telcordia Technologies, RRC1B-321, 1 Telcordia Drive, Piscataway, NJ 08854. E-mail: archan@research.telcordia.com.*
- *J.G. Lim is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia. E-mail: jool@cse.unsw.edu.au.*

networks, both for the single-rate [10] and the multirate case [7], [8]. To the best of our knowledge, this is the first work to address the issues of rate and channel diversity for broadcasting in an MR^2-MC WMN, assuming both a theoretical centralizer scheduler and a practical 802.11-based distributed MAC. We shall show that the MLB problem for MR^2-MC meshes is a more complex problem than for SR-SC multirate meshes (single-radio meshes are a special case of multiradio meshes). The differences between *single-rate* and *multirate* MLB problem, for the case of *SR-SC* meshes, are demonstrated in [7], [11] and the complexity of each problem is proven NP-hard in [7], [10], [11], respectively.

The rest of the paper is organized as follows: The background and related work is presented in Section 2. The network model and problem statement are defined in Section 3. We present four heuristic MLB solutions in Section 4. We then present performance results of our heuristics, assuming idealized and IEEE 802.11 MAC schedulers, for a single broadcasted packet in Sections 6 and 7, respectively. Thereafter, we evaluate the performance of our heuristics for a stream of broadcast packets assuming IEEE 802.11 MAC scheduler in Section 8. We finally conclude our work in Section 9.

2 BACKGROUND AND RELATED WORK

Broadcasting in wireless networks is a fundamentally different problem to broadcasting in wired networks due to the "*wireless broadcast advantage*" (WBA) [12]. The WBA arises due to the broadcast nature of the wireless channel where—assuming omnidirectional antennas are being used—a transmission by a node can be received by all neighboring nodes that lie within its communication range. This is quite different from wired networks where the cost to reach two neighbors is generally the sum of the costs to reach them individually. A lot of research has focused on achieving "*efficient*" broadcast in multihop wireless networks and mobile ad hoc networks. The metrics typically used are energy consumption [12], [13], the number of transmissions [14], [15], or the overhead in route discovery and management [16]. The limited work done for the broadcast latency metric has focused only on SR-SC networks [7], [10], [11].

Our current work builds upon our previous work on minimizing broadcast latency in an SR-SC *multirate* WMN [7], [8], where we introduced the new concept of link-layer *multirate* multicast, in which a node can adjust its link-layer multicast transmission rate to its neighbors. We showed that multicast in a multirate WMN has two features not found in a single-rate WMN. First, if a node has to perform a link-layer multicast to reach a number of neighbors, then its transmission rate is limited by the smallest rate on each individual link, e.g., if a node n is to multicast to two neighboring nodes m_1 and m_2, and if the maximum unicast rates from n to m_1 and m_2 are, respectively, r_1 and r_2, then the maximum rate n can use is the minimum of r_1 and r_2. Consequently, the choice of child nodes in a routing tree is coupled with the transmission rate that may be employed at the link layer. Second, for a multirate WMN, the broadcast latency can be minimized by exploiting an extra degree-of-freedom where some nodes transmit the same packet more than once, but at a different rate to different subsets of neighbors (called as "*distinct-rate transmissions*"). Based on these insights, we presented the "*weighted connected dominating set*" (WCDS) and "*bandwidth incremental broadcast*" (BIB) algorithms as heuristic solutions for the MLB problem in SR-SC multirate mesh networks. Both these algorithms consider the WBA and the multirate capability of the network, and also incorporate the possibility of multiple *distinct-rate* transmissions by a single node. Details of these algorithms are available at [7], [11].

The assignment of channels in MR^2-MC WMNs plays a very important part in determining the actual performance of the network. Generally, there are two conflicting objectives for any channel assignment protocol: while the routing layer usually benefits from increased "connectivity" between nodes, transport-layer protocols usually benefit from a reduction in the "interference" between neighboring nodes. Channel assignment strategies can be broadly classified as *static*, *dynamic*, and *hybrid* schemes [6]. For our current work, we do not consider dynamic and hybrid schemes, due to the nonnegligible interface switching delay and synchronization requirements involved in such strategies.

Among the static channel assignment strategies, the simplest approach is the "*common channel approach*" (CCA) (e.g., [4]), in which all nodes are assigned a common set of channels. The benefit of this approach is its simplicity and that the connectivity of the network is a multiple of the connectivity of a single-channel mesh. In an alternative approach called "*varying channel approach*" (VCA), interfaces of different nodes may be assigned to a different set of channels (e.g., [5]). With this approach, there is a possibility of a network partition, unless the interface assignment is done carefully. In yet another approach called "*interference survivable topology control*" (INSTC) [17], the channel assignment is made such that the induced network topology is interference-minimum among all k-connected topologies.

3 NETWORK AND INTERFERENCE MODEL

Our mathematical notation, which is similar to that used in [17], is tabulated in Table 2 for easy reference. We assume the system has a total of C nonoverlapping orthogonal frequency channels and each node is equipped with Q radio interfaces where $Q \leq C$. The Q radio interfaces have omnidirectional antennas, and a unit disk graph model is assumed (although the algorithms can accommodate any arbitrary rate-range relationships between node pairs). In order to efficiently utilize the network resources, two radio interfaces at the same node are not tuned to the same channel. Each node can transmit at multiple rates. Using the Qualnet simulator [18] as a reference, we obtain the transmission rate versus transmission range relationship for 802.11a and 802.11b in Table 1, assuming a two-ray propagation model. Note also that the interference range in Qualnet is 520 m. The transmission range is a decreasing function of transmission rate as Table 1 illustrates.

Our framework assumes that channel assignment has already been performed. We model an MR^2-MC WMN as a graph $G = (V, E, l, \mathcal{A})$, where V, E, l, and $\mathcal{A}$ are, respectively, the set of nodes, the set of links, a function that defines the latency of each link, and the channel assignment

TABLE 1
The Rate-Range Relationship of 802.11a and 802.11b

802.11a Rate (Mbps)	802.11a Range (m)	802.11b Rate (Mbps)	802.11b Range (m)
9	369	1	483
18	261	2	370
36	130	5.5	351
54	37	11	283

function. The channel assignment function $\mathcal{A}$ assigns each vertex $v \in V, Q$ different channels denoted by the set: $A(v) = \{a_1(v), a_2(v), \ldots, a_Q(v) : a_i(v) \neq a_j(v), \forall i \neq j;\ a_i(v) \in \mathcal{C}, \forall i\}$, where $a_i(v)$ is the channel assigned to the ith radio interface at node v and $\mathcal{C}$ denotes the set of all channels with $|\mathcal{C}| = C$. Each edge $e \in E$ is represented by a 3-tuple (u, v, k), where $u, v \in V$ and $k \in \mathcal{C}$. Specifically, there is an edge $e = (u, v, k) \in E$ between nodes $u, v \in V$ on channel k if and only if the euclidean distance $d(u, v)$ between u and v is less than or equal to the maximum transmission range and $k \in \mathcal{A}(u) \bigcap \mathcal{A}(v)$. *Note that G may be a multigraph, with multiple edges between the same pair of nodes, when the node pair shares two or more channels.* The function $l(e)$, which maps E to the set of all possible latencies $\mathcal{L}$, is the minimum latency (or highest rate) transmission supported on the edge $e \in E$. For simplification, we do not consider contention and queuing delays, and assume that the latency $l(e)$ of a link e is inverse of the largest transmission rate supported on link e.[1]

As an illustrative example, consider the WMN in Fig. 1. The WMN consisting of four nodes $\{w, x, y, z\} = V$ (shown as oval), with each node equipped with $(Q =)2802.11\mathrm{b}$ radio interfaces (shown as rectangles). Since 8021.11b has four transmission rates (see Table 1), the set of link latencies $\mathcal{L}$ is set to $\{1, \frac{1}{2}, \frac{1}{5.5}, \frac{1}{11}\}$. There are $(C{=})$ 3 channels with $\mathcal{C} = \{1, 2, 3\}$. The channel assignment for each node is defined by $\mathcal{A}(w) = \{1, 2\}$, $\mathcal{A}(x) = \{1, 2\}$, $\mathcal{A}(y) = \{2, 3\}$, and $\mathcal{A}(z) = \{1, 3\}$. The euclidean distances between the nodes are given by $d(w, x) = d(y, z) = 300m$, $d(w, z) = d(x, y) = 400m$, and $d(w, y) = d(x, z) = 500m$. Given the transmission range of 802.11b is 483 m (see Table 1) and the channel assignment $\mathcal{A}$, the edges in the WMN are $(w, x, 1)$, $(w, x, 2)$, $(w, z, 1)$, $(x, y, 2)$, and $(y, z, 3)$. Note that there are two edges connecting nodes w and x. Since $d(w, x) = d(y, z) = 300m$, the maximum 802.11b transmission rate that can be supported on the links $(w, x, 1)$, $(w, x, 2)$, and $(y, z, 3)$, according to Table 1, is 5.5 Mbps. Therefore, $l(w, x, 1) = l(w, x, 2) = l(y, z, 3) = \frac{1}{5.5}$. Similarly, $l(w, z, 1) = l(x, y, 2) = 1$.

Channel assignment algorithms. We assume that the channel assignment is done independently from our broadcasting framework. This design decision reflects the practical reality that the channel assignment strategy will likely be dictated by other factors, including the presence of *unicast* traffic on the WMN. We have used the following three *static* channel assignment strategies: CCA, VCA, and INSTC. For CCA, dedicated interfaces are allocated for the *same* Q channels at every node; therefore, only Q channels are used in the network when using CCA. In VCA, an interface at all nodes is allocated the same channel to ensure a connected network; for the remaining $Q - 1$ interfaces, channels are chosen randomly from the remaining $C - 1$ channels. The last channel assignment scheme used is INSTC, which we use to construct 1-connected topologies.

