

Minimum Latency Broadcasting in Multi-Radio Multi-Channel Multi-Rate Wireless Meshes

Junaid Qadir¹, Archan Misra², Chun Tung Chou¹

¹School of Computer Science and Engineering, University of New South Wales, Australia

²IBM T J Watson Research Center, Hawthorne, New York, USA

Email: {junaidq, ctchou}@cse.unsw.edu.au, archan@us.ibm.com

Abstract: We address the problem of minimizing the worst-case broadcast delay in multi-radio multi-channel multi-rate (MR²-MC) wireless mesh networks (WMN). The problem of ‘efficient’ broadcast in such networks is especially challenging due to the numerous inter-related decisions that have to be made. The multi-rate transmission capability of WMN nodes, interference between wireless transmissions, and the hardness of optimal channel assignment adds complexity to our considered problem. We present four heuristic algorithms to solve the minimum latency broadcast problem for such settings and show that the ‘best’ performing algorithms usually adapt themselves to the available radio interfaces and channels. 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. To the best of our knowledge, this work constitutes the first contribution in the area of broadcast routing for MR²-MC WMN.

I. INTRODUCTION

Wireless mesh networks (WMN) [1], where potentially mobile mesh clients connect over a relatively-static multi-hop 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 multi-hop wireless environments (due to wireless interference) remains 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 employing concurrent transmissions in the network [4][5][6]. Another feature widely available in commodity wireless cards,

This work is part of the ongoing AIOLOS project, supported by Australian Research Council (ARC) Discovery Grant DP0664791, at UNSW. The project website is <http://www.cse.unsw.edu.au/aiolos/>

Junaid Qadir is supported by an University of New South Wales (UNSW) University Postgraduate Award (UPA) and by Government of Pakistan Higher Education Commission (HEC)/ National University of Science and Technology (NUST) Scholarship.

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 multi-rate 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, multi-radio multi-channel multi-rate (MR²-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 from the research community.

An important open question in MR²-MC WMNs which we attempt to address in this paper is how to perform ‘efficient’ broadcast¹ 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 MR²-MC meshes due to a myriad of complex, inter-twined decisions that need to be made. The authors of [6] have hinted about some of the potential problems that can be faced for broadcast routing in multi-radio meshes (*vis-a-vis* channel assignment).

The MLB problem, apart from its theoretical significance, is an important practical problem in WMN. The presence of several multi-party applications—such as software updates to all devices, local content distribution (e.g., video feeds) in community networks and multimedia gaming—often impose stringent latency requirements on the underlying network and motivate the study of the MLB problem. The MLB problem has been studied for ‘single-radio single-channel’ (SR-SC) wireless networks, both for the single-rate [10] and the multi-rate case [7] [8]. To the best of our knowledge, the MLB problem for MR²-MC WMNs has not been addressed in literature and our work is the first contribution in this area. We shall show that the MLB problem for MR²-MC meshes is a more complex problem than for SR-SC multi-rate meshes (single-radio meshes are a special case of multi-radio meshes). The differences between *single-rate* and *multi-rate* 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] [11] and [10] respectively.

¹We assume that multicast can be realized by pruning the broadcast tree.

II. 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 omni-directional antennas are being used—a transmission by a node can be received by all neighboring nodes that lie within its communication range. This situation is quite different to that of wired networks where the cost to reach two neighbors is generally the sum of the costs to reach them individually. This is due to the shift in paradigm from the ‘*link-centric*’ nature of wired networks to the ‘*node-centric*’ nature of wireless communications. A lot of research has focussed on achieving ‘*efficient*’ broadcast in multi-hop 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 focussed only on SR-SC networks [7] [10] [11].

Our current work builds upon our previous work on minimizing broadcast latency in a SR-SC *multi-rate* WMN [7] [8], where we introduced the new concept of link-layer *multi-rate* multicast, in which a node can adjust its link-layer multicast transmission rate to its neighbors. We showed that multicast in a multi-rate WMN has two features not found in a single-rate WMN. Firstly, 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 . Secondly, for a multi-rate 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 in [7] [11] as heuristic solutions for the MLB problem in SR-SC multi-rate mesh networks. Both these algorithms consider the WBA and the multi-rate 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 MR²-MC wireless meshes 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 nodes will usually benefit from increased ‘connectivity’ among themselves, the channel assignment protocol also tries to reduce some measure of ‘interference’. The different channel assignment strategies can be classified into *static*, *dynamic*, and *hybrid* categories [6]. For our current work, we do not consider dynamic and hybrid strategies, due to the non-negligible interface switching delay and synchronization requirements involved in such strategies.

Amongst 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.

Transmission rate (Mbps)	Transmission range
1	483
2	370
5.5	351
11	283

TABLE I
RELATIONSHIP BETWEEN TRANSMISSION LATENCY AND RANGE

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.

III. NETWORK AND INTERFERENCE MODEL

We follow the notation introduced by [17] to represent our channel assignment; also we use a similar network model to that described by [5]. Each node in the network can transmit at multiple-rates. There are totally C non-overlapping orthogonal frequency channels in the system and each node is equipped with Q radio interfaces where $Q \leq C$. The Q radio interfaces have omni-directional antennas, and unit disk graph model is assumed. In order to efficiently utilize the network resources, two radio interfaces at the same node are not tuned to the same channel. Using the Qualnet simulator [18] as a reference, we obtain the transmission rate versus transmission range relationship in Table I, 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 I illustrates.

We use an undirected graph $G_T = (V, E_T, L_T)$ to model the given mesh network topology *before* channel assignment, where V is the set of vertices, E_T is the set of edges and L_T is the set of weights of edges in E_T . The vertex v in V corresponds to a wireless node in the network with a known location. An undirected edge (u, v) , corresponding to a wireless link between u and v , is in the set E_T *if and only if* $d(u, v) \leq r$ where $d(u, v)$ is the Euclidean distance between u and v and r is the range of the *lowest-rate* transmission. The latency of a link $l(u, v)$ is the latency of the ‘fastest’ transmission rate that can be supported between nodes u and v . The set L_T contains the latencies of all links in E .

Channel assignment: A channel assignment \mathcal{A} assigns each vertex v in V , Q different channels denoted by the set: $A(v) = \{a_1(v), a_2(v), \dots, a_Q(v) : a_i(v) \neq a_j(v), \forall i \neq j; a_i(v) \in C, \forall i\}$ where $a_i(v)$ represents the channel assigned to i^{th} radio interface at node v .

