Ad Hoc Networks 47 (2016) 99–113

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Interference-aware multicast trees and meshes for wireless multihop networks[☆]

Daniel Lertpratchya, Douglas M. Blough[*]

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0765, United States

ARTICLE INFO

Article history:
Received 29 January 2016
Revised 20 April 2016
Accepted 9 May 2016
Available online 10 May 2016

Keywords:
Wireless multihop networks
Multicast routing
Interference

ABSTRACT

The objective of this paper is to re-evaluate multicast structures for wireless multihop networks by taking interference into account. We group nodes in multicast routing structures into different classes based on their roles in the multicast routing. Through mathematical analyses of these different node classes under an accurate interference model, we derive optimal routing strategies for the different classes in the presence of interference. These analyses inform the design of four interference-aware multicast routing algorithms. We evaluate the performances of our proposed algorithms using wireless network simulations, where we have implemented a detailed model of wireless communication. We show that, by taking interference into account when building the multicast routing structures, our proposed algorithms are able to significantly improve throughput, delay, and robustness compared to existing multicast routing algorithms, which do not consider interference.

1. Introduction

In multicast routing, a single message is routed to multiple destinations in a network. Multicast is an important operation in a variety of wireless network settings. Just a few of the many example uses of multicast are: (1) disseminating network state for updating routing tables and allocating network resources, (2) communicating with groups of users in applications such as emergency response networking, where teams of users must coordinate search and rescue operations, (3) distributing popular live media content to interested parties, and (4) implementing emergency alert services such as Amber alerts and weather emergency information. Multicast protocols for wireless multihop networks are surveyed in [3]. A major limitation of prior research in this area is that the vast majority ignores interference, which is a significant factor in wireless multihop networks and is typically the limiting factor in dense network deployments.

In this paper, we consider the problem of interference-aware multicast routing in wireless multihop networks, using an accurate physical interference model. First, we study the problem for low-intensity multicast, where the packet generation rate is low enough that one multicast packet can be completely delivered to all destinations before another packet is sent by the source. We classify nodes into different classes and derive optimal interference-aware routing strategies for each class. Based on these analyses, we propose a new interference-aware multicast routing structure based on Steiner trees for the low-intensity case. Next, we consider multicast scenarios with higher packet generation rates. In this more general case, we modify our interference analyses to help design a second new multicast routing structure, referred to as the fixed-distance tree merging (FTM) algorithm. In addition to handling higher packet generation rate scenarios, the FTM algorithm does not use global network knowledge, which is required to construct the Steiner tree structure. To address the problem of *robust* interference-aware multicast algorithms, we propose two interference-aware mesh-based multicast algorithms. Interference-aware multicast meshes are built by extending our interference-aware multicast tree designs. Our first algorithm creates a mesh by creating two redundant paths for each overlay link in the multicast tree. The second algorithm uses Delaunay triangulation to build a multicast overlay mesh structure.

We perform formal analyses of interference and use those analyses to drive the design of the aforementioned interference-aware multicast routing structures. In addition to our anaytical evaluations, we evaluate our proposed structures through simulation in both TDMA and CSMA/CA settings. Simulation results demonstrate that, compared to existing approaches, our proposed tree-based structures reduce schedule length by up to 57% in TDMA networks and achieve up to 41% higher goodput in CSMA/CA networks. Moreover, our proposed multicast mesh structures provide up to 80% increase in goodput over existing tree-based routing

[☆] This paper builds on our preliminary work in two conference papers [1,2].

[*] Corresponding author.
E-mail address: dblough@ece.gatech.edu, doug.blough@ece.gatech.edu (D.M. Blough).

http://dx.doi.org/10.1016/j.adhoc.2016.05.001
1570-8705/© 2016 Elsevier B.V. All rights reserved.

structures and up to 45% increase in goodput over existing mesh-based routing structures. The simulations also show that our proposed interference-aware mesh routing structures are robust to bursty wireless links and link failures.

2. Related work

Most multicast routing approaches can be classified into three main categories: tree-based, mesh-based, and structure-less. Tree-based protocols, e.g. [4–7] use different kinds of trees as the routing structure for multicast messages. Trees provide simple and cost effective routing structures but do not handle dynamic network topologies, e.g due to mobility and link failures, well. Mesh-based protocols, e.g. [8–10] provide robustness by having multiple routes between the source and different destinations at the cost of maintaining the mesh structure. Structure-less multicast protocols do not create an explicit routing structure but rely on other methods such as network coding [11,12] and geographic routing [13,14]. In this paper, we focus on tree-based and mesh-based multicast protocols and consider how to account for wireless interference in their design.