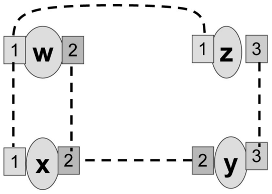


Fig. 1. Example to illustrate our network notation.

Link-layer multicast transmissions. In order to exploit WBA, we use link-layer multicast transmissions (instead of unicast transmissions) to realize network-wide broadcast. We specify a link-layer multicast transmission (or, simply, transmission when the context is clear) b as a 4-tuple $(p, I, \bar{l}, k)$, where the node $p \in V$ is to transmit to all the nodes in the set $I \subseteq V$ in a link-layer multicast using latency $\bar{l}$ on channel k. In order that the transmission b is physically realizable on the topology $G = (V, E, l, \mathcal{A})$, we require 1) $(p, i, k) \in E$ and 2) $\bar{l} \geq \max\{l(p, i, k)\}, \forall i \in I$; intuitively, the node p must be able to reach the nodes in I using latency $\bar{l}$ and all nodes in $\{p\} \cup I$ must have an interface tuned to channel k. Note that the link-layer multicast transmission m can also be viewed as a collection of links $\cup_{i \in I}\{(p, i, k)\}$. Based on the earlier example, $(w, \{x, z\}, 1, 1)$ is a realizable transmission and it can be viewed as the collection of links $\{(w, x, 1), (w, z, 1)\}$.

Interference Model. We use a generalized conflict graph *based on link-layer multicast transmissions* to model the effects of wireless interference in MR^2-MC meshes. The conflict graph indicates which transmissions mutually interfere, and hence, cannot be active simultaneously. A transmission b_i interferes with a transmission b_j, if both transmissions b_i and b_j are taking place on the *same* channel, *and* the receivers of the transmission b_i are within the interference range of the transmitting node of b_j or vice versa. The transmissions b_i and b_j do not interfere otherwise.

Problem statement. The MLB problem is defined as follows: Given the graph $G = (V, E, l, \mathcal{A})$, and the broadcast source node s, the objective is to find a set of link-layer multicast transmissions such that: 1) The union of all links from all the transmissions *forms a spanning tree* on G (note that we are viewing each transmission as a collection of links) and 2) The broadcast latency—which is defined as the maximum delay (ensuring no scheduling conflicts) between the transmission of a packet by the source node and its eventual reception by all receivers while ensuring no scheduling conflicts—*is minimized*. More formally, the broadcast latency for any receiving node equals the sum of the transmission latencies on the path (in the spanning tree) from the source to this node (and thus, ignores other constraints, e.g., channel contention delays).

As noted in [7], [11], the MLB problem is NP-hard for SR-SC WMNs, and by extension, for MR^2-MC WMNs as well. Accordingly, our focus in this paper is on the design of specific heuristics.

---

[1] The proposed heuristics, however, are general and will optimize the maximum broadcast latency based on whatever latency values are input.

TABLE 2
Table of Frequently Used Mathematical Symbols

V	Set of vertices (or, nodes)	E	Set of edges (or, links)
Q	Number of radio interfaces at each node	C	Total number of orthogonal channels
N	Total number of nodes in network	$\mathcal{A}(v)$	Set of channels assigned to node v
$\mathcal{L}$	Set of distinct latencies supported by MAC	$a_i(v)$	Channel assigned to i^{th} interface of node v
$N(n,l,c)$	n's neighbors when it transmits on channel c with latency l	$f(n,l,c)$	Quality metric of n's transmission on channel c with latency l
(u,v,k)	an edge as a 3-tuple from node u to node v on channel k	(p,I,l,k)	link-layer multicast transmission from node p to the nodes in the set I on channel k using latency $\bar{l}$

4 HEURISTICS TO CONSTRUCT MLB TREE IN MR^2-MC MULTIRATE MESH

Our MLB heuristics aim to compute a broadcast tree (or a spanning tree) that exploits the WBA, the multirate transmission capability, *and the plurality of radio interfaces and channels available*. The transmitting nodes, their interfaces used for transmissions, and the children/parent relationships between different nodes are all decided at this stage. We assume that each node will transmit only once on a particular channel, i.e., it will not perform multiple distinct-rate transmissions on the same channel. This follows from our earlier study which showed that the performance improvement using muliple distinct-rate transmissions is not significant [7].

We present four MLB heuristic algorithms; the first (Section 4.1) does not exploit the WBA, the second (Section 4.2) exploits WBA but not the availability of multiple interfaces on the same node, while the other two (Sections 4.3 and 4.4) differ in how they exploit both WBA and the interface diversity on individual nodes. The common input to all our MLB heuristic algorithms is the network topology $G = (V, E, l, \mathcal{A})$, broadcast source $s \in V$, and the set of latencies $\mathcal{L}$. The common output of all the heuristics is the set of all link-layer multicast transmissions $B = \{b_1, b_2, \ldots\}$ with the property that the collection of all the edges in b_i forms a spanning tree. Note again that each transmission b_i is a 4-tuple which specifies its transmitting node, multicast recipients, latency, and channel.

4.1 Multiradio, Multichannel Shortest Path Tree (MSPT)

The MSPT algorithm (see Algorithm 1) is used to construct the SPT for MR^2-MC WMNs. The MSPT algorithm is very similar to the Dijkstra algorithm, except that it also has to choose appropriate links since a pair of nodes can have multiple links between them on different channels. We include this algorithm mainly as a baseline, to illustrate the comparative performance of routing strategies that do not exploit the unique characteristics of either the wireless medium or multichannel WMNs.

Algorithm 1. MSPT construction

```
1: Input: $[s, \mathcal{C}, G = (V, E, l, \mathcal{A}), \mathcal{L}]$
2: Initialize label $(v) = \infty, \forall v \in V; \mathcal{I}(v,c) = \emptyset \; \forall v \in V,$
     $c \in \mathcal{A}(v); d = s; \mathcal{R} = V \backslash \{s\}; B_{MSPT} = \emptyset$
3: while $(V \backslash \mathcal{R} \neq \emptyset)$ do
4:     $N(d)$ = neighbors of $d$;
5:     for $i \in N(d)$ do
6:         $label(i) = \min(label(d) + l(d,i), label(i))$;
7:     end for
8:     $d_{perm} = \arg\min_{i \in \mathcal{R}} label(i)$
9:     Let $d$ and $d_{perm}$ be connected by channels $\{c_1, \ldots, c_j, \ldots\}$
         and let $f(c_j) = \text{number}$ of existing multicast
         transmissions that interferes with the transmission
         from $d$ to $d_{perm}$ on channel $c_j$. Let $\hat{c}$ be the channel that
         minimizes $f$.
10:    $\mathcal{I}(d,\hat{c}) = \mathcal{I}(d,\hat{c}) \cup \{d_{perm}\}$
11:    $d = d_{perm}; \mathcal{R} = \mathcal{R} \backslash \{d\}$
12: end while
13: for $v \in V, c \in \mathcal{A}(v)$ such that $\mathcal{I}(v,c) \neq \emptyset$ do
14:     $\hat{l} = \max_{i \in \mathcal{I}(v,c)} l(v,i,c)$
15:     $B_{MSPT} = B_{MSPT} \cup \{(v, \mathcal{I}(v,c), \hat{l}, c)\}$
16: end for
17: Output: $B_{MSPT}$
```

Algorithm. The MSPT algorithm is based on the principle of label setting, with a single label becoming permanent in each iteration. Like Dijkstra's algorithm, the label of a node represents the "*cost*" of its *current* shortest path to the source s. The set $\mathcal{R}$ represents the nodes, whose shortest path to s have not been finalized yet. The basic operation of MSPT is edge relaxation (line 6) where $l(d,i)$ is the latency between nodes d and i. Note that although the nodes d and i may be connected over multiple channels, the latency between them is the same over all the channels connecting them. Therefore, we have abused the notation and used $l(d,i)$ in line 6 instead of $l(d,i,k)$, where k indicates the channel.

In each iteration, the label of the node d_{perm} becomes permanent, implying that node d will be the parent node for node d_{perm} in the spanning tree. If there are multiple common channels connecting d and d_{perm} (which also means multiple links between them), we choose the link whose channel is the "*least-used*" in the conflict graph of this transmission. Our notion of "least used channel"—scoped to the broadcast tree itself, and to the transmission of a singe packet—utilizes the concept of a transmission-based conflict graph (as explained in Section 3): it is the channel with the smallest number of "conflicting" transmissions, with ties broken randomly.

When all labels become permanent (after line 12), the set $\mathcal{I}(v,c)$ contains all the downstream nodes of node v over channel c in the broadcast tree. Since node v can reach all of the nodes in $\mathcal{I}(v,c)$ by only one link-layer multicast transmission, line 14 determines the smallest latency (the largest of the latencies of each individual link) to realize this. Finally, note that *MSPT does not use WBA to determine the broadcast tree but simply group downstream nodes in a transmission for convenience after the tree is computed.*

Complexity. We first consider the complexity of the MSPT algorithm if an array (or linked list) is used to store the *label*s. The search for the minimum label will take $O(|V\backslash\mathcal{R}|)$. In addition, each run of line 9 takes $O(Q|V\backslash\mathcal{R}|)$. Since the **while**-loop iterates $(N-1)$ times and $|V\backslash\mathcal{R}|$ increases by one in each loop, the overall complexity is $O(N^2+QN^2)=O(QN^2)$.

If a Fibonacci heap is used to store the labels instead, then the overall complexity of the MSPT becomes $O(M+N\log(N)+QN^2)$, where M is the number of edges in the network. Note that the $O(QN^2)$ part in the complexity expression again comes from the execution line 9. For a wireless network with N nodes uniformly distributed in an area A, the average number of neighbors that a node has is $\frac{\pi R^2}{A}N$, where R is the transmission range. Therefore, for a WMN, where each node has Q interfaces, the number of edges M is $O(QN^2)$. (Note that our setting assumes that the transmission range is not adjustable.) Hence, a more complex implementation based on Fibonacci heap does not yield lower complexity because of the need to choose channels. We conclude, therefore, that *the complexity of MSPT is* $O(QN^2)$.