The topology defined by \mathcal{A} is represented by $G = (V, E, L, \Lambda)$ in the following natural way: There is an edge $e = (u, v, k)$ on channel $\lambda(e) = k$ between nodes u and v in G *if and only if* $d(u, v) \leq r$ (*i.e.* $\text{edge}(u, v) \in E_T$) and $\lambda(e) \in \mathcal{A}(u) \cap \mathcal{A}(v)$. The latency of the edge e is the latency of the fastest transmission rate supported on e . The set L contains the latency of each edge in E ; similarly the set Λ

contains the channel used on each edge in E . Note that G may be a *multi-graph*, with multiple edges between the same pair of nodes, when the node pair shares two or more channels. We use the same notation to refer to *vertices and nodes*, to *edges and links*, and to *weight of edges and latency of links* without confusion, the usage being clear from the context.

It is assumed 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 in our current work: 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 *at least* a 1-connected topology (*i.e.* a connected topology).

Interference Model: We use a generalized conflict graph based on transmissions to model the effects of wireless interference between different multicast transmissions in MR²-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.

IV. PROBLEM STATEMENT

Objective: The MLB problem is formally defined as follows. Let the connectivity graph $G = (V, E, L, \Lambda)$, number of radio interfaces (Q), number of channels (C), channel assignment \mathcal{A} to all Q interfaces at each node $v \in V$, and the broadcast source node s be given. The objective is to construct a spanning tree $T_{MLB} = (V, E', L', \Lambda')$ (where $E' \subseteq E, L' \subseteq L, \Lambda' \subseteq \Lambda$) that minimizes the broadcast delay, found after interference-free transmission scheduling, to reach all broadcast nodes. For an arbitrary tree T , let the time that $v \in V$ receives the broadcast from the root node (the source) be $\delta_T(v)$. If there are J possible spanning trees of G , then T_{MLB} is the tree that reaches all broadcast nodes with least delay, amongst all J spanning trees of G as shown below:

$$T_{MLB} = \arg \min_{T_j, \forall j=1, \dots, J} (\max(\delta_{T_j}(v_i), \forall v_i \in V))$$

A. Hardness results for the problem

The NP-hardness of the MLB problem for MR²-MC meshes can be established by following from the fact that the MLB problem is NP-hard for specific instances of MR²-MC scenario *i.e.* for the SR-SC single-rate case [10] and for the SR-SC multi-rate case [7]. Thus the MLB problem for MR²-MC WMNs is at least NP-hard.

V. HEURISTIC TO CONSTRUCT MLB TREE IN MR²-MC MULTIRATE MESH

In this section, we present heuristic algorithms to create efficient delivery trees for broadcasting in a MR²-MC WMNs. Since the channel assignment is performed independently of our framework, the topology defined by the channel assignment process \mathcal{A} is an input to our framework. Broadly speaking, any heuristic algorithm designed to solve the MLB tree in MR²-MC meshes must make three important decisions at each node. Firstly, it has to decide whether a node should transmit (*i.e.*, be a non-leaf node in the broadcast tree) or not, and if so, whether the transmission should occur over all or some of its radio interfaces. Secondly, the number of transmissions the node will actually make *must be determined according to the number of radio interfaces and channels available*, alongside the nodes covered in each of these transmissions. Lastly, the transmissions at each node must be scheduled to minimize the broadcast delay after due consideration of radio interference and the number of interfaces available.

MLB is a combination of many closely inter-related hard problems *e.g.* minimum latency tree construction, interference free transmission scheduling and the choice of rate and interface to use for transmissions are all intertwined sub-problems of the overall MLB problem. With the complexity of the problem in mind, we have decomposed our solution into three logically independent steps:

1) *Topology Construction:* The aim of this step is to compute a broadcast tree (or a spanning tree) T of the given topology that exploits the WBA, the multi-rate 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 in this stage. However, the decision on the number of distinct-rate transmissions (either one or more) on any particular chosen interface is deferred to the next step.

2) *Downstream Multicast Grouping:* The aim of the ‘multicast grouping’ algorithm is to take the spanning tree constructed during the ‘topology construction’ stage and *determine both the rates and number of distinct-rate transmissions that each interface should perform*. Intuitively, the rationale behind multiple transmissions is to allow faster transmission to the more *critical* child nodes—*i.e.*, those nodes that have leaf nodes with larger delivery latencies—at the expense of larger transmission latency to the other child nodes.

3) *Transmission Scheduling:* While the number of transmissions at each non-leaf node of the tree is determined after ‘topology construction’ and ‘multicast grouping’, the exact timing of the various transmissions especially relative to different branches of the tree still needs to be determined. The final step schedules all transmissions while taking into account that a node can only transmit after it has received the packet and interfering transmissions cannot occur concurrently. We are conceptually assuming a centralized scheduler in our current work, and plan to investigate the use of decentralized MAC schedulers in future work.

This decomposition of the overall optimization problem is clearly not optimal. For example, we obtain the multicast transmission sets and the transmission rate associated with each link layer multicast only after the ‘multicast grouping’.

However, a joint optimization is computationally infeasible except for trivially small mesh topologies.

We do not address MAC issues in this work, and assume a centralized idealized scheduler. This is to relax the complex inter-related problems of our considered problem. However, a practical algorithmic solution to the MLB problem must address MAC scheduling in a distributed manner while incorporating the needs of the multicasting algorithm. This is the course of our future work. The results presented in this paper, hence, serve as the lower-bound on the achievable worst-case latency and as a benchmark for future distributed algorithms.

We present four heuristic algorithms for the ‘topology construction’; the first (Section VI-A) does not exploit the WBA, the second (Section VI-B) exploits WBA but not the availability of multiple interfaces on the same node, while the other two (Sections VI-C and VI-D) differ in how they exploit both WBA and the interface diversity on individual nodes. We follow the ‘topology construction’ stage with a broad algorithmic approach for the ‘multicast grouping’ in Section VII. We then present the ‘transmission scheduling’ heuristic in Section VIII.

VI. TOPOLOGY CONSTRUCTION

The common input to each of our ‘topology construction’ heuristic algorithm is the channel assignment defined input topology $G = (V, E, L, \Lambda)$, broadcast source s in V , the set $\mathcal{L} = \{l_1, l_2, \dots, l_k\}$ denoting set of latencies of all possible k transmission rates, and the channel assignments to all interfaces at each node \mathcal{A} .