Tree-based protocols have focused mainly on shortest path trees or Steiner trees. The goal of shortest path trees (e.g. [5,6]) is to minimize the distance from the source to the destinations, while Steiner trees (e.g. [15,16]) attempt to minimize the sum of the distances in the multicast tree. A few studies comparing multicast tree structures have been done. Ruiz and Gomez–Skarmeta [17] studied shortest path trees and Steiner trees. The authors argued that Steiner trees are not appropriate for wireless networks and proposed that the problem should instead be formulated to minimize the number of forwarding nodes in the multicast structure. They proposed a greedy heuristic algorithm, called MNT, and showed that the proposed algorithm is able to reduce the number of forwarding nodes compared to earlier multicast structures. Nguyen [18] revisited the question and evaluated the packet delivery ratio of shortest path trees, Steiner trees, and the MNT structure. The author showed that shortest path trees offer the best performance in terms of packet delivery ratio. Neither of these comparative studies accounted for interference in their evaluations.

Tree structures provide simple and cost effective routing infrastructures but they sacrifice robustness, because they provide only one path between the source and each destination. A technique that has been proposed to improve robustness is to use a mesh as the underlying routing structure instead of a tree. A mesh is a connected graph where there is more than one path from a multicast source to each multicast destination. These extra paths can deliver multicast packets to the destinations if transmissions on other paths fail. A few studies have been done on theoretical aspects of multicast mesh structures [19,20]. Zhao, et al. [19] proposed four heuristics to build a resilient multicast mesh structure. Two heuristics, NDT and RNDT, build a multicast mesh by merging two node-disjoint MNT multicast trees [17] to form a multicast mesh. The other two heuristics, SDM and MDM, build a multicast mesh by finding a pair of node-disjoint shortest paths from the source node to each multicast receiver, then merging all the node-disjoint paths to form a multicast mesh. Neither of these studies of multicast mesh structures take interference into account.

Some other prior works are primarily concerned with the asymptotic multicast capacity of multihop wireless networks, e.g. [21–23]. Most of these works do not consider specific multicast topologies, but in [23], Shakkottai, et al. do propose a comb structure for multicast trees that achieves capacity in the order sense. While this work does account for interference, it uses the protocol model for interference, which is not as accurate as the physical interference model. Furthermore, the work was concerned primarily with asymptotic scaling results, rather than best performance on finite networks.

A number of studies consider the multicast problem with different goals such as energy [24–26] or the cost of building and maintaining multicast trees [27]. We do not study these issues in our paper. Two papers do account for interference in some manner [28,29]. In [29], directional antennas are used to reduce the effects of interference while considering the multicast problem. While directional antennas eliminate or dramatically reduce interference in certain directions, they do not reduce it in the intended direction of transmission. When directional antenna beams overlap, [29] assumes behavior is governed by the protocol interference model, which is a quite simplified model as compared to the SINR-based physical interference model used in this paper. We apply a more accurate interference model to general tree and routing multicast structures without assuming special hardware such as directional antennas. In [28], the authors consider the effects of interference on location-based multicast in sensor networks. The approach has each forwarding node sense the environment to estimate the current level of interference and then adapts transmission power to account for the interference level. In our work, we analyze the precise interference patterns of multicast traffic in order to optimize multicast structures, rather than generally sensing background interference levels.

Scheduling is an important aspect of wireless multihop networks with interference. Scheduling can improve network performance by having devices access the channel in an orderly fashion instead of using more conservative contention-based access methods. The only multicast scheduling works of which we are aware are [30,31]. However, [30] is primarily concerned with power control and scheduling plays only a minor role, while [31] only deals with one-hop (not end-to-end) multicast.

Finally, connected dominating sets are often used as general wireless backbones, which can support a variety of functions including multicast [32,33]. However, when specifically considering multicast, dominating sets provide neither minimal forwarding (as do trees) nor guarantees of robustness (as do mesh structures). Since we are interested solely in the multicast problem herein, we focus on tree and mesh structures that are specifically designed with multicast in mind. We believe some of the principles we report on in the paper could be applied to interference-aware multicast with connected dominating sets as well. However, since those approaches require good performance across a wide variety of different network functions, we do not evaluate modifications to them for multicast since they might be at cross-purposes for other uses of the dominating set structure.