4.2 Multiradio, Multichannel WCDS Tree (MWT)

The MWT algorithm (see Algorithm 2) is a relatively simple multichannel extension to the WCDS algorithm proposed in [7], [11] for the MLB problem for SR-SC multirate networks. It is included here for comparative purposes, to benchmark the capabiltiies of wireless and rate diversity-aware techniques that are *not* specifically tuned for multichannel WMNs. In SR-SC multirate WMNs, WCDS performs creditably against other low-latency broadcast heuristics, because WCDS considers both: the multirate nature of the network and the WBA of the underlying wireless medium. The MWT, like WCDS, is a greedy heuristic algorithm that decides the "*best*" transmission in each round, from a set of eligible transmissions. However, as we shall see, *MWT does not consider the availability of multiple interfaces on each node, and thus, fails to exploit the potential advantage of parallel transmissions at any intermediate node.*

Algorithm 2. MWT construction

```
1: Input: $[s, C, G=(V,E,l,\mathcal{A}), \mathcal{L}]$
2: $\mathcal{R} \leftarrow \{s\}$. $B_{MWT}=\emptyset$
3: while ($V\backslash\mathcal{R} \neq \emptyset$) do
4:   $(\hat{n},\hat{l},\hat{c}) = \arg\max_{n\in\mathcal{R}, l\in\mathcal{L}, c\in\mathcal{A}(n)} f(n,l,c)$
5:   (where $f(n,l,c) = (|N(n,l,c)\backslash\mathcal{R}| \div l)$)
6:   {if multiple $(\hat{n},\hat{l},\hat{c})$ with max $f$, choose whose
7:   $\hat{c}$ is least used in the conflict graph of $(\hat{n},\hat{l},\hat{c})$ }
8:   $NYC \leftarrow N(\hat{n},\hat{l},\hat{c})\backslash\mathcal{R}$;
9:   $\mathcal{R} \leftarrow \mathcal{R} \cup NYC$
10:  if $\hat{n}$ already transmitting on channel $\hat{c}$ with latency $\tilde{l}$
       to the nodes in $\tilde{I}$ (i.e., $(\hat{n},\tilde{I},\tilde{l},\hat{c}) \in B_{MWT}$) then
11:    $\bar{l} = max(\hat{l},\tilde{l})$; $I = NYC \cup \tilde{I}$;
12:    $B_{MWT} = (B_{MWT}\backslash\{(\hat{n},\tilde{I},\tilde{l},\hat{c})\}) \cup \{(\hat{n},I,\bar{l},\hat{c})\}$
13:  else
14:    $B_{MWT} = B_{MWT} \cup \{(\hat{n}, NYC, \hat{l}, \hat{c})\}$
15:  end if
16: end while
17: Output: $B_{MWT}$
```

Algorithm. The algorithm starts by making the source node s eligible to transmit. This is done by moving s to the set $\mathcal{R}$ which keeps track of the eligible nodes (nodes that have received the transmission already and are eligible to transmit). We say that a node is covered and is eligible for transmission if it is in the set $\mathcal{R}$. We refer to (n,l,c) as a "*combination*" or as a "*transmission combination*," and define it as the transmission by an eligible node $n\in\mathcal{R}$, with latency $l\in\mathcal{L}$, on channel $c\in\mathcal{A}(n)$. We use the term $N(n,l,c)$ to refer to all neighbors of n which are reachable by the transmission combination (n,l,c). For any transmission combination (n,l,c)—the quantity $|N(n,l,c)\backslash\mathcal{R}|$ (also represented as NYC in Algorithm 2) is the number of "not-yet-covered nodes" reachable by this transmission combination.

All eligible combinations $(\forall n\in\mathcal{R}, \forall l\in\mathcal{L}, \forall c\in\mathcal{A}(n))$ are given a "priority" measure defined as the product of "not-yet-covered nodes" and the rate of transmission, i.e., $\frac{1}{l}$, or as $|N(n,l,c)\backslash\mathcal{R}| \div l$. The priority reflects the desire to both cover as many nodes as possible in a single transmission, yet keep the transmission rate high.

In each round of the algorithm, the node with maximum "priority" is selected. In case of multiple combinations (n,l,c) having the same priority, the combination transmitting on the channel $\hat{c}$, the least loaded channel within the conflict graph of the transmission is chosen (as explained in Section 4.1). The algorithm completes its execution when all the nodes have been covered, i.e., when $V\backslash\mathcal{R}=\emptyset$. The algorithm returns the set B_{MWT} whose collection of link-layer multicast transmissions forms the MWT.

Complexity. The MWT algorithm will require at most $N-1$ rounds. In the worst case, each round requires the computation of: 1) $f(n,l,c)$ and 2) the number of transmissions interfering the transmission combination (n,l,c) for breaking ties. (Note: Ties will need to be broken in each round in the worst case.) These two quantities can be computed by scanning through all the nodes in, respectively, the transmission and interference ranges of (n,l,c). Let N_t and N_i denote, respectively, the number of nodes in the transmission and interference range of each node. Therefore, each computation of 1) and 2) requires $O(N_t)$ and $O(N_i)$, respectively. Since the number of transmission combinations in each round is $|\mathcal{R}||\mathcal{L}|Q$ (Note: Each node n has $|\mathcal{L}|Q$, not $|\mathcal{L}|C$, different transmission combinations.) with $|\mathcal{R}|<N$; therefore, the complexity of MWT is $O(N^2|\mathcal{L}|Q(N_i+N_t))$. Note that the above complexity analysis is derived assuming that a linked list is used to store $f(n,l,c)$.

4.3 Locally Parallelized, Multiradio, Multichannel WCDS Tree (LMT)

LMT is our first proposed algorithm that is designed to better exploit the interface diversity on individual nodes in MR^2-MC WMNs. It tries to rectify MWT's inherent bias (via its choice of the priority metric) in favor of transmissions that cover greater number of uncovered nodes. This metric works well when the number of radio interfaces/channels is small. However, it fails to exploit the increased opportunities for parallel "faster" transmissions (on different orthogonal channels) when the number of interfaces is higher.

Accordingly, the LMT algorithm is based on the observation that a node m covered by a transmission combination (n, l, c) may also be covered by combination $(n, \hat{l}, \hat{c})$, where $l > \hat{l}$ and $c \neq \hat{c}$. Thus, we may be able to cover node m for free on an orthogonal channel $\hat{c}$ without paying a delay penalty. This is done by considering node m as a covered node of $(n, \hat{l}, \hat{c})$ but not (n, l, c). More importantly, eliminating m as a child of (n, l, c) opens up the possibility of improving the latency l of the transmission (n, l, c), in case the link latencies for the remaining candidate nodes are lower.

Algorithm 3. LMT construction

```
Input: $[s, C, G = (V, E, l, \mathcal{A}), \mathcal{L}]$
$\mathcal{R} = \{s\}$. $B_{LMT} = \emptyset$
while ($V \backslash \mathcal{R} \neq \emptyset$) do
    $(\hat{n}, \hat{l}, \hat{c}) = \arg\max_{n \in \mathcal{R}, l \in \mathcal{L}, c \in C} f(n, l, c)$
    {where $f(n, l, c) = (|N(n, l, c) \backslash \{\mathcal{R} \bigcup RN_{(n,l,c)}\}| \div l)$
    and $RN_{(n,l,c)} = \cup_{\forall (l_i \in \mathcal{L}) < l, \forall (c_i \in (\mathcal{A}(n) \backslash \{c\}))} N(n, l_i, c_i)\}$
    {if multiple $(\hat{n}, \hat{l}, \hat{c})$ with max $f$, choose whose
    $\hat{c}$ is least used in conflict graph of $(\hat{n}, \hat{l}, \hat{c})$}
    $N_{covered} = N(\hat{n}, \hat{l}, \hat{c}) \backslash \{\mathcal{R} \cup RN_{(\hat{n}, \hat{l}, \hat{c})}\}$
    $NYC \leftarrow N_{covered}$;
    $\mathcal{R} \leftarrow \mathcal{R} \cup NYC$
    if $\hat{n}$ already transmitting on channel $\hat{c}$ with latency $\tilde{l}$ to
      the nodes in $\tilde{I}$ (i.e., $(\hat{n}, \tilde{I}, \tilde{l}, \hat{c}) \in B_{LMT}$) then
        $\bar{l} = max(\hat{l}, \tilde{l})$; $I = NYC \cup \tilde{I}$;
        $B_{LMT} = (B_{LMT} \backslash \{(\hat{n}, \tilde{I}, \tilde{l}, \hat{c})\}) \cup \{(\hat{n}, I, \bar{l}, \hat{c})\}$
    else
        $B_{LMT} = B_{LMT} \cup \{(\hat{n}, NYC, \hat{l}, \hat{c})\}$
    end if
end while
Output: $B_{LMT}$
```

Algorithm. The LMT algorithm is identical to MWT, except in the calculation of the priorities of eligible transmissions at each round. In MWT, the "*best*" transmission in any particular round is the transmission (n, l, c) with maximum $f(n, l, c) = (|\text{neigh covered}| \div l)$ where "neigh covered" is $(N(n, l, c) \backslash \mathcal{R})$. In LMT, the term "neigh covered" is redefined to be $N(n, l, c) \backslash \{\mathcal{R} \cup RN_{(n,l,c)}\}$, where the set $RN_{(n,l,c)}$ contains all nodes that n can cover in *parallel*, at a lower latency than l, on a channel different than c of the (n, l, c) combination.