A. Multi-Radio, Multi-Channel, Shortest-Path Tree (MSPT)

The MSPT algorithm (see Algorithm 1) is used to construct the SPT for MR²-MC wireless meshes. The MSPT algorithm is very similar to the greedy Dijkstra algorithm, and works on the principle of edge relaxation. The MSPT algorithm differs from the general Dijkstra’s algorithm, in that it also has to choose appropriate channels for each link it chooses for the MSPT (since a node pair can have multiple links on distinct channels). The ‘transmission scheduling’ step (discussed next in Section VIII) greatly depends on the channel selections made during the ‘topology construction’. Channel selection, if done poorly without due consideration of radio interference, can dramatically degrade performance even for the same spanning tree.

1) *Algorithm:* The MSPT algorithm starts by initializing the ‘labels’ of all nodes to ∞ . The label of any node represents the ‘cost’ of its *current* shortest path to the source s ; with a label of ∞ indicating the absence of a path. The set \mathcal{R} (representing the nodes, whose shortest paths to s have not been finalized yet) is initialized to contain all nodes in V . The algorithm starts by putting d (the node relaxed at the next iteration) equal to s for the initial round. The basic operation of MSPT algorithm is edge relaxation: if there is an edge from u to v , then the shortest known path from s to u (having cost $label(u)$) can be extended to a path from s to v by adding edge (u, v) at the end. This path will have length $label(u) + l(u, v)$ where $l(u, v)$ is the latency of link between vertices u and v . If this is less than the current $label(v)$, we can replace the current value of $label(v)$ with the new value. After edge

Algorithm 1 MSPT construction

```

1: Input:  $[s, G = (V, E, L, \lambda), \mathcal{L} = \{l_1 \dots l_k\}]$ 
2: Initialize  $label(v_i) = \infty, \forall v_i \in V;$ 
3:  $\mathcal{R} = [1 \dots |V|]; d = s; R = R \setminus \{s\};$ 
4: while  $(V \setminus \mathcal{R} \neq \emptyset)$  do
5:    $N =$  connecting nodes of  $d;$ 
6:    $label_{new} = label;$ 
7:    $label_{new}(N) =$ 
8:    $\min((label(d) + cost(d, N)), (label(N)));$ 
9:    $I \leftarrow$  nodes s.t.  $label_{new}(nodes) < label(nodes)$ 
10:   $P_{MSPT}(I) = d;$ 
11:   $E_{MSPT} = E_{MSPT} \cup edge(d, I)$ 
12:   $L_{MSPT}(edge(d, I)) = l(d, I);$ 
13:  for  $u = 1$  to  $|unique-latency-transmissions|$  at  $d$  do
14:    find all nodes  $I_{lu}$  s.t.  $I_{lu} \in I$  and  $l(d, I_{lu}) = l_u \in \mathcal{L}$ 
15:     $\Lambda_{MSPT}(I_{lu}) =$  least-used channel in the
      conflict graph of the transmission  $I_{lu}$ 
16:  end for
17:   $label = label_{new}; d = \arg \min(label(\mathcal{R}));$ 
18:   $R = R \setminus \{d\}$ 
19: end while
20: Output:  $[P_{MSPT}, L_{MSPT}, \Lambda_{MSPT}, label]$ 

```

relaxation in each round, the set of nodes whose labels are reduced from their former values are referred to as I .

Amongst the nodes in I , those connecting to d on the same latency transmission $l_u \in \mathcal{L}$ are denoted by I_{lu} , and are assigned a single channel if sharing a common channel. The channel chosen is the ‘least-used’ in the conflict graph of this transmission. Thus, MSPT is based on the Dijkstra algorithm and does not explicitly consider the WBA; it only considers the use of a less contended channel among available channels between a candidate node pair. Edge relaxation is applied until all values $label(v)$ represent the cost of the shortest path from s to v . MSPT is mathematically described in Algorithm 1. After $|V| - 1$ rounds, the shortest path from each vertex $v \in V$ to s is determined.

B. Multi-Radio, Multi-Channel, Weighted Connected Dominating Set Tree (MWT)

The MWT algorithm (see Algorithm 2) is an extension to the WCDS algorithm, which is designed for the MLB problem for SR-SC multi-rate networks [7] [11]. In SR-SC multi-rate WMNs, WCDS performs creditably against other low-latency broadcast heuristics, because WCDS considers both: the multi-rate 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.

1) *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 the n which are reachable by the transmission combination (n, l, c) . For any transmission combination (n, l, c) —the

Algorithm 2 MWT construction

```

1: Input:  $[s, \mathcal{A}, C, G = (V, E, L, \Lambda), \mathcal{L} = \{l_1 \cdots l_k\}]$ 
2:  $\mathcal{R} \leftarrow \{s\}$ 
3: while  $(V \setminus \mathcal{R} \neq \emptyset)$  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  $f(n, l, c) = (|N(n, l, c) \setminus \mathcal{R}| \div l)$ )
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:  $A \leftarrow N(\hat{n}, \hat{l}, \hat{c}) \setminus \mathcal{R}$ ;
9:  $P_{MWT}(A) = \hat{n}$ ;  $L_{MWT}(A) = \hat{l}$ ;  $\Lambda_{MWT}(A) = \hat{c}$ ;
10:  $\mathcal{R} \leftarrow \mathcal{R} \cup A$ 
11: end while
12: Output:  $[P_{MWT}, L_{MWT}, \Lambda_{MWT}]$ 

```

quantity $|N(n, l, c) \setminus \mathcal{R}|$ (also represented as A 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 $|N(n, l, c) \setminus \mathcal{R}| \div l$. The priority is defined such to reflect the desire to both include as many nodes as possible in a single transmission, yet keep the transmission rate high (even though a higher transmission rate implies a smaller range, and thus, a smaller set of covered nodes).

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} , which is the least-loaded channel within the conflict graph of the transmission as explained in Section VI-A, is chosen. The algorithm completes its execution when all the nodes have been covered, i.e. when $V \setminus \mathcal{R} = \emptyset$. The algorithm returns the sets P_{MWT}, L_{MWT} and Λ_{MWT} , where $P_{MWT}(v_i)$ is the parent node of v_i , $L_{MWT}(v_i)$ is the latency of the link connecting v_i and $P_{MWT}(v_i)$, and $\Lambda_{MWT}(v_i)$ is the channel used on the link connecting v_i and $P_{MWT}(v_i)$, $\forall v_i \in V$. The MWT is now readily constructed using these sets.