3. System model and problem formulation

We consider a communication graph $G = (V, E)$, where V is the set of wireless nodes. An edge $(u, v) \in E$ if and only if nodes u and v have a strong enough signal between them to communicate with each other in the absence of interference.[1] We are given a multicast source $s \in V$ and a set of multicast destinations $M \subset V$. The first problem we consider is to construct a multicast tree rooted at s that spans M and a schedule S of the links of the tree. A schedule is defined as a partition of the links of the multicast tree $S_1, S_2, \ldots, S_k$, where S_i is a set of feasible transmissions with the given interference model. A feasible transmission set is defined as a set where, if all nodes in the set transmit to their respective receivers at the same time, all receivers will successfully receive the transmissions under the given interference model. Our

[1] That is, the SNR at each node, when acting as receiver for the other node's transmission, is above the threshold for reception.

goal in this first problem is to construct a multicast tree and feasible transmission schedule of minimum length, where the length of $S_1, S_2, \ldots, S_k$ is k. Constructing a feasible schedule of minimum length ensures that the multicast is robust to interference while at the same time maximizing throughput. In the second problem that we consider, we relax the constraint that the multicast structure has to be a tree. For this problem, we specifically consider mesh structures, where there are multiple paths from the multicast source to each of the multicast destinations. The goal remains to maximize throughput while accounting for interference.

We adopt the classical model for radio signal propagation, which is referred to as the log-distance propagation loss model. The radio signal strength at a distance d from the transmitter is given by $\frac{P}{d^{\alpha}}$, where P is the transmission power and α is the path loss coefficient. We assume that nodes are not equipped with interference cancellation capabilities. To model interference, we adopt the physical interference (PI) model [34], which is known to be more accurate than binary interference models such as protocol interference [34] or hop-based interference [35]. In the PI model, interference from all concurrent transmitters in the network, no matter how distant, is factored into the signal-to-interference ratio (SIR) value at the receiver. Specifically, the transmission will be correctly received by the receiver if and only if the SIR value at the receiver is larger than the SIR threshold (SIR_{min}).

In our multicast routing, whenever a packet reaches a branching point, we assume that it is forwarded to the next nodes along all branches simultaneously using 1-hop wireless broadcast. In our analyses, we do not consider retransmissions since basic one-hop broadcast is typically done unreliably without acknowledgements and retransmissions. However, in our simulation results, we do use a reliable 1-hop broadcast mechanism, which produces retransmissions in some cases. Although this will cause some deviations from the analytical results, we show that the algorithms we develop based on those analyses still work extremely well even in the presence of retransmissions.

4. Interference-aware multicast routing trees

In this section, we study the problem of building interference-aware multicast routing trees. Multicast routing trees provide a minimal multicast routing structure for a source node to send multicast messages to multiple destinations. We first study the problem of multicast routing trees for low-intensity multicast before proceeding to the general multicast problem.

4.1. Interference-aware multicast routing trees for low intensity multicast

In this section, we consider low intensity multicast, which is defined as a situation where the time between two consecutive packets generated by the source is greater than the time it takes to deliver one multicast packet to all destinations. We first consider an ideal network where we are given a source node s and a set of destination nodes M and we are able to place extra nodes anywhere in the network when building a multicast tree.[2] Our goal is to construct a tree rooted at s that spans M while taking interference into account. To achieve this, we classify nodes in a multicast tree into three classes.

- Leaf nodes – nodes with no children.
- Path nodes – nodes with exactly one child.
- Branching nodes – nodes with more than one child.

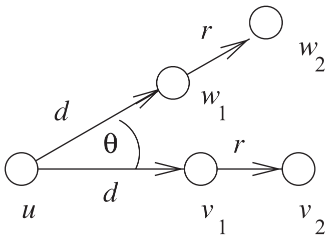


Fig. 1. A branching node u branching into two paths.

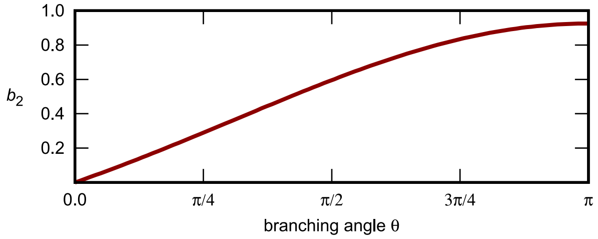


Fig. 2. Values of b_2 with different branching angle θ.

According to our classification, leaf nodes do not forward data in the multicast tree and so they do not generate interference. Furthermore, along a single path, there is no interference between nodes since there is only one packet being transmitted at a time. Thus, the optimal structure along a single path is for each transmission to travel as far as possible, i.e. to separate consecutive nodes by the maximum transmission range. Next, we determine optimal structures involving branching nodes, which are non-trivial to analyze.