The nodes covered in each round are added to $\mathcal{R}$, which contain nodes eligible to transmit during the next round. Unlike MWT, where all noncovered neighboring nodes $N(\hat{n}, \hat{l}, \hat{c}) \backslash \mathcal{R}$ of the chosen transmission $(\hat{n}, \hat{l}, \hat{c})$ are added to $\mathcal{R}$; in LMT, only the nodes in $N(\hat{n}, \hat{l}, \hat{c}) \backslash \{\mathcal{R} \cup RN_{(\hat{n}, \hat{l}, \hat{c})}\}$ are added.

The algorithm completes its execution when all the nodes have been covered, i.e., when $V \backslash \mathcal{R} = \emptyset$. The algorithm returns the set B_{LMT} whose collection of link-layer multicast transmissions forms the LMT.

Complexity. The complexity of LMT can be shown to be the same as that of MWT, which is $O(N^2 |\mathcal{L}| Q(N_i + N_t))$ Note that the sets $RN_{(n,l,c)}$ can be computed once in the beginning with complexity $O(N |\mathcal{L}| Q N_t)$, which is dominated by other computations in the algorithm.

4.4 Parallelized, Approximate-Shortest, Multiradio, Multichannel WCDS Tree (PAMT)

The PAMT algorithm is also based on the MWT algorithm, and is designed to be *adaptive* to the number of radio interfaces and channels available. PAMT tries to achieve a balance between Dijsktra's shortest path algorithm (which is ideal if nodes have an infinite number of radios and infinite channels, but which can cause a large number of contending transmissions when resources are more scarce) and the LMT algorithm (which tries to reduce the number of contending transmissions, but does not as readily exploit the possibility of concurrent noninterfering transmissions when the numbers of interfaces and orthogonal channels are higher). In particular, PAMT is designed to improve LMT, which, during any particular round, might decide to cover some nodes with a transmission that has a longer latency path to s (the source node) compared to other eligible transmissions (*by currently unused interfaces on other intermediate nodes*) that can take place concurrently on an alternative, noninterfering channel. The following simple example illustrates the basic idea.

Algorithm 4. PAMT construction

```
Input: $[s, C, G = (V, E, l, \mathcal{A}), \mathcal{L}]$
$\mathcal{R} = \{s\}$; $label(s) = 0$; $B_{PAMT} = \emptyset$
while ($V \backslash \mathcal{R} \neq \emptyset$) do
    $(\hat{n}, \hat{l}, \hat{c}) = \arg\max_{n \in \mathcal{R}, l \in \mathcal{L}, c \in \mathcal{A}(n)} f(n, l, c)$
    {if multiple $(\hat{n}, \hat{l}, \hat{c})$ with max $f$, choose whose
    $\hat{c}$ is least used in conflict graph of $(\hat{n}, \hat{l}, \hat{c})$}
    where $f(n, l, c)$ is calculated as:
        $X = Y_{(n,l,c)} = N(n, l, c) \backslash \mathcal{R}$
        $label_{trans} = label(n) + l$;
        if $X \neq \emptyset$ then
            $nodes_{tmp} = \cup_{(\forall c_{tmp} \in \mathcal{A}(n) \backslash \{c\}, \forall l \in \mathcal{L})} N(n, l, c_{tmp})$
            $nodes_p = nodes_{tmp} \cap \mathcal{R}$
            for $x = 1$ to $|X|$ do
                for $y = 1$ to $|nodes_p|$ do
                    $latency_{node}(y) = l(nodes_p(y), X(x))$
                    $label_{node}(y) = label(nodes_p(y))$
                    $label_{round}(y) = latency_{node}(y) + label_{node}(y)$
                    if $label_{round}(y) < label_{trans}$ then
                        $Y_{(n,l,c)} = Y_{(n,l,c)} \backslash \{X(x)\}$; break
                    end if
                end for
            end for
        end if
        $X = Y_{(n,l,c)}$; $f(n, l, c) = |X| \div l$
    $NYC \leftarrow Y_{(\hat{n}, \hat{l}, \hat{c})}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup NYC$
    $label(NYC) = label(\hat{n}) + \hat{l}$
    if $\hat{n}$ already transmitting on channel $\hat{c}$ with latency $\tilde{l}$ to
      the nodes in $\tilde{I}$ (i.e., $(\hat{n}, \tilde{I}, \tilde{l}, \hat{c}) \in B_{PAMT}$) then
        $\bar{l} = max(\hat{l}, \tilde{l})$; $I = NYC \cup \tilde{I}$;
        $B_{PAMT} = (B_{PAMT} \backslash \{(\hat{n}, \tilde{I}, \tilde{l}, \hat{c})\}) \cup \{(\hat{n}, I, \bar{l}, \hat{c})\}$
    else
        $B_{PAMT} = B_{PAMT} \cup \{(\hat{n}, NYC, \hat{l}, \hat{c})\}$
    end if
end while
Output: $B_{PAMT}$
```

First of all, let us define the *label* of n as the total cost (latency) of the path from a node n to source s. Let us assume that node n can reach a set of nodes Y by transmitting on channel c with latency l_1. The *labels* of all nodes in Y would then be $label(n) + l_1$. Let us assume further that $Y' \subset Y$ can also be covered by a transmission of some other node n' (assume $label(n') < label(n)$) on channel c', with latency l_2. If covered by transmission of n', nodes in $Y' \subset Y$ would have a label of $label(n') + l_2$. Now, suppose that $label(n') + l_2 < label(n) + l_1$. As $Y' \subset Y$, LMT would still prefer the transmission of n to that of n' (as it covers more nodes), and therefore, would cover all the nodes in Y with n's transmission; this is despite the fact that nodes in $Y' \subset Y$ can be covered with a smaller path cost to s, if n' transmits in parallel on an alternative channel c'.

Algorithm. The PAMT algorithm works in a greedy manner, similar to the method of MWT and LMT, to choose the "best" transmission in each round. The priority metric $f(n, l, c)$ for each transmission (n, l, c), however, is calculated differently for PAMT. The PAMT algorithm maintains an extra parameter called *label* for each node, denoting the cost of its path to s (source node). The algorithm begins by adding node s to $\mathcal{R}$, which is the set of nodes that are eligible to transmit during the next round. The *label* of s is set to 0, and the *label* for all other nodes is set to ∞. During the execution of each round, PAMT tries to find out which transmission (or edge(s)) should be added to the tree. The set $Y_{(n,l,c)} = N(n, l, c) \backslash \mathcal{R}$ contains all hitherto "uncovered nodes" that can be covered by this transmission (n, l, c). The label of this transmission denoted by $label_{trans}$ is equal to $label(n) + l$.

During the calculation of priority for each transmission $(n, l, c), X$ contains the neighboring nodes $Y_{(n,l,c)}$ of the transmission (n, l, c). For each node in X, neighboring nodes are searched ($nodes_p$ in Algorithm 3) to find out if they can offer a lower cost path to s, on an *alternative* channel to c. If such a path is found, then this node *should* not be covered in the transmission (n, l, c). This node, therefore, is not considered a covered-node of (n, l, c) and is deleted from $Y_{(n,l,c)}$. After all nodes in X are checked in a similar manner, $Y_{(n,l,c)}$ contains the actual number of nodes that will be covered by the transmission (n, l, c). The priority of the transmission (n, l, c) is then calculated by dividing $Y_{(n,l,c)}$ by l.

In case of multiple transmissions having the same priority, the transmission whose channel $\hat{c}$ is least used in the conflict graph of that transmission, is chosen, as before. After completion of each round, covered nodes are added to $\mathcal{R}$. The algorithm completes its execution when all the nodes have been covered, i.e., when $V \backslash \mathcal{R} = \emptyset$. The algorithm returns the set B_{PAMT} whose collection of link-layer multicast transmissions forms the PAMT.

Example. We use an example topology shown in Fig. 2 (in which $Q = 2$ and $C = 4$) to describe PAMT's operation. Each WMN node is shown as three squares, where the bigger centralized box shows the node ID (with S being the broadcast source), and the two side squares indicate the channels used by the node's two interfaces. The channel used on the interface is shown as follows: $C1$ (blue full box), $C2$ (green horizontally lined box), $C3$ (red crossed box), and $C4$ (transparent box). Let us assume that PAMT initially decides that node S must cover nodes 5 (on $C3$) and 1 (on $C2$) using rates 11 and 5.5 Mbps, respectively. The label (which denotes the transmission latency to node S) of nodes 5 and 1 is 1 and 2, respectively. The nodes 1 and 5 (shaded in light blue) are now eligible transmitters (i.e., they are in $\mathcal{R}$) since they have now received transmission. Node 1 now covers node 2 through a 11 Mbps transmission; note that node 2 cannot be covered by by node 5 since it is outside its transmission range. *Node 1 can cover a total of two nodes (nodes 3 and 4) on a 5.5 Mbps transmission on* $C1$. *However, node 4 is also covered by node 5 (which has a lower label) on a 5.5 Mbps transmission orthogonal to* $C1$ *(i.e., on* $C3$). *Therefore, node 4 is not considered as a covered node of node 1, but is covered in the next round by node 5. By choosing node 5 as the parent of node 4 (and not node 1 which is transmitting to reach node 3), we can reduce the broadcast latency especially if we assume that node 5 connects to many downstream nodes.*

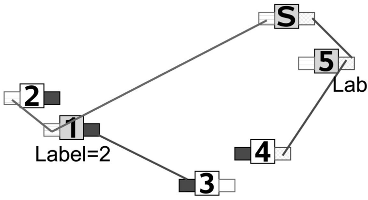


Fig. 2. Example topology to explain the working of PAMT.

Complexity. The complexity analysis of PAMT differs from that of LMT only in that each computation of $f(n, l, c)$ requires $O(N_t^2)$. Therefore, the complexity of PAMT is $O(N^2 |\mathcal{L}| Q (N_t^2 + N_i))$.