C. Locally Parallelized, Multi-Radio, Multi-Channel, WCDS Tree (LMT)

The development of LMT algorithm, which we discuss in this section, is motivated by the observation that MWT—while taking into account the WBA and multi-rate nature of the underlying medium—does not ‘as readily’ exploit the interface diversity on individual nodes. This observation can be explained more intuitively by noting that MWT is inherently biased, by its priority metric, to include transmissions that cover greater number of uncovered nodes. This metric tends to work well when the number of radio interfaces/channels are small. However, it fails to exploit the increased opportunities for parallel ‘faster’ transmissions (on different orthogonal channels) when the number of interfaces are 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 $\hat{l} > l$ and $c \neq \hat{c}$. Thus we may be able to cover node m for free on an orthogonal channel \hat{c} without paying penalty on delay. This is done by considering node m as a covered node of (n, \hat{l}, \hat{c}) but not (n, l, c) .

Algorithm 3 LMT construction

```

1: Input:  $[s, \mathcal{A}, C, G = (V, E, L, \Lambda), \mathcal{L} = \{l_1 \cdots l_k\}]$ 
2:  $\mathcal{R} = \{s\}$ 
3: while  $(V \setminus \mathcal{R} \neq \emptyset)$  do
4:  $(\hat{n}, \hat{l}, \hat{c}) = \arg \max_{n \in \mathcal{R}, l \in \mathcal{L}, c \in C} f(n, l, c)$ 
5: (where  $f(n, l, c) = (|N(n, l, c) \setminus \{\mathcal{R} \cup RN_{(n, l, c)}\}| \div l)$ )
6: and  $RN_{(n, l, c)} = \cup_{\forall (l_i \in \mathcal{L}) < l, \forall (c_i \in \mathcal{A}(n) \setminus \{c\})} N(n, l_i, c_i)$ 
7: {if multiple  $(\hat{n}, \hat{l}, \hat{c})$  with max  $f$ , choose whose
8:  $\hat{c}$  is least used in conflict graph of  $(\hat{n}, \hat{l}, \hat{c})$  }
9:  $N_{covered} = N(\hat{n}, \hat{l}, \hat{c}) \setminus \{\mathcal{R} \cup RN_{(\hat{n}, \hat{l}, \hat{c})}\}$ 
10:  $A \leftarrow N_{covered}$ ;
11:  $\mathcal{R} \leftarrow \mathcal{R} \cup A$ 
12:  $P_{LMT}(A) = \hat{n}$ ;  $L_{LMT}(A) = \hat{l}$ ;  $\Lambda_{LMT}(A) = \hat{c}$ 
13: end while
14: Output:  $[P_{LMT}, L_{LMT}, \Lambda_{LMT}]$ 

```

1) *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 $(N(n, l, c) \setminus \mathcal{R})$. In LMT, the term ‘neigh covered’ is redefined to be $N(n, l, c) \setminus \{\mathcal{R} \cup RN_{(n, l, c)}\}$ 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 contains nodes eligible to transmit during the next round. Unlike MWT, where all non-covered neighboring nodes $N(\hat{n}, \hat{l}, \hat{c}) \setminus \mathcal{R}$ of the chosen transmission $(\hat{n}, \hat{l}, \hat{c})$ are added to \mathcal{R} ; in LMT, only the nodes in $N(\hat{n}, \hat{l}, \hat{c}) \setminus \{\mathcal{R} \cup RN_{(\hat{n}, \hat{l}, \hat{c})}\}$ are added.

The algorithm completes its execution when all the nodes have been covered, i.e. when $V \setminus \mathcal{R} = \emptyset$. The algorithm returns the sets P_{LMT}, L_{LMT} and Λ_{LMT} , where $P_{LMT}(v_i)$ is the parent node of v_i , $L_{LMT}(v_i)$ is the latency of the link connecting v_i and $P_{LMT}(v_i)$, and $\Lambda_{LMT}(v_i)$ is the channel used on the link connecting v_i and $P_{LMT}(v_i)$, $\forall v_i \in V$. LMT can now be readily constructed from these sets.

D. Parallelized, Approximate-Shortest, Multi-Radio, Multi-Channel, WCDS Tree (PAMT)

The PAMT algorithm, like the LMT algorithm, is adapted from the MWT algorithm, and is designed to be *adaptive* to number of radio interfaces and channels available. The PAMT algorithm is intended as an improvement over the LMT algorithm. The LMT algorithm, 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 possibly take place on an alternative, non-interfering channel in parallel. Such a decision is possible despite the fact that in LMT, nodes always attempt to use ‘fastest’ possible transmitting rates to connect to its neighbors. The following simple example illustrates this idea.

First of all, let us define as the total cost (latency) of the path from a node n to source s as *label* of n . 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

Algorithm 4 PAMT construction

```

1: Input:  $[s, \mathcal{A}, C, G = (V, E, L, \Lambda), \mathcal{L} = \{l_1 \cdots l_k\}]$ 
2:  $\mathcal{R} = \{s\}; \text{label}(s) = 0$ 
3: while  $(V \setminus \mathcal{R} \neq \emptyset)$  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: {if multiple  $(\hat{n}, \hat{l}, \hat{c})$  with max  $f$ , choose whose
6:  $\hat{c}$  is least used in conflict graph of  $(\hat{n}, \hat{l}, \hat{c})$ }


---


7: where  $f(n, l, c)$  is calculated as:
8:  $X = Y_{(n, l, c)} = N(n, l, c) \setminus \mathcal{R}$ 
9:  $\text{label}_{\text{trans}} = \text{label}(n) + l;$ 
10: if  $X \neq \emptyset$  then
11:  $\text{nodes}_{\text{tmp}} = \cup_{(\forall c_{\text{tmp}} \in \mathcal{A}(n) \setminus \{c\}, \forall l \in \mathcal{L})} N(n, c_{\text{tmp}}, l)$ 
12:  $\text{nodes}_p = \text{nodes}_{\text{tmp}} \cap \mathcal{R}$ 
13: for  $x = 1$  to  $|X|$  do
14: for  $y = 1$  to  $|\text{nodes}_p|$  do
15:  $\text{latency}_{\text{node}}(y) = l(\text{nodes}_p(y), X(x))$ 
16:  $\text{label}_{\text{node}}(y) = \text{label}(\text{nodes}_p(y))$ 
17:  $\text{label}_{\text{round}}(y) = \text{latency}_{\text{node}}(y) + \text{label}_{\text{node}}(y)$ 
18: if  $\text{label}_{\text{round}}(y) < \text{label}_{\text{trans}}$  then
19:  $Y_{(n, l, c)} = Y_{(n, l, c)} \setminus \{X(x)\};$  break
20: end if
21: end for
22: end for
23: end if
24:  $X = Y_{(n, l, c)}$ 
25:  $f(n, l, c) = |X| \div l$ 


---


26:  $A \leftarrow Y_{(\hat{n}, \hat{l}, \hat{c})}$ 
27:  $\mathcal{R} \leftarrow \mathcal{R} \cup A$ 
28:  $\text{label}(A) = \text{label}(\hat{n}) + \hat{l}$ 
29:  $P_{\text{PAMT}}(A) = \hat{n}; L_{\text{PAMT}}(A) = \hat{l}; \Lambda_{\text{PAMT}}(A) = \hat{c}$ 
30: end while
31: Output:  $[P_{\text{PAMT}}, L_{\text{PAMT}}, \Lambda_{\text{PAMT}}]$ 