Because our algorithm is based on Euclidean Steiner trees, which have a maximum degree of 3, branching nodes can either have two or three children. In the following subsections, we analyze these two cases separately.

4.1.1. Branching nodes with two children

We first consider a branching node u with two children v_1 and w_1 as shown in Fig. 1. The distance of the first hop from u to v_1 and u to w_1 is d, where $d \leq r_t$, which is the maximum transmission range. The first transmission from u to v and w is done by using a multicast or broadcast. The second transmissions from v_1 to v_2 and from w_1 to w_2 occur at almost the same time. Let r be the distances from v_1 to v_2 and from w_1 to w_2, our goal is to select r such that both transmissions will be successful. The cross interference from the concurrent transmission is given by

$$\frac{P}{[r^2 + 2d(1-\cos\theta)r + 2d^2(1-\cos\theta)]^{\alpha/2}}.$$

Combining the received signal strength and the interference, we set SIR to SIR_{min} and convert to decibel to get

$$0 = \left(1 - 10^{\frac{\text{SIR}_{\text{min}}^{\text{dB}}}{5\alpha}}\right) r^2 + 2d(1-\cos\theta)r + 2(1-\cos\theta)d^2.$$

Solving the equation, we get $r = b_2 \cdot d$ where

$$b_2 = \frac{1 - \cos\theta + \sqrt{(\cos\theta - 1)(1 - 2\cdot 10^{(\frac{\text{SIR}_{\text{min}}^{\text{dB}}}{5\alpha})} + \cos\theta)}}{-1 + 10^{(\frac{\text{SIR}_{\text{min}}^{\text{dB}}}{5\alpha})}}.$$

The result shows that the distance between nodes after the branching point is proportional to the distance of the first hop. We show the values of b_2 at different branching angle θ in Fig. 2. The figure shows that if $\theta = \pi$, i.e. the branching paths go in opposite directions, that the second hops can almost achieve the maximum

[2] Alternatively, we can think of an extremely dense network, where there will always be a node that is extremely close to the ideal location for a relay node, which we can select.

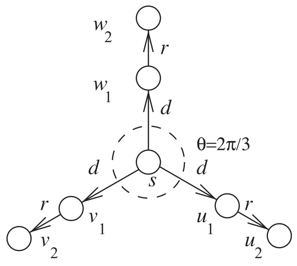


Fig. 3. A branching node s branching into three paths.

transmission range. However, as θ decreases, the two paths interfere more with each other and the range of the second hops decreases, becoming only around 30% of the maximum transmission range when $\theta = \frac{\pi}{4}$.

Let i be the depth from u and r_i be the distance between a node at depth i and a node at depth $i+1$. For all $i > 0$, we can consider d as a sum of all r_j, where $0 \le j < i$, we get

$$r_i = b_2 \sum_{j=0}^{i-1} r_j = b_2 (b_2+1)^{i-1} r_0, \text{ where } r_i \le r_t \text{ and } r_0 \le r_t$$

This shows that the distance between v_i and v_{i+1} grows as v_i gets further away from u until the distance reaches r_t, which is the maximum transmission range and corresponds to the optimal distance for the single-path case.

4.1.2. Branching nodes with three children

Next, we consider a branching node s with three children u_1, v_1, and w_1 as shown in Figure 3. Applying a similar analysis to the two-child case, we get $r = b_3 \cdot d$ where

$$b_3 = \frac{3 + \sqrt{9 - 12(1 - 10^{\frac{10\log 2 + \mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}})}}{2(10^{\frac{10\log 2 + \mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}} - 1)}.$$

Again, the distance between nodes grows as nodes get further away from the branching node, albeit in a slightly different manner. Here also, the distance will eventually reach the limit r_t.

Using the preceding analyses, we present an approximation algorithm to build a multicast tree. The algorithm applies different routing strategies depending on which is the applicable case (single path, two-way branch, or three-way branch).

4.1.3. Interference-aware Steiner tree algorithm (IAST)

Since the optimal routing strategies are different for different classes, our goal is to use different routing strategies for different classes when building a multicast tree. However, one of the difficulties of applying the approach is identifying where branching should take place. Our goal is to first identify branching nodes locations and use different routing strategies for different classes when building a multicast tree.