5 IDEALIZED TRANSMISSION SCHEDULING

To forward a single broadcast packet with minimal latency, we not only need to form the forwarding tree (calculated using one of the heuristics of Section 4), but also coordinate the actual transmission schedules at different forwarding nodes. In this section, we present how such idealized "transmission scheduling" may be computed. It should be emphasized that we are not proposing this technique as a practical protocol that can be implemented on a WMN; rather the utility of this technique is in computing the "theoretically lowest latency" that a forwarding tree can achieve.

The "transmission scheduling" algorithm tries to schedule the transmissions to minimize the broadcast delay *for a single packet*, while ensuring no interfering transmissions are scheduled for simultaneous transmission. The broadcast tree calculated by the heuristics in Section 4 is modeled by a set of link-layer multicast transmissions $B = \{b_1, \ldots, b_j, \ldots\}$ such that the collection of all edges in all transmissions forms a spanning tree. With the notation introduced in Section 3, the transmission b_j is a 4-tuple $(p_j, I_j, \bar{l}_j, k_j)$, meaning the node p_j performs a link-layer multicast transmission to the nodes in I_j at latency $\bar{l}_j$ in channel k_j. In fact, the transmitting node in each transmission b_j corresponds to a branching node (i.e., nonleaf nodes) in the broadcast tree.

Formally, the transmission schedule is a map $\tau : B \rightarrow \mathbb{R}$, where $\tau(b_j)$ is the start time of transmission b_j. Since transmission b_j has a latency of $\bar{l}_j$, this transmission occupies the time interval $[\tau(b_j), \tau(b_j) + \bar{l}_j]$. The transmission schedule must obey the following constraints:

1. The source node s must transmit at least once at time zero.
2. All nodes must follow precedence constraint, i.e., a node can only transmit after receiving the packet from its parent, e.g., nodes in I_j receive the packet from its parent p_j at time $\tau(b_j) + \bar{l}_j$.
3. Two arbitrary transmissions can be scheduled together (i.e., have overlapping transmission time interval) if and only if they do not interfere with each other. (See Section 3 for interference model.)

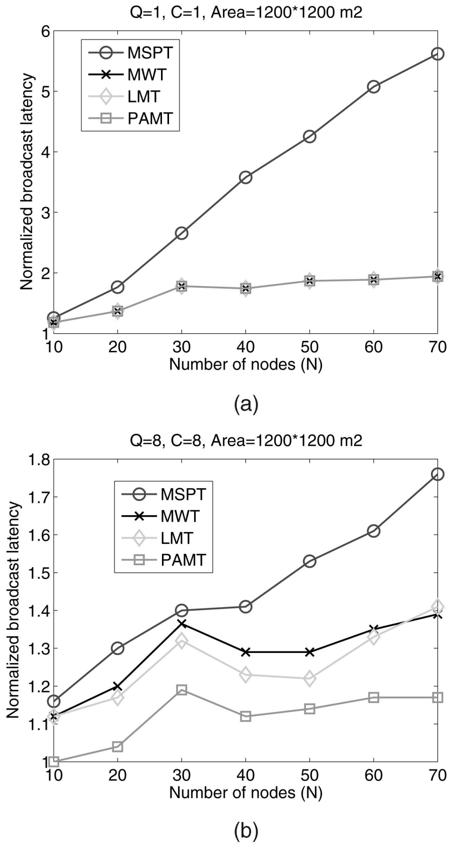
The aim of the ideal scheduling is to minimize the broadcast latency, as defined in Section 3. We limit our discussion of the "transmission scheduling" algorithm due to space constraints, but we point out that the scheduling algorithm presented for the case of SR-SC multirate WMNs [7], [11] can be used for scheduling in MR^2-MC multirate scenario with slight modifications. In fact, the main difference (details are provided in [19]) lies in the fact that the interference model needs to be modified, as described in Section 3, to ensure that transmissions transmitting on orthogonal channels can be scheduled together.

6 RESULTS USING IDEALIZED SCHEDULER

We have assumed static WMNs composed of N nodes randomly located in an area of $1{,}200 \times 1{,}200\ \mathrm{m}^2$. The transmission rate-range relationship for 802.11b depicted in Table 1 is assumed. We have considered three channel assignment schemes in our current work: CCA, VCA, and INSTC (discussed earlier in Section 2). The effects of the number of nodes in the network, the number of radio interfaces at each node, and the channel assignment strategy are observed on the broadcast latency when using our algorithms. *The results presented in this section were obtained using MATLAB [20] simulations while making the following idealized assumptions.*

1. We assume a binary interference model as follows: If while a node k is receiving a frame, a node j within interference range from node k transmits a frame, then the frame that k is receiving is assumed to be corrupted and lost.
2. We assume an ideal MAC layer as follows: Two nodes i and j can multicast at the same time if and only if node i's multicast does not interfere with the intended recipients of node j's multicast and vice versa.
3. We assume a centralized scheduler (described in Section 5) which schedules these multicasts such that, under the ideal MAC layer assumption, no two multicasts will interfere with each other.

The outline of the remaining section is: we use the CCA channel assignment scheme to study the effect of varying node density in Section 6.1, and varying radio interfaces in Section 6.2 on broadcast performance. The effect of the choice of channel assignment scheme on broadcast latency is then described in Section 6.3.



(a)

(b)

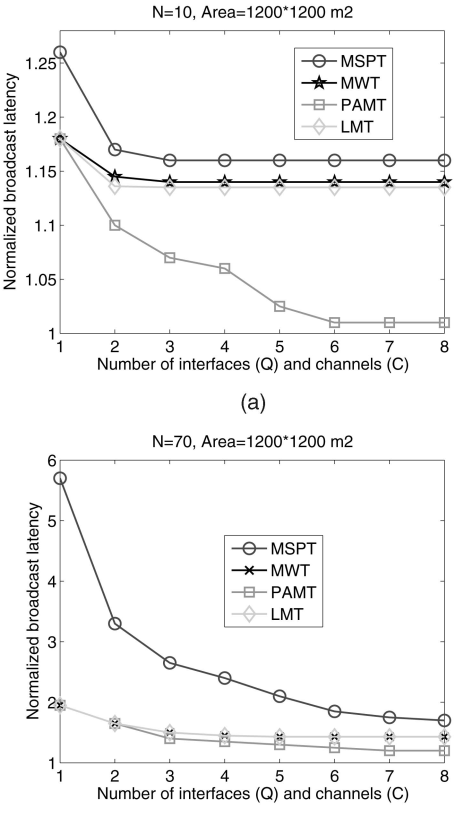
Fig. 3. Normalized broadcast latency against varying N, for $Q(=C) = 1$ and 8, respectively, in an area of $1{,}200 \times 1{,}200\ \mathrm{m}^2$ using idealized scheduler. (a) $Q = C = 1$. (b) $Q = C = 8$.

6.1 Effect of Node Density

The effect of network's node density on the performance of our heuristics can be seen in Figs. 3a and 3b for the case of Q and C being 1 and 8, respectively. The vertical axis shows the broadcast latency of our heuristics normalized against the broadcast latency of the Dijkstra's tree with infinite number of Q and C. Normalization here refers to dividing the latency of our algorithms by the latency of the "*idealized*" Dijkstra tree (which is the optimal value that may be achieved when the numbers of radios and channels are unlimited); accordingly, a normalized value of 1 indicates the idealized lower bound on achievable latency. It is observed that the PAMT algorithm performs the best of all algorithms for the range of network node density (10-70 nodes in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$). The performance of LMT, although it uses parallelization like PAMT, is not as good as PAMT's (Figs. 3a and 3b) but nonetheless is better than MWT and MSPT. The performance of MSPT, expectedly, is poor and worsens as the network node density is increased.

6.2 Effect of Number of Radio Interfaces

The performance of our heuristic algorithms for the case of an SR-SC multirate WMN is presented in Fig. 3a. MWT, LMT, and PAMT perform identically for the specific case of an *SR-SC multirate WMN* (i.e., when $Q = C = 1$), with MSPT performing considerably worse (Fig. 3a). These results are similar to those obtained in our earlier work [7]. MWT performs better than MSPT since it considers both the WBA



(a)

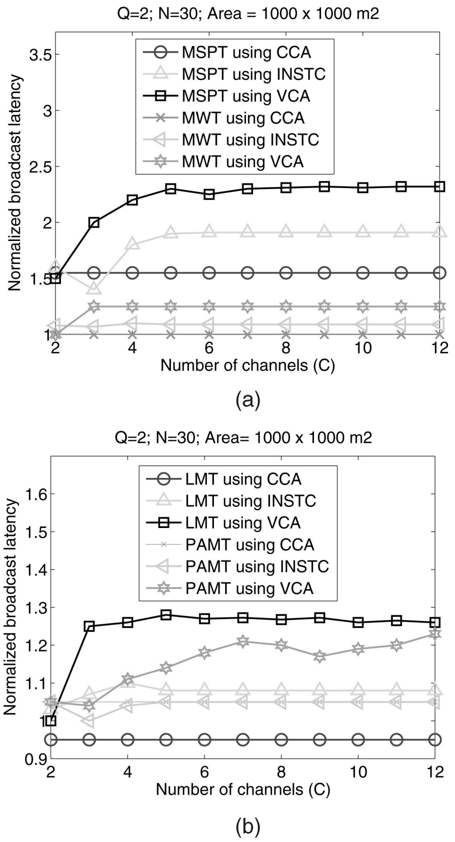
(b)

Fig. 4. Normalized broadcast latency against varying Q and C ($Q = C$) in an area of $1{,}200 \times 1{,}200\ \mathrm{m}^2$ using idealized scheduler. (a) $N = 10$. (b) $N = 70$.

and the multirate nature of the mesh (Fig. 3a). The LMT and PAMT algorithms, both adapted from MWT, can only *match* and *not improve* the performance of MWT (Fig. 3a) in SR-SC multirate scenarios, since both cannot find *alternative* channel paths to "*parallelize*" transmissions. *Thus, for SR-SC multirate WMN, the performance of LMT and PAMT is exactly the same as MWT.*

For the cases of MR^2-MC multirate meshes, where $Q > 1$, all of our proposed heuristics improve their performance. This is true both for small networks ($N = 10$, Fig. 4a) and for large networks ($N = 70$, Fig. 4b). Figs. 4a and 4b display representative performance of different heuristics for MR^2-MC WMNs across the range of radio interfaces from $Q = 2$ to $Q = 8$.