```

n' (assume $\text{label}(n') < \text{label}(n)$) on channel c' , with same latency l_1 . If covered by transmission of n' , nodes in $Y' \subset Y$ have a label of $\text{label}(n') + l_1$. Since $Y' \subset Y$, LMT would 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' .

1) *Algorithm:* The PAMT algorithm is also adapted from the MWT algorithm, like the LMT algorithm. PAMT 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 $Y_{(n, l, c)} = N(n, l, c) \setminus \mathcal{R}$ contains all hitherto ‘uncovered nodes’ that can be covered by this transmission (n, l, c) . The label of this transmission denoted by $\text{label}_{\text{trans}}$ is equal to $\text{label}(n) + l$.

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 4) 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. 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 $V \setminus \mathcal{R} = \emptyset$. The algorithm returns the sets $P_{\text{PAMT}}, L_{\text{PAMT}}$ and Λ_{PAMT} , where $P_{\text{PAMT}}(v_i)$ is the parent node of v_i , $L_{\text{PAMT}}(v_i)$ is the latency of the link connecting v_i and $P_{\text{PAMT}}(v_i)$, and $\Lambda_{\text{PAMT}}(v_i)$ is the channel used on the link connecting v_i and $P_{\text{PAMT}}(v_i)$, $\forall v_i \in V$. The PAMT is constructed from these sets.

VII. MULTICAST GROUPING

The output of the ‘topology construction’ stage is a directed broadcast tree. The non-leaf nodes of the tree represent the transmitting nodes. A non-leaf node can have possibly multiple outgoing edges with different weights. This translates in a ‘physical’ sense into multiple link layer multicasts. These link-layer multicasts with *different* transmission rates can take place simultaneously, *if and only if*, these transmissions take place on orthogonal channels (of-course, multiple outgoing edges, at any node, having same latency weight corresponds to a single transmission due to WBA). In the case of different-latency transmissions on the same channel, a decision has to be made to either retain or discard the lower latency transmission(s). The function of the ‘multicast grouping’ stage is to make this very decision.

The decision is made, while keeping in mind that a ‘slower’ transmission has a wider ‘reach’ and vice versa. This implies that the ‘slowest’ transmission can cover all neighboring nodes, albeit at the cost of increased latency. This trade-off has earlier been studied for the case of SR-SC multi-rate WMNs in our earlier work [11]. The case of SR-SC multi-rate WMNs, where all transmissions take place on the same channel, requires grouping decisions whenever there are multiple *different-latency* transmissions at a node. For the case of MR²-MC WMN, a grouping decision needs to be made *only* when the different-latency transmissions are on the *same* channel; with no restrictions on simultaneous transmission on different channel. Our multicast grouping algorithm for MR²-MC is very similar to the grouping algorithm for the case of SR-SC multi-rate WMN described in [11], the only difference being that in MR²-MC WMN the grouping algorithm is only invoked for the case of different-latency transmissions *on the same channel* (and *not on a different orthogonal channel*) at a node.

In order to find the topology which minimizes the broadcast latency, we must make a number of decisions, including which node is to multicast, and if so, how many times it is to multicast, whom the recipients are and its timing. As stated earlier, the result of the topology construction is a broadcast tree which specifies that the non-leaf nodes of the broadcast

tree will multicast to its child nodes, in possibly multiple transmissions. However, the number of times a transmitting node (i.e. non-leaf node of the broadcast tree) will multicast and the recipients of each multicast still have not been decided. In case where a node multicasts only once, then the recipients will be all its child nodes. For the case where a node is to multicast more than once, a different subset of child nodes will be reached in each multicast such that these subsets together form a partition of the set of child nodes. The aim of the multicast grouping stage is to determine the number of multicasts to be made and their recipients.

We give limited account of our ‘multicast grouping’ algorithm in this paper due to space limitations, and refer the interested reader to [19] for more details of our ‘multicast grouping’ algorithm. However, we point out that a key difference of our multicast grouping algorithm and the grouping algorithm in [7] [11] is that in our grouping algorithm, grouping (and the choice of transmission rates), at each node, is performed at each interface independently, and the downstream latency (called Cardinal Value in [7] [11] of the node is subsequently determined by the maximum of the downstream latency across all interfaces.

VIII. TRANSMISSION SCHEDULING

After ‘topology construction’ and ‘multicast grouping’ has been performed. The ‘transmission scheduling’ algorithm tries to schedule the transmissions to minimize the broadcast delay, whilst ensuring no interfering transmissions are scheduled for simultaneous transmission. The broadcast tree generated after the multicast grouping phase is modeled, as before, by a directed tree $T = (V, E, L, \Lambda)$. The transmitting nodes are represented by branching vertices (i.e. non-leaf nodes) in the tree T . Let us denote the number of transmitting nodes in the network by k , the set of transmitting nodes denoted by $V_b = \{b_1, b_2 \dots b_k\}$, and the set of w different-latency transmissions at any arbitrary node b_i by $\hat{L}_{b_i} = \{b_{i1} \dots b_{iw}\}$. The set B contains all the transmissions in the network, $B = \bigcup_{i=1, \dots, k} \hat{L}_{b_i}$. We model the interference between transmissions in an MR²-MC WMN by using a conflict graph for each channel. The conflict graph for any channel i is represented by $G_{c_i} = (B, E_{c_i})$ and models the interference, on channel i , between the set of transmissions B . The set of conflicting edges, represented as E_{c_i} , contains an edge between b_{ij} and b_{kl} , if and only if, both transmissions are on the same channel i and the transmitter of b_{ij} interferes with the receivers of the transmission b_{kl} or vice versa.