The high level idea of the IAST algorithm is as follows: we are given nodes that must be connected in a multicast tree. These nodes are the source node and all the destination nodes. The first step is to identify how these nodes should be connected in a tree. The algorithm uses a Euclidean Steiner Tree approximation algorithm to locate ideal Steiner nodes in the network, using $M \cup \{s\}$ as input, where M represents the set of multicast destinations. We call the returned Steiner Tree a Steiner Overlay Tree since it shows the "big picture" connections between nodes in the network. An edge between two nodes in the Steiner Overlay Tree implies that the two nodes should be connected by a path in the final multicast tree. Note that if we view $M \cup \{s\}$ as a multicast group, the group will have the same Steiner Overlay Tree regardless of which node is considered to be the source node.

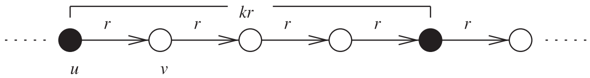


Fig. 4. An infinitely-long, equally-spaced linear network.

For each Steiner node, the algorithm finds a real node nearest to the ideal Steiner node location to act as the Steiner node. The algorithm consults the Steiner Overlay Tree to determine if the current node should be considered as the source node, a path node, or a branching node. The algorithm uses different distances for different node classes based on the analyses in Sections 4.1.1 and 4.1.2. Note that Euclidean Steiner trees contain only two-way branches and up to one three-way branch (at the source), meaning that our analyses are sufficient to handle all cases.

To construct the final multicast tree, intermediate nodes are filled in along the overlay links between Steiner nodes. This is done by choosing intermediate node locations as close as possible to the ideal locations based on the optimal distances derived in Sections 4.1.1 and 4.1.2.

4.2. Interference-aware multicast routing trees for general multicast

In this section, we consider a more general multicast scenario where the source node may begin transmission of the next packet before all receivers have received the previous packets. As a result, there may be more than one application layer packet being forwarded in the network. Multiple application layer packets being forwarded in the network means that interference in the network will be higher than in the low intensity multicast case.

Since more than one application layer packet may be present in the network, we are unable to directly apply the analyses from Section 4.1. The extra interference means that the distances between nodes must be shortened to allow for concurrent transmissions. To tackle this problem, we introduce a scaling factor, f, to be used to scale down the distances derived in Section 4.1. Our goal is to find an appropriate value for the scaling factor f.

4.2.1. Scaling factor in path nodes

Consider a linear path in one dimension with infinite length as shown in Fig. 4. If all nodes are equally separated by a separation $r = r_t$, then all links in the path are on the edge of the SINR threshold and no concurrent transmission is possible. The schedule length in this case will be on the order of the length of the path. This phenomenon is known as the black-gray link paradox [36]. An edge with distance exactly equal to the transmission range is referred to as a "black" link, and such an edge does not allow a single concurrent transmission, no matter how distant. An edge with distance slightly below the transmission range is referred to as a "gray" link. Gray links do allow concurrent transmission, although the spatial separation between them must be quite large.

Given the black-gray link paradox, if the separations between nodes are scaled down to $r = f \cdot r_t$, where $0 < f < 1$, it should be possible to have multiple nodes transmit concurrently. If the schedule length is k, then two transmitters assigned to the same times slot will be separated by a distance $k \cdot r$. Our goal is to find an appropriate value of f such that all transmissions in a same slot will be successful. Consider a transmission from node u to node v, the total interference experienced by v is given by

$$\sum_{i=1}^{\infty} \left[\frac{P}{(ikr+r)^{\alpha}} + \frac{P}{(ikr-r)^{\alpha}} \right].$$

The SIR at the receiver v can be obtained by combining the received signal strength and the total interference at v, the SIR at v

is given by the equation

$$\mathrm{SIR}(v) = \frac{1}{\sum_{i=1}^{\infty}\left[\frac{1}{(ik+1)^{\alpha}} + \frac{1}{(ik-1)^{\alpha}}\right]}.$$

The transmission will be correctly received by v if and only if $\mathrm{SIR}(v) \geq \mathrm{SIR}_{\min}$. Thus, we get the following inequality:[3]

$$\sum_{i=1}^{\infty}\left[\frac{1}{(ik+1)^{\alpha}} + \frac{1}{(ik-1)^{\alpha}}\right] \leq \mathrm{SIR}_{\min}^{-1}. \qquad (1)$$

We use the convergence integral test to the left hand side of (1) and get

$$\frac{1}{(k+1)^{\alpha}} + \frac{1}{(k-1)^{\alpha}} + \frac{(k+1)^{1-\alpha} + (k-1)^{1-\alpha}}{k(\alpha-1)} \leq \mathrm{SIR}_{\min}^{-1}. \qquad (2)$$

Eq. (2) shows that according to SIR, the optimal schedule length of an infinitely-long equally-spaced linear network depends only on the path-loss coefficient and the $\mathrm{SIR}_{\min}$.