The improvement seen in MR^2-MC performance can be attributed to two main reasons: *First*, the usage of MR^2-MC minimizes the interference in the network and allows interfering transmissions to be transmitted simultaneously using orthogonal channels. This improvement factor called "*interference reduction factor*" applies to all our proposed heuristics and substantially improves performance when the heuristic-constructed tree involves many transmissions (e.g., as in MSPT). *Second*, a heuristic broadcasting algorithm that parallelizes its transmission, according to the number of available interfaces and channels, reaps extra benefits by efficient usage of the available resources. This improvement factor called the "*radio adaption factor*" is specific to broadcasting algorithms such as LMT and PAMT.



(a)

(b)

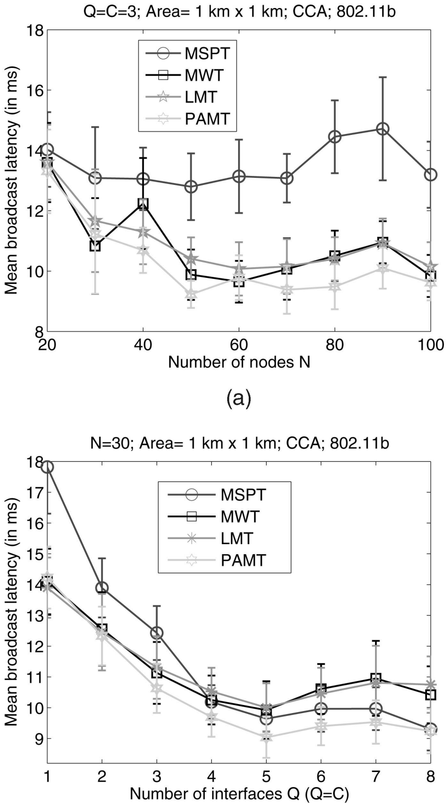
Fig. 5. The impact of channel assignment on broadcast latency performance, where $Q = 2$ and $N = 30$ in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$ using an idealized scheduler. (a) MSPT and MWT and (b) LMT and PAMT.

Finally, we point out the performance gain due to multiple radio interfaces in MR^2-MC meshes over SR-SC multirate meshes. Referring to Figs. 4a and 4b, we see that for Q as less as 3 or 4, the broadcast latency decreases by about 30-40 percent compared to the scenario where well-designed heuristics are used and by as much as 80 percent when poorly designed heuristics (e.g., MSPT) are used for $Q = 1$.

6.3 Effect of Channel Assignment Scheme

The graphs of the performance of different channel assignment schemes (CCA, VCA, and INSTC) are shown in Figs. 5a and 5b for the cases of $Q = 2$. The results shown are representative of similar results seen across different values of Q. The vertical axes in the graphs show broadcast latency of the algorithm normalized against the MWT algorithm with channels assigned through CCA. All the channel assignment schemes considered have different *connectivity* and *interference* characteristics. As noted earlier, the topology given as input to our heuristics greatly affects the broadcast performance; with the input topology being defined by the channel assignment scheme, broadcast performance is closely affected by the channel assignment scheme chosen.

In CCA, a set of common channels is shared among all nodes; hence, both the connectivity and interference are maximum. In VCA, although connectivity is ensured by tuning one interface at all nodes to a common channel, the remaining interfaces are assigned channels randomly from



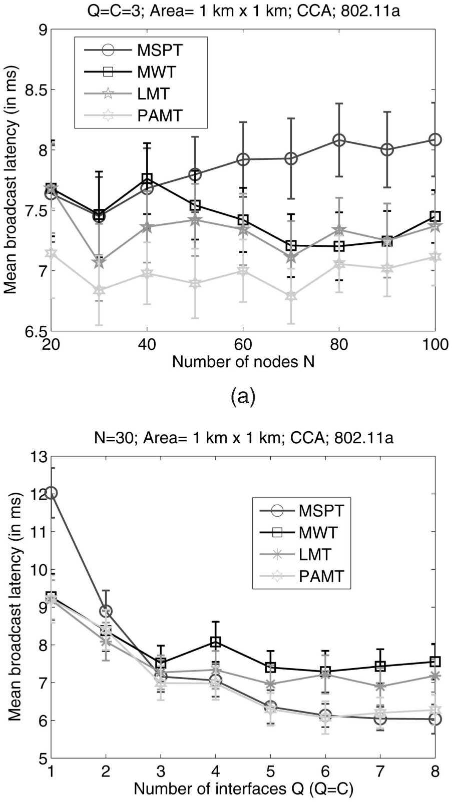
(a)

(b)

Fig. 6. Broadcast latency (in millisecond) for varying N and Q (with $Q = C$) in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$ for 802.11b networks using Qualnet simulator. (a) Varying nodes while using 802.11b. (b) Varying radio resources while using 802.11b.

the remaining channels in C. The connectivity, therefore, can suffer at the cost of reduced interference. In INSTC, much like VCA, network interference is reduced by increasing channel diversity; however, this is at the cost of reduced connectivity which can possibly mitigate the WBA. An ideal channel assignment algorithm has to balance the two conflicting requirements of low interference and high connectivity. In the presence of low interference, more transmissions can be scheduled simultaneously resulting in reduced broadcast latency. Similarly, with large connectivity, there are increased opportunities of availing the WBA.

From Figs. 5a and 5b, it can be seen that for values of C only slightly larger than Q, VCA and INSTC can, sometimes, outperform CCA. This is because in such a scenario, the effect of reduced interference outweighs any reduction in connectivity. However, with further increase in C, the reduced connectivity can adversely affect the broadcast latency of the heuristics by neutralizing WBA. This leads to generally more transmissions (not availing the WBA), and higher broadcast latencies. The characteristic of reduced interference in VCA and INSTC schemes has a more pronounced effect on the performance of MSPT, LMT, and PAMT than on MWT, since these algorithms generally involve more transmissions (on possibly interfering channels). As we can see from Figs. 5a and 5b, the best performing channel assignment scheme for broadcast generally is CCA (which performs poorly for unicast flows [6]). Although the channel assignment scheme INSTC gives improved performance for unicast traffic, it is not necessarily the best performing channel assignment scheme for broadcast. Thus, we make an important observation that *a channel assignment scheme designed for unicast flows may, sometimes, perform poorly for broadcast/multicast flows*.



(a)

(b)

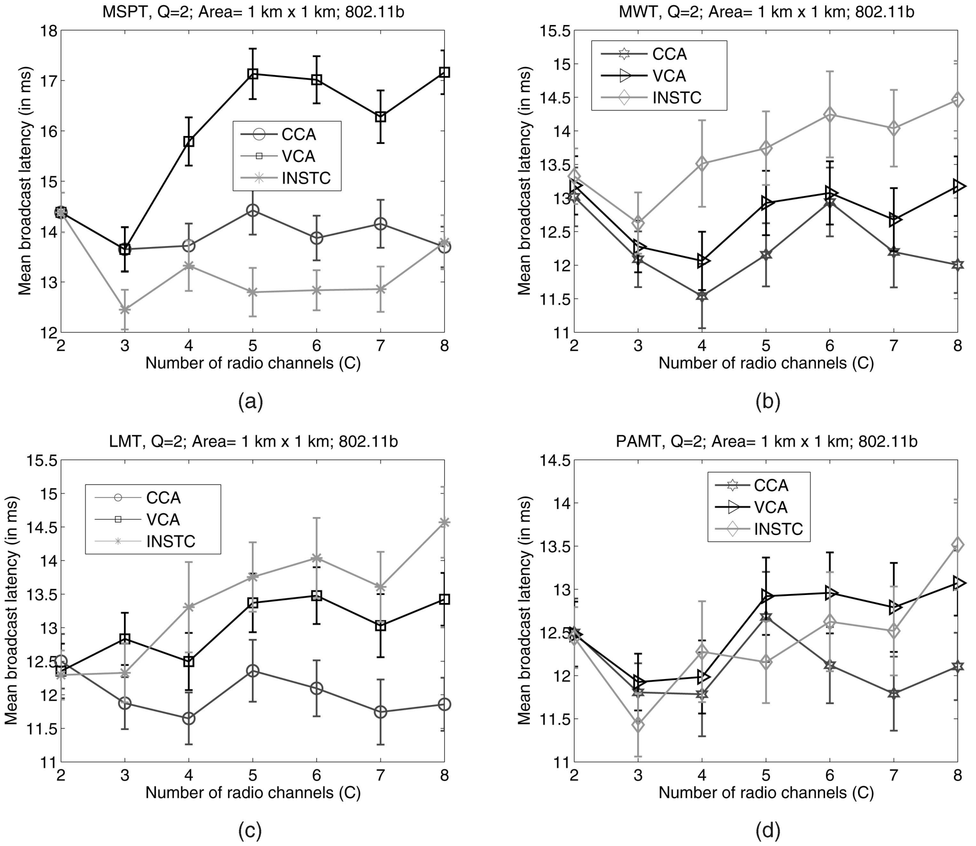
Fig. 7. Broadcast latency (in millisecond) for varying N and Q (with $Q = C$) in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$ for 802.11a networks using Qualnet simulator. (a) Varying nodes while using 802.11a. (b) Varying radio resources while using 802.11a.