Formally, the transmission schedule is the mapping $\tau : b_{ij} \rightarrow \mathbb{R}, \forall b_{ij} \in B$ which gives the starting time of b_{ij} . 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.
- 3) Two arbitrary transmissions b_{ij} and b_{kl} can be scheduled together on an arbitrary channel z , if and only if, the edge $(b_{ij}, b_{kl}) \notin G_{c_z}$.
- 4) At most one transmission can take place at a time on one interface of any node.

We limit our discussion of the ‘transmission scheduling’ algorithm due to space constraints, but we do point out that the scheduling algorithm presented for the case of SR-SC multi-rate WMNs [7] [11] can be used for scheduling in MR²-MC multi-rate scenario with slight modifications. Especially the interference model, as described in Section III, would have to be applied to ensure that transmissions transmitting on orthogonal channels can be scheduled together. The details of the ‘transmission scheduling’ algorithm is provided in [19].

IX. SIMULATED PERFORMANCE EVALUATION

In this section, we evaluate the performance of our heuristic algorithms via simulations. We consider static wireless mesh networks with N nodes randomly located in a 1200×1200 m² region. The transmission rate/range relationship depicted in Table I is assumed. The interference range is assumed to be 520 m. We have considered three channel assignment schemes in our current work: CCA, VCA and INSTC (discussed earlier in Section II). The effect of the number of nodes in the network, the number of radio interfaces at each node, and the channel assignment strategy is observed on the broadcast latency when using our algorithms. We use the CCA channel assignment scheme for studying the effect of change in the number of nodes and interfaces on the broadcast latency in Section IX-A. Later, we show the effect of different channel assignment schemes on broadcast latency in Section IX-B.

A. Performance of our heuristic algorithms

We present the performance of our heuristic algorithms for the case of a SR-SC multi-rate WMN in Figure 1. The vertical axis shows the broadcast latency of our heuristics normalized against the broadcast latency of the Dijkstra tree with infinite number of Q and C . Since determining the actual optimal is NP-hard, we are using Dijkstra tree performance as a theoretical lower bound on the optimal achievable latency in a corresponding wired network. For the specific case of SR-SC multi-rate WMN—MSPT compares unfavorably to our other three heuristics (Figure 1). These results are similar to those shown in [7]. MWT performs better than MSPT since it considers both the WBA and the multi-rate nature of the mesh (Figure 1). The LMT and PAMT algorithms, both adapted from MWT, can only match and not improve the performance of MWT (Figure 1) in SR-SC multi-rate scenarios, since both cannot find *alternative* channel paths to parallelize transmissions on. Thus for SR-SC multi-rate WMN, the performance of LMT and PAMT is exactly the same as WMT.

For the cases of MR²-MC multi-rate meshes where $Q > 1$, all of our proposed heuristics improve their performance. This is true both for small networks ($N=10$, Figure 2) and for large networks ($N=70$, Figure 3). The Figures 2 and 3 display representative performance of different heuristics for MR²-MC multi-rate meshes across the range of radio interfaces from $Q=2$ to $Q=8$.

The improvement seen in MR²-MC performance can be attributed to two main reasons: Firstly, the usage of MR²-MC minimizes the interference in the network and allows interfering transmissions to be transmitted simultaneously using orthogonal channels. This improvement factor called

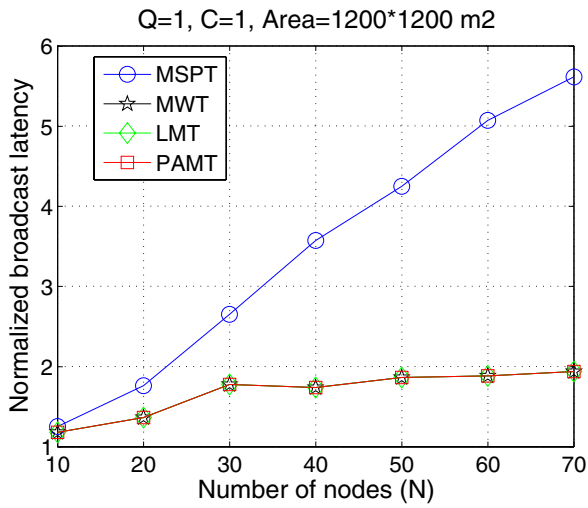


Fig. 1. Normalized broadcast latency against varying number of nodes N for $Q=1, C=1$

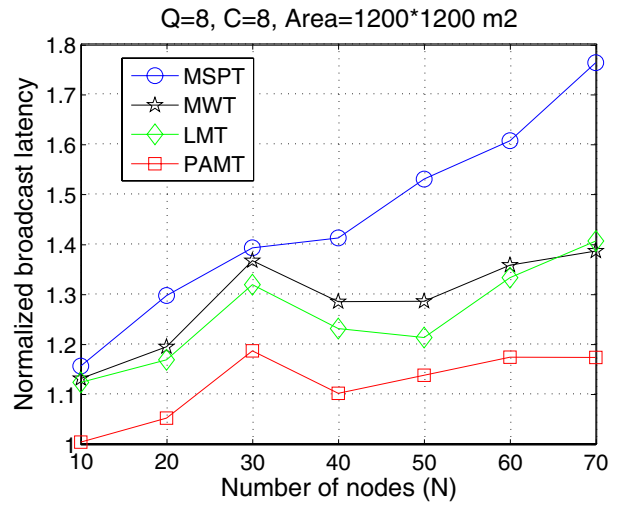


Fig. 4. Normalized broadcast latency against varying number of nodes for $Q=8, C=8$

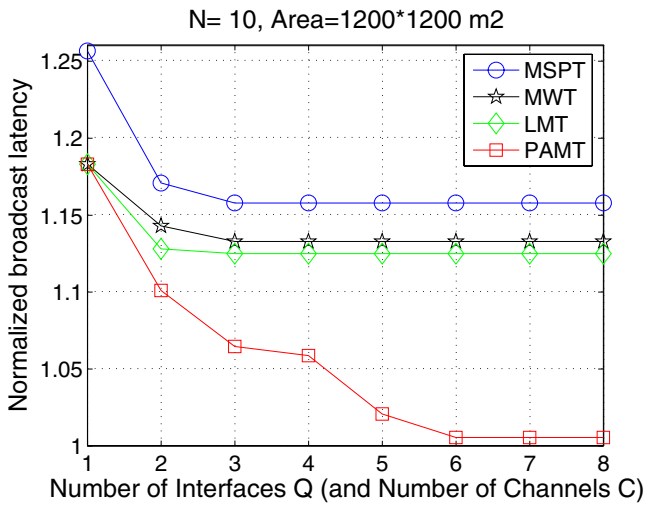


Fig. 2. Normalized broadcast latency against varying number of radio interfaces Q ($C = Q$) with $N = 10$ (Area= $1200 \times 1200 m^2$)