Unfortunately, this analysis does not provide a simple way to compute the optimal transmission distance and corresponding scaling factor. However, the analysis shows that as the scaling factor is reduced, there will be a point at which the optimal schedule length is reached and further reductions in scaling factor will not provide any additional benefit. To address the choice of scaling factor, we perform experimental evaluations, which are reported in Section 6.2.2.

4.2.2. Fixed-distance tree merging algorithm

The IAST algorithm requires global knowledge of the network to approximate a Euclidean Steiner tree and identify candidates to act as Steiner nodes. Our second algorithm, the fixed-distance tree merging (FTM) algorithm, does not require global knowledge of the network. The algorithm grows the source tree by merging the source tree with the nearest destination node until all destination nodes are connected.

The high level description of FTM algorithm is as follows. The algorithm uses the scaling factor concept introduced for the IAST algorithm in the general multicast scenario. The preferred scaling factor (f_p) is an input to the algorithm, where $0 < f_p \leq 1$. The algorithm uses a common distance $r = f_p \cdot r_t$ when building a multicast tree. Using this common distance, the algorithm finds shortest paths, in terms of hop counts, from each multicast destination to the source and chooses the closest destination to the source as a starting point. The algorithm then proceeds to build the multicast tree by adding one destination at a time into the existing tree. The algorithm grows the tree from the existing tree towards the next destination node being added by using the following criteria to select the next hop node.

1. The next hop node *must* be closer to the destination node being added.
2. If there exist multiple next hop node possibilities, the algorithm picks the next hop node that is closest to all other remaining destination nodes.

After a destination node is merged with the existing tree, the algorithm selects the next destination node nearest to the new tree and grows the tree toward the selected node until all destination nodes are included in the tree. In the case where node density is too low and growing the tree with $r = f_p \cdot r_t$ is not possible, FTM gradually increases r until the selected node is successfully merged or $r = r_t$.

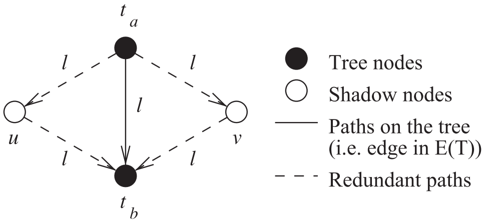


Fig. 5. Overlay link extension algorithm.

4.2.3. Scheduling algorithm

To accommodate routing trees with different distances for different node classes, we propose a modified version of the Greedy-Physical scheduling algorithm [37], called Tree-Based GreedyPhysical. The basic GreedyPhysical algorithm constructs a schedule one slot at a time and simply packs as many links as possible into the current slot without violating the SIR requirement at any receiver. In Tree-Based GreedyPhysical, we group links together that are at the same depth in the multicast tree. We then schedule as many nodes at the same depth as possible into the same slot until no more nodes at that depth can be accomodated. After no more nodes at the same depth can be added to a slot, Tree-Based GreedyPhysical reverts back to the original GreedyPhysical, where as many additional links as possible are added to the slot in a greedy fashion, without accounting for their tree depth and while maintaining the $\mathrm{SIR}_{\min}$ threshold for each receiver in the slot.

5. Interference-aware multicast routing meshes

In this section, we present algorithms to build interference-aware multicast mesh structures. The algorithms build a mesh by creating redundant paths on an interference-aware Steiner tree (IAST) [1] to form an interference-aware multicast mesh. The goal of the algorithms is to reduce the impact of interference along the paths in the mesh.

5.1. Overlay link extension algorithm

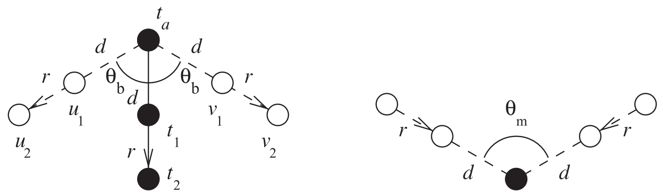
Our first tree extension algorithm is called the overlay link extension algorithm (OLE). Given an overlay tree T, such as the Steiner Overlay Tree from Section 4.1.3, this first mesh algorithm creates two redundant paths for each edge that connects two multicast destinations in the overlay tree. For each such edge (t_a, t_b) $\in E(T)$, OLE creates two redundant paths between t_a and t_b. OLE places two shadow nodes u and v and creates two paths between t_a and t_b, one through each shadow node. Let $d(t_a, t_b) = l$, OLE places u and v such that $d(t_a, u) = d(t_a, v) = d(u, t_b) = d(v, t_b) = l$. The general idea of our first algorithm is illustrated in Fig. 5.