7 RESULTS USING QUALNET'S 802.11 MAC

In this section, we present our algorithms' performance evaluation results, assuming a decentralized MAC scheduler, obtained using the Qualnet [18] and MATLAB [20] simulators. We have used both 802.11a and 802.11b as our MAC scheduler with PHY 802.11a and PHY 802.11b, respectively, at the physical layer which use a preconfigured BER-based packet reception model. The transmission rate versus transmission range relationship for 802.11a and 802.11b, as obtained from Qualnet, is shown in Table 1. The IEEE 802.11 MAC with Distributed Coordination Function (DCF) was chosen as the medium access control protocol. All default parameters are assumed unless stated otherwise. The ticks in the graphs represent 5th and 95th percentiles over 100 uniformly distributed random topologies. The broadcast latency results are shown in units of milliseconds. We will now proceed to discuss the results in the next few sections.

7.1 Effect of Node Density

The effect of network's node density on the performance of our heuristics can be seen in Figs. 6a and 7a for $802.11b$ and 802.11a networks, respectively. It is observed that the



(a) (b)

(c) (d)

Fig. 8. Broadcast latency performance (in millisecond) for different channel assignment techniques, for varying C with $Q = 2$ and $N = 30$ in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$, assuming 802.11b networks in Qualnet simulator. (a) MSPT. (b) MWT. (c) LMT. (d) PAMT.

PAMT algorithm, as presented in Section 6 for an idealized scheduler, performs the best of all algorithms for the range of network node density (20-100 nodes in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$). The performance of LMT, although it uses parallelization like PAMT, is not as good as PAMT's (Figs. 6a and 7a) as in the idealized MAC case where LMT performed better than MWT (Section 6) but worse than PAMT. The performance of MSPT, expectedly, is poor and worsens as the network node density is increased.

7.2 Effect of Number of Radio Interfaces

The effect of varying the number of radio interfaces Q on the performance of our algorithms can be seen in Figs. 6b and 7b for $802.11b$ and $802.11a$ networks, respectively. It is observed that, for values of Q as low as 2 or 3, PAMT outperforms all other algorithms. It is also seen that MSPT's performance improves with increasing values of Q; MSPT performs comparably to MWT and LMT for values of Q as low as 4 and 3 (Figs. 6b and 7b) for 802.11b and 802.11a networks, respectively. However, MSPT requires a very high value of Q ($Q = 8$ for 802.11b and $Q = 4$ for 802.11a) to match PAMT's performance. Here, it must be pointed out that MSPT with infinite number of Q is the ideal solution to the MLB problem (however, it requires unrealistically large number of radio resources).

7.3 Effect of Channel Assignment Scheme

The performance of MSPT, MWT, LMT, and PAMT with different channel assignment schemes is shown in Figs. 8a, 8b, 8c and 8d, respectively. We note that the algorithms that incorporate WBA in their design (i.e., MWT, LMT, and PAMT) can benefit from increased "connectivity" present in schemes like CCA that present more opportunities for exploiting WBA. These results are similar to those presented in Section 6.3 (where an idealized MAC scheduler was used). Interestingly, the MSPT algorithm, which does not take WBA into account, behaves like a typical unicast protocol wherein INSTC presents better results than CCA or VCA.

8 SIMULATION RESULTS IN QUALNET FOR A STREAM OF BROADCAST PACKETS

We have noted in Sections 6 and 7 that PAMT improves the performance of MWT and LMT throughout increased parallelization. These improved results are relevant to the case of a single broadcast packet. Since most broadcast comprises a stream of transmitted packets, we are also interested in knowing our protocol's broadcast performance for a stream of packets. Toward this end, we have programmed the Qualnet simulator to simulate the broadcast of a stream of 100 packets where each packet comprises 1,500 bytes. We assume that successive packets of the broadcast stream are separated in time by an interval called the "*interpacket delay interval*."

We will compare our algorithms for a stream of broadcast packets using two metrics: 1) the "total broadcast latency" and 2) the "broadcast delivery percentage." The

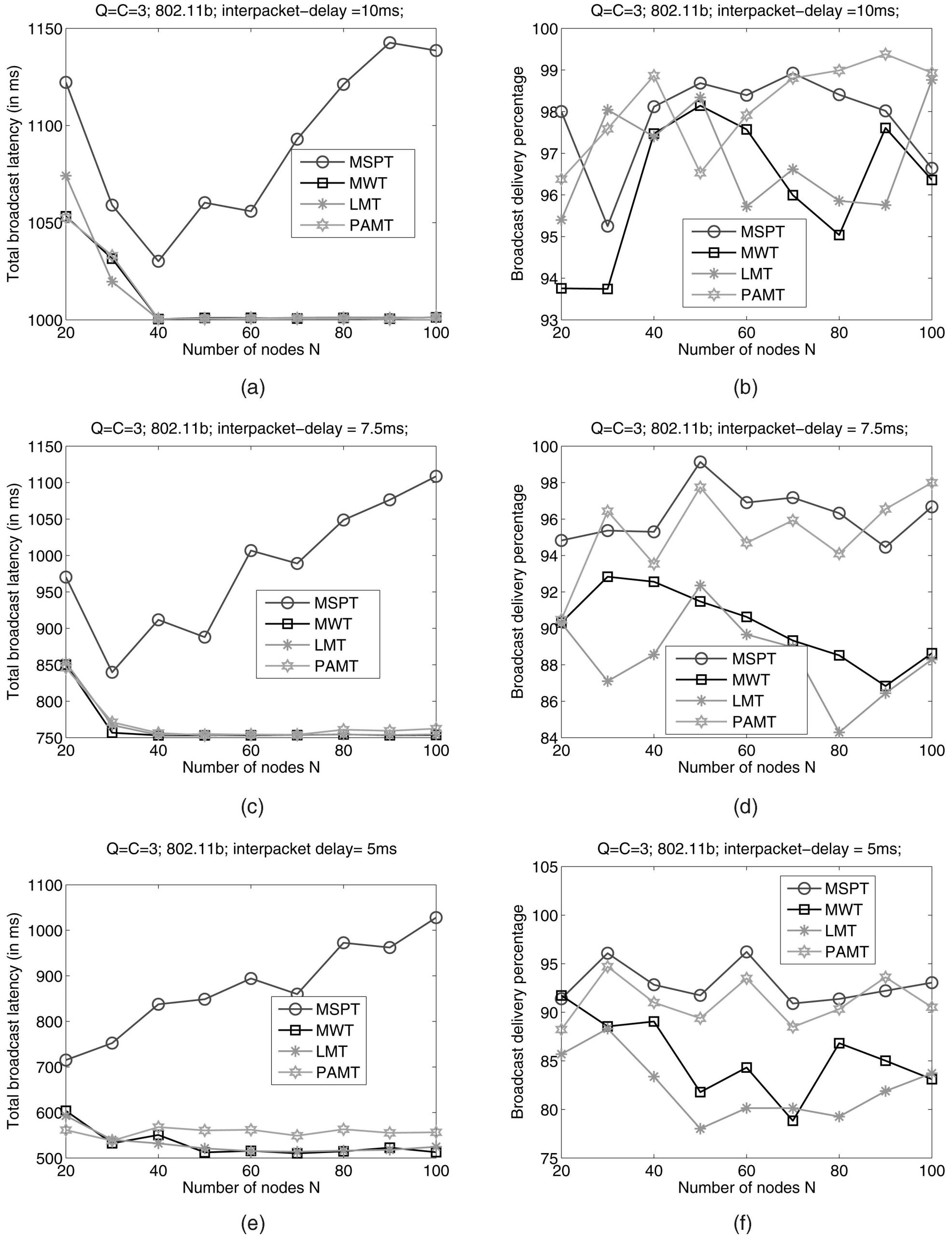


Fig. 9. "Total broadcast latency" (in millisecond) and "broadcast delivery percentage" for varying N in an area of $1{,}000 \times 1{,}000\ \mathrm{m}^2$ with $Q = C = 3$ assuming interpacket delays of 10, 7.5, and 5 ms, respectively. (a) Broadcast delay (millisecond); interpacket $\mathrm{delay} = 10\ \mathrm{ms}$. (b) Delivery percentage; interpacket delay = 10 ms. (c) Broadcast delay (millisecond); interpacket $\mathrm{delay} = 7.5\ \mathrm{ms}$. (d) Delivery percentage; interpacket $\mathrm{delay} = 7.5\ \mathrm{ms}$. (e) Broadcast delay (millisecond); interpacket $\mathrm{delay} = 5\ \mathrm{ms}$. (f) Delivery percentage; interpacket $\mathrm{delay} = 5\ \mathrm{ms}$.