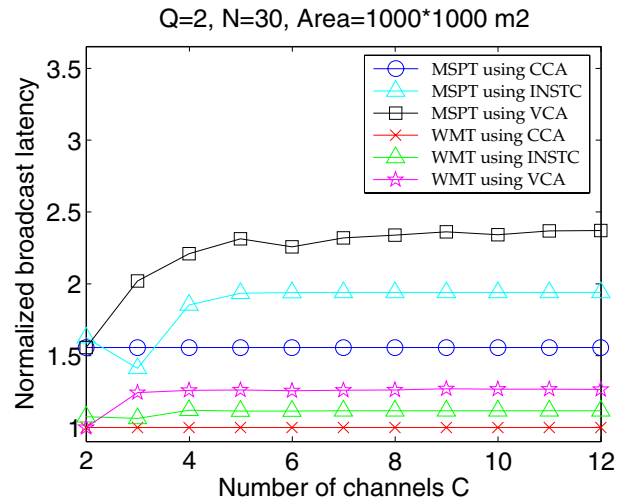


Fig. 5. The impact of channel assignment on MSPT and MWT algorithms for $Q=2, N=30$ and Area= $1200 \times 1200 m^2$)

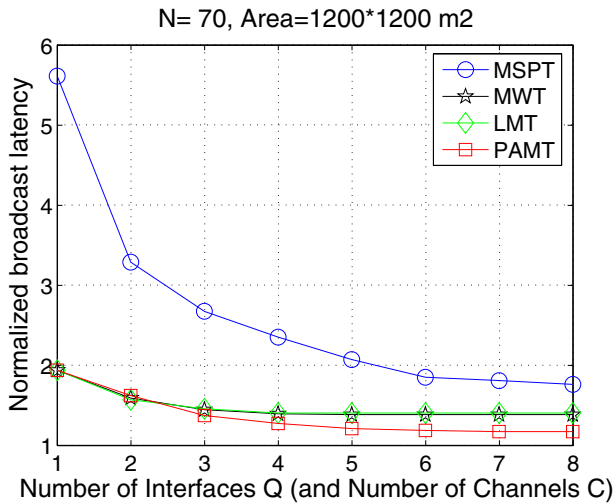


Fig. 3. Normalized broadcast latency against varying number of radio interfaces Q ($C = Q$) with $N = 70$ (Area= $1200 \times 1200 m^2$)

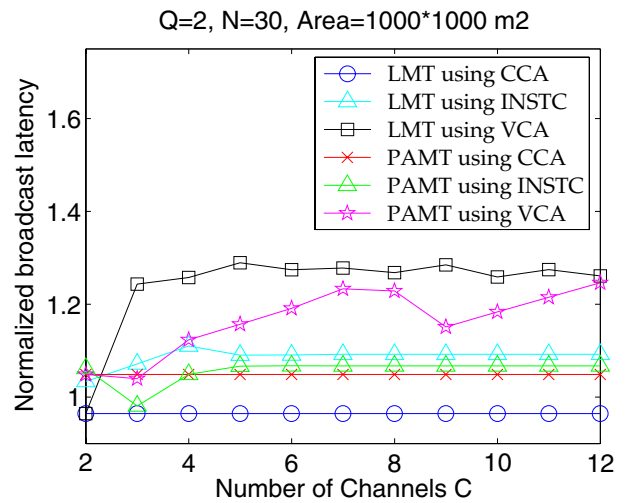


Fig. 6. The impact of channel assignment on LMT and PAMT algorithms for $Q=2, N=30$ and Area= $1200 \times 1200 m^2$)

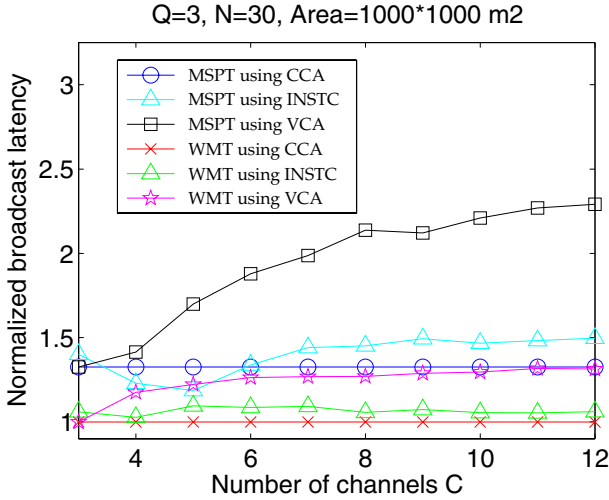


Fig. 7. The impact of channel assignment on MSPT and WMT for 3 Radios, $N=30$ and $\text{Area}=1200 \times 1200 \text{ m}^2$)

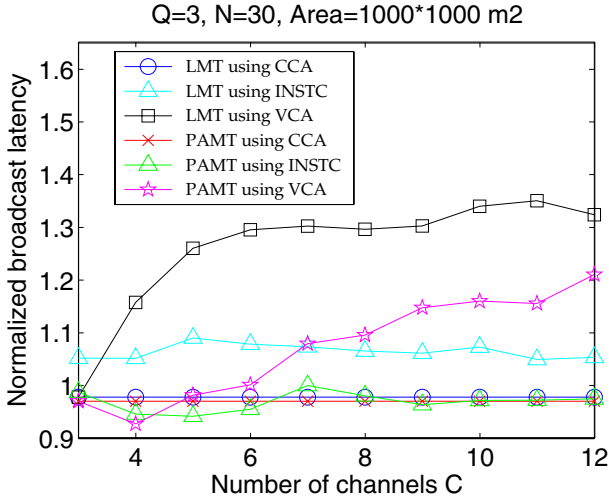


Fig. 8. The impact of channel assignment on LMT and PAMT for 3 Radios, $N=30$ and $\text{Area}=1200 \times 1200 \text{ m}^2$)

‘interference reduction factor’ is general and applies to all our proposed heuristics. The ‘interference reduction factor’ substantially improves performance when the heuristic constructed tree involves many transmissions (e.g. as in MSPT). Secondly, 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 resources available. This improvement factor called the ‘radio adaption factor’ is specific to broadcasting algorithms such as LMT and PAMT.