By introducing two redundant paths between t_a and t_b, we have created one mesh branching point at t_a and one mesh merging point at t_b where the two redundant paths merge to t_b. Next, we determine optimal structures involving the branching point and the merging point in an ideal network.

5.1.1. Mesh branching nodes

Consider a mesh branching node t_a that branches into three nodes – one node on the overlay tree (t_1) and two nodes on the redundant paths (u_1 and v_1) as shown in Fig. 6(a). Let $d = d(t_a, t_1) = d(t_a, u_1) = d(t_a, v_1)$ and $r = d(t_1, t_2) = d(u_1, u_2) = d(v_1, v_2)$. The transmission from t_a is done by broadcasting to all three children. Assuming that all three nodes successfully received the packet from t_a, the three nodes forward the packet to their next hops at the same time. Our goal is to find r such that the three transmissions will be successful. Among the three receivers, the node t_2 on

[3] We assume that $\mathrm{SIR}_{\min} > 0$, where $\mathrm{SIR}_{\min}$ is in a linear ratio.



(a) a branching point (b) a merging point

Fig. 6. Two scenarios to be considered for OLE.

the overlay tree experiences the largest interference. The total interference at t_2 is given by

$$\frac{2P_t}{(d^2+dr+r^2)^{\alpha/2}}.$$

Combining the received signal strength and the interference, we set SIR to $\mathrm{SIR}_{\min}$ and convert units to decibel, we get $r = b \cdot d$ where

$$b = \frac{1+\sqrt{1+4(10^{\frac{10\log 2+\mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}}-1)}}{2(10^{\frac{10\log 2+\mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}}-1)}.$$

The result shows that the distances between the transmitting nodes and the receiving nodes are proportional to the distance between the transmitting node and the mesh branching node. The distances grow as the transmitting node gets farther away from the mesh branching node until the distance reaches the maximum transmission range.

5.1.2. Mesh merging nodes

Next, we consider a mesh merging node (Fig. 6(b)), where the redundant paths merge back to the node on the overlay tree. Since the length of the tree path is l and the length of the redundant path is $2l$, the packet on the tree path will have already arrived at the merging node when the packets on the two redundant paths arrive. Applying similar analysis as for the mesh branching nodes, we get $r = m \cdot d$ where

$$m = \frac{3+\sqrt{12\cdot 10^{\frac{\mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}}-3}}{2(10^{\frac{\mathrm{SIR}_{\min}^{\mathrm{dB}}}{5\alpha}}-1)}.$$

Here, the distances between nodes must shorten as the nodes get closer to the merging node. However, the last transmissions to the merging node cannot take place at the same time. The transmissions to the merging node may need to be scheduled to avoid collision.

5.1.3. Additional details for OLE algorithm

After the overlay mesh in an ideal network is formed, the algorithm builds a multicast mesh routing structure using the overlay mesh as a guideline. Since the real network is finite, it is not possible to find nodes that are exactly at the shadow nodes' locations. The algorithm first finds the node that is nearest to a shadow node location and assigns the selected node as the shadow node. There will always be some nearest node that can be selected as shadow node. However, if the network is sparse, it is possible that the same node will be selected as shadow node for multiple primary nodes. In this case, if packets are forwarded through the alternate paths at different times, then this has no impact on performance. However, if packets from different alternate paths are forwarded to the shadow node at around the same time, MAC-layer mechanisms will resolve the contention and operation will deviate somewhat from our analytical evaluation. While we did see this situation arise occasionally in our simulations, we noticed no significant impact on performance due to this phenomenon.

Next, the algorithm builds a mesh by connecting nodes with paths, using the analyses for the mesh branching nodes and mesh merging nodes to select nodes on the paths. Again, since the real network is finite, finding nodes that at exactly the locations determined by the analyses is not possible. Hence, in this situation, the algorithm again chooses nodes as close as possible to the desired locations.

One advantage of the OLE algorithm is that the analyses can be used as guidelines to select nodes on the final multicast mesh. However, the OLE algorithm creates two redundant paths for each overlay link between two multicast destinations. This can produce a large number of additional links in the multicast structure, which also tends to increase interference. In the next subsection, we present an alternative mesh structure that adds redundant paths between multicast destinations while attempting to limit the number of additional links created.