"*total broadcast latency*" is defined as the time taken from the transmission of the first packet till the time the 100th packet of the broadcast stream is received at *all* nodes. We note that successive packets of the broadcast stream are delayed by the interpacket delay interval. If we do not place the condition that the 100th packet be received at all recipient nodes, then the last received packet at different nodes (for different algorithms and topology instances) can be different (e.g., nodes further away from the broadcast source might not receive the packet), and comparison of our algorithms by stamping the latest receive time of the last packet at all nodes can artificially show lower broadcast latency due to packet loss. However, with the condition that 100th packet must be received, packets lost earlier do not make a big difference, since the 100th packet, which is received at all nodes, started at a fixed time (dictated by the interpacket delay interval); thus with the condition enforced that the 100th packet is received at *all* nodes, a more consistent comparison can be made between our different algorithms. The "*broadcast delivery percentage*," on the other hand, is defined as the average (over all network nodes) of the average number of packets received at a network node among the 100 packets of broadcast stream. The "total broadcast latency" results and "broadcast delivery percentage" results are displayed in Figs. 9a, 9c, and 9e and Figs. 9b, 9d, and 9f for different interpacket delay intervals of 10, 7.5, and 5 ms, respectively. *These results assume the CCA channel assignment and the usage of IEEE 802.11b MAC scheduler in Qualnet simulations.*

The simulation presents interesting results. While PAMT offered significant broadcast latency performance improvement over MWT and LMT (due to increased parallelization) for a single broadcast packet, we see that the "broadcast latency" performance of MWT, LMT, and PAMT is similar (shown in Figs. 9a, 9c, and 9e) when broadcasting a packet stream. In fact, with decreasing interpacket delay interval (e.g., with 5 ms, Fig. 9e), we see that the "total broadcast latency" performance of MWT and LMT is better than that of PAMT. This results from the fact that PAMT's increased parallelization (the activation of more interfaces) is not helpful for packet streams due to the imperfections of the IEEE 802.11 MAC (where a larger number of transmissions imply greater probability of collision-based loss, especially for broadcast transmissions that do not employ RTS/CTS/ACK exchanges and retransmissions). In particular, PAMT parallelizes the transmissions *for the same packet* on orthogonal channels; when packets arrive in a stream, it is entirely possible for transmissions (of *different* packets) to collide (since they are scheduled for the same channels), as interpacket delay interval decreases. In MWT and LMT algorithms, where the parallelization is more conservative, but where WBA is utilized nonetheless, streamed transfer of broadcast packets is not impeded as much as in PAMT algorithms. For similar reasons, the performance of MSPT (which causes the largest number of separate transmissions) remains much poorer than either of the other three heuristics.

The other results that we have studied measure the "broadcast delivery percentage." This is calculated by taking the average percentage delivery of all the broadcast receiving nodes. The results show that for a given interpacket broadcast interval, generally, the higher the broadcast latency, the better the delivery ratio. *When the interpacket broadcast interval is equal to or less than the "broadcast latency,"* a subsequent broadcast packet (behind the earlier packet) is broadcasted by the source node before the earlier packet is received at all nodes. Due to transmissions of different packets in a stream, interference can result since for all packets, the same set of transmitting nodes will transmit on the same channels. *On the other hand, when the "broadcast latency" is larger than the interpacket broadcast interval*, the subsequent packet is only started after the earlier packet has reached all nodes. This results in fewer collisions, as successive packet transmissions do not vie for the same channel. We note here that while for the same packet transmissions, our algorithms attempt to parallelize transmissions on orthogonal channels; for different packet transmissions, as the interpacket broadcast interval becomes less than or equal to "broadcast latency," the interference between transmissions on the same channel increases. We note, therefore, in our results that the delivery percentage performance of MSPT and PAMT is better than the performance of MWT and LMT, especially when the interpacket broadcast interval is reduced, as seen in Figs. 9a, 9c, and 9e.

9 CONCLUSIONS

In this paper, we have studied the MLB problem for multiradio, multichannel, multirate WMNs. We have proposed two heuristic mechanisms (called MSPT and MWT) as direct extensions of corresponding algorithms for SR-SC environments, and then developed two enhanced heuristic routing schemes (called LMT and PAMT), designed specifically to exploit the greater concurrency permitted in MR^2-MC WMNs.

We studied the performance of our algorithms through detailed simulations using both 1) an idealized scheduler (with idealized MAC assumptions) and 2) a practical IEEE 802.11 MAC-based scheduler. We show that PAMT outperforms the other algorithms by increasing the likelihood of concurrent transmissions taking place on different interfaces (at the same node, or at different nodes). However, when the transmitted flow consists of a stream of packets broadcast using the 802.11 MAC, LMT's less aggressive use of parallel transmissions results in performance comparable to PAMT.

The simulation results and performance studies (for both the idealized and 802.11-based scheduler) also show the impact of channel assignment strategies on broadcast latency, due to the conflict between greater connectivity and lower channel contention. Perhaps, a more important observation is that a channel assignment scheme designed for unicast flows may sometimes perform poorly for broadcast/multicast flows. In our simulations for both the idealized and 802.11-based scheduler, the performance of CCA (which usually performs poorly for unicast flows) is generally better than both VCA and INSTC.

ACKNOWLEDGMENTS

This work was supported in part by the Australian Research Council (ARC) Discovery Grant DP0664791. This work was done when Junaid Qadir was at the University of New South Wales (UNSW). He was supported by UNSW's University Postgraduate Award (UPA), and jointly by Pakistan's Higher Education Commission (HEC) and National University of Science and Technology (NUST).

REFERENCES

[1] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, no. 4, pp. 445-487, 2005.
[2] B. Raman and K. Chebrolu, "Design and Evaluation of a New MAC Protocol for Long-Distance 802.11 Mesh Networks," *Proc. ACM MobiCom*, pp. 156-169, 2005.
[3] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory*, vol. 46, no. 2, pp. 388-404, Mar. 2000.
[4] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," *Proc. ACM MobiCom*, pp. 114-128, 2004.
[5] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks," *ACM SIGMOBILE Mobile Computing and Comm. Rev.*, vol. 8, no. 2, pp. 50-65, 2004.
[6] P. Kyasanur and N.H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi Interface Wireless Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05)*, vol. 4, pp. 2051-2056, 2005.
[7] C.T. Chou, A. Misra, and J. Qadir, "Low Latency Broadcast in Multi-Rate Wireless Mesh Networks," *IEEE J. Selected Areas inComm.*, special issue on wireless mesh networks, vol. 24, no. 11, pp. 2081-2091, Nov. 2006.
[8] C.T. Chou and A. Misra, "Low Latency Multimedia Broadcast in Multi-Rate Wireless Meshes," *Proc. First IEEE Workshop Wireless Mesh Networks (WiMesh '05), in Conjunction with IEEE Sensor and Ad Hoc Comm. and Networks (SECON '05)*, 2005.

[9] *The Status of the IEEE 802.11s Standard Project,* http://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm, 2009.

[10] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks," *Proc. ACM MobiHoc,* pp. 222-232, 2003.

[11] C.T. Chou, A. Misra, and J. Qadir, "Low Latency Broadcast in Multi-Rate Wireless Mesh Networks," Technical Report 0514, School of Computer Science and Eng., Univ. of New South Wales, ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0514.pdf, 2005.

[12] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Mobile Networks Application,* vol. 7, no. 6, pp. 481-492, 2002.

[13] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Minimum-Energy Broadcasting in Ad-Hoc Networks," *Proc. IEEE INFOCOM,* vol. 3, pp. 2210-2217, 2003.

[14] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," *IEEE Trans. Mobile Computing,* vol. 1, no. 2, pp. 111-122, Apr.-June 2002.

[15] P.M. Ruiz and A.F. Gomez-Skarmeta, "Heuristic Algorithms for Minimum Bandwidth Consumption Multicast Routing in Wireless Mesh Networks," *Proc. AD-HOC Networks and Wireless (ADHOC-NOW),* pp. 258-270, 2005.

[16] C. Gui and P. Mohapatra, "Scalable Multicasting in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* pp. 2119-2129, 2004.

[17] J. Tang, G. Xue, and W. Zhang, "Interference-Aware Topology Control and qos Routing in Multi-Channel Wireless Mesh Networks," *Proc. ACM MobiHoc,* pp. 68-77, 2005.

[18] Scalable Networks, Inc., http://www.scalable-networks.com, 2009.

[19] J. Qadir, C.T. Chou, and A. Misra, "Low Latency Broadcast in Multi-Radio Multi-Channel Multi-Rate Wireless Mesh Networks," Technical Report 0608, School of Computer Science and Eng., Univ. of New South Wales, ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0608.pdf, 2006.

[20] The MathWorks, Inc., http://www.mathworks.com, 2009.

Junaid Qadir received the BS degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, and the PhD degree in computer engineering from the University of New South Wales, Australia. He is currently an assistant professor at the School of Electrical Engineering and Computer Science (SEECS) of the National University of Sciences and Technology (NUST), Pakistan, where he is also the director of the IP Technology Center of Excellence. His current research interests include IP-based next-generation networks, cognitive networking technologies, routing issues in wireless networks, and context-aware IPTV services.

Chun Tung Chou received the BA degree in engineering science from the University of Oxford, UK, and the PhD degree in control engineering from the University of Cambridge, UK. He is an associate professor at the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He has published more than 100 papers in leading journals and conferences on various topics, including wireless mesh networks, wireless sensor networks, multimedia, traffic engineering, system identification, and process control. His current research interests include wireless networks, participatory sensing, multimedia, and network optimization.

Archan Misra received the BTech degree in electronics and communication engineering from IIT Kharagpur, India, in July 1993, and the PhD degree in electrical and computer engineering from the University of Maryland, College Park, in May 2000. He is currently a senior scientist in the Advance Technology Services Division of Telcordia Technologies, where he is leading research initiatives on network science theory, cognitive networking technologies, and pervasive healthcare solutions. Most recently, he was a researcher at the IBM TJ Watson Research Center, Hawthorne, New York, where he worked on algorithms and protocols for sensor data dissemination, distributed infrastructures for stream analytics, and presence technologies for converged networks. He has published extensively in the areas of wireless networking and pervasive computing and is a coauthor on papers that received the Best Paper Awards in ACM WOWMOM 2002 and IEEE MILCOM 2001. He is presently on the editorial boards of the *IEEE Wireless Communications Magazine* and the *Journal of Pervasive and Mobile Computing* and was the chair of the IEEE Computer Society's Technical Committee on Computer Communications (TCCC) between 2007 and 2008.

Joo Ghee Lim received the BE and ME degrees in electrical and computer engineering from the National University of Singapore (NUS) in 1998 and 2001, respectively. He is currently working toward the PhD degree at the School of Computer Science and Engineering, University of New South Wales (UNSW), Australia. From 2001 to 2005, he was an associate scientist at the Institute for Infocomm Research, Singapore. His current research interests include MAC, QoS, and resource optimization and internetwork coexistence issues in WLANs and wireless multihop networks.