We will now discuss the performance of each of our heuristic in MR^2MC wireless meshes with increasing Q and C . The performance of MSPT improves with increasing radio resources due to ‘interference reduction factor’—however, in our considered range of nodes (10 to 70) and interfaces (1 to 8), its performance compared to other proposed heuristics is modest (Figures 2 and 3). MSPT’s poor performance is explained by its lack of accounting for WBA during its construction, which in turn implies that too many transmissions are involved in a MSPT. Another reason is its lack of adaption

to the available radio resources. It must be pointed out that although MSPT’s performance is poor in the practical range of values of Q , its performance with the non-practical value of $Q = \infty$ corresponds to optimal achievable performance.

The performance of MWT can be seen in the Figures 2, 3 and 4. It is worth noting that the performance of MWT improves with increasing Q , till a point, beyond which increasing Q does not produce any noticeable gain. Note in the Figures 2 and 3, that although good gains are achieved when increasing the Q from 1 to 3, increasing Q further does not produce any gain. This is because MWT does not parallelize its transmission by adapting to increasing number of interfaces unlike LMT and PAMT. Thus for MWT, like MSPT, only ‘interference reduction factor’ is relevant and the ‘radio adaption factor’ does not apply.

It is interesting that both LMT and PAMT improve upon MWT’s performance when Q and C are increased, as depicted in the Figures 2, 3 and 4. This implies that both these algorithms are adaptive to the available radio resources, and can therefore benefit from both the ‘interference reduction factor’ and the ‘radio adaption factor’. The LMT algorithm is the best performing heuristic for $Q = 2$ and $N = 70$ (Figure 3). In such large networks with limited resources (in the considered case, $Q = 2$), the effect of interference is dominant and the trees that transmit less generally perform better. Since LMT is more conservative than PAMT in adding *parallel* links, it performs slightly better than PAMT in this case.

PAMT is generally the most adaptive of our algorithms to the available Q and C . The broadcast performance of PAMT is very close to optimal for small networks and/or large Q (Figure 2). PAMT also performs consistently well across all ranges of Q and N . Interestingly, PAMT can approach the performance of MSPT with $Q = \infty$ with relatively few radio interfaces in most instances.

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

B. Impact of Different Channel Assignment Strategies

The graphs of the performance of different channel assignment schemes (CCA, VCA and INSTC) for the case of $Q=2$ are shown in Figures 5 and 6, and for the case of $Q=3$ in Figures 7 and 8. The results shown are representative of similar results seen across different values of Q . Due to space constraints, we omit results for other values of Q . The vertical axis in the graphs show broadcast latency of the algorithm normalized against the WMT algorithm with channel 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 are shared amongst 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, with the other interfaces being assigned from channels randomly from the remaining channels in C , the connectivity can suffer at the cost of reduced interference. INSTC, like VCA, reduces interference in the network by increasing channel diversity at the risk of reducing its connectivity and possibly mitigating 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 Figures 5, 6, 7 and 8, 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 the 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 have a more pronounced effect on the performance of MSPT, LMT and PAMT than on WMT, since these algorithms generally involve more transmissions (on possibly interfering channels). As we can see from Figures 5, 6, 7 and 8, 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.*

X. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied the problem of minimum latency broadcasting/multicasting in MR²-MC wireless meshes. We have presented four heuristic algorithms, the first (MSPT) does not exploit the WBA, the second (MWT) exploits WBA but not the availability of multiple interfaces on the same node, while the other two (LMT and PAMT) differ in how they exploit both WBA and the interface diversity on individual nodes. Interestingly, both PAMT and WMT perform fairly close to the theoretical optimal, resulting in latencies that are on average only $\sim 10 - 20\%$ higher.

The simulation results and performance studies 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, the performance of CCA (which generally performs poorly for unicast flows) is generally better than both VCA and INSTC.

The trees constructed by our broadcasting/multicasting algorithms are built per-source. This creates a scalability problem when there are numerous broadcast/multicast sources. We aim

to address scalable low-latency broadcasting/multicasting in WMNs in our future work. In our immediate future work, we plan to modify our heuristics to support QoS-aware broadcasting, where the algorithms consider both the existing traffic load on different interfaces and the data rate of the newly arriving broadcast flow. We shall also study the performance of the algorithms with a distributed (similar to 802.11) contention-based MAC. In the long term, we plan to address the important challenge in WMN design of finding channel assignment strategies that perform well with a mixture of unicast and broadcast traffic.

REFERENCES

- [1] I.F. Akyildiz, X. Wang, W. Wang, I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.
- [2] Bhaskaran Raman and Kameswari Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *Proceedings of 11th Annual International Conference on Mobile Computing and Networking paper (MOBICOM)*, Sep 2005.
- [3] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.
- [4] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Mobicom*, pages 114–128, 2004.
- [5] Ashish Raniwala, Kartik Gopalan, and Tzi cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *Sigmobile Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.
- [6] P. Kyasanur and NH Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. *Wireless Communications and Networking Conference, 2005 IEEE*, 4, 2005.
- [7] C. T. Chou, A. Misra, and J. Qadir. Low latency broadcast in multi-rate wireless mesh networks. *IEEE JSAC special issue on wireless mesh networks*, 2006.
- [8] Chun Tung Chou and Archan Misra. Low latency multimedia broadcast in multi-rate wireless meshes. In *First IEEE Workshop on Wireless Mesh Networks, Held in conjunction with SECON-2005*.
- [9] The status of the IEEE 802.11s standard project. http://grouper.ieee.org/groups/802/11/reports/tgs_update.htm.
- [10] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 222–232, New York, NY, USA, 2003. ACM Press.
- [11] C.T. Chou, A. Misra, and J. Qadir. Low latency broadcast in multi-rate wireless mesh networks. Technical Report <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 and Applications*, 7(6):481–492, 2002.
- [13] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 3:2210–2217.
- [14] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Trans on Mobile Computing*, 1:111, 2002.
- [15] P. M. Ruiz and A. F. Gomez-Skarmeta. Heuristic algorithms for minimum bandwidth consumption multicast routing in wireless mesh networks. In *ADHOC-NOW*, pp. 258–270, 2005.
- [16] C. Gui and P. Mohapatra. Scalable multicasting in mobile ad hoc networks. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 3:2119–2129, 2004.
- [17] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 68–77, New York, NY, USA, 2005. ACM Press.
- [18] Scalable Networks Inc. <http://www.scalable-networks.com/>.
- [19] J. Qadir, C.T. Chou, and A. Misra. Low latency broadcast in multi-radio multi-channel multi-rate wireless mesh networks. Technical Report <ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0608.pdf>, 2006.