5.2. Delaunay mesh extension algorithm

In this section, we propose our second algorithm to extend a multicast tree to form a mesh, called the Delaunay mesh extension algorithm (DME). Our goal in designing the second algorithm is to take the *overall* tree structure into account when building a multicast mesh.

5.2.1. Basic Delaunay mesh extension

The idea of the Delaunay mesh extension algorithm is to use Delaunay triangulation on the set of overlay nodes in the overlay multicast tree to identify the positions of the mesh nodes (also called Delaunay nodes). Given a Delaunay triangulation, the algorithm identifies a center point of each triangle as a position of a potential mesh node. The algorithm creates redundant paths between a potential mesh node and each of its corresponding nodes in the Delaunay triangulation if the overlay node is not a Steiner node. We do not create a redundant path between the potential mesh node and a Steiner node since a Steiner node is not a source or a receiver in the multicast group and does not need to be protected by a redundant path.

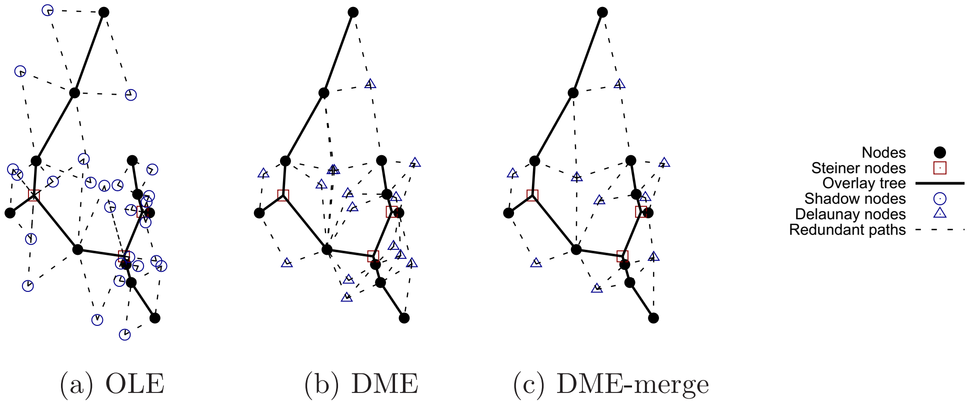
One advantage of the DME algorithm is that the redundant paths will not overlap each other. However, one drawback of the DME algorithm is that the analysis cannot be applied directly like the overlay link extension since the lengths of the redundant paths are not identical. To address this problem, a scaling factor can again prove useful. In the DME algorithm, the distance between two nodes is fixed to $r = f \cdot r_t$ where $0 < f \leq 1$. In other words, the algorithm uses a fixed, shortened distance when building a multicast mesh instead of using different routing strategies for branching nodes and merging nodes.

Even though the DME algorithm does not create overlapping redundant paths, it is still possible for two Delaunay nodes to be located close together. To solve this problem, we next propose a variation of the DME algorithm.

5.2.2. Delaunay mesh extension with node merging

The goal of the Delaunay mesh extension with node merging algorithm (DME-merge) is to merge two Delaunay nodes that are located closer than a given distance.

The algorithm takes the mesh structure from the DME algorithm and repeatedly combines two Delaunay nodes if they are separated by a distance smaller than a given `merging_distance` by placing a new Delaunay node at the midpoint of the two nodes. The new Delaunay node connects to all the overlay nodes that were connected to the two Delaunay



(a) OLE (b) DME (c) DME-merge

Fig. 7. Examples of different interference-aware mesh multicast routing structures.

nodes. The merging process continues until no two Delaunay nodes satisfy the merging condition.

For comparison, we show an example of the three proposed interference-aware mesh multicast routing structures in Fig. 7.

6. Performance evaluations

We evaluate our algorithms in four different settings. First, we start by evaluating the algorithms in a low intensity multicast setting where the analyses in Section 4.1 can be applied directly. Next, we evaluate our algorithms in a general multicast scenario where the analyses cannot be applied directly and scheduling is required. We also evaluate our algorithms in a CSMA/CA setting where nodes access the channel in a contention-based manner rather than with TDMA. Finally, we evaluate our algorithms in a CSMA/CA setting where wireless links are bursty.

6.1. Simulation parameters and assumptions

We use the ns-3.15 simulator to evaluate all of our proposed algorithms as well as several comparison algorithms. We employ the physical interference model [34], for which ns-3 has built-in support, in all simulations. All simulations were carried out with a physical model of 802.11g at the data rate of 6 Mbps, transmission power of 40 mW for all nodes, and a thermal noise value of 290K. Except when specifically evaluating the effect of node density, 2000 nodes were uniformly distributed in a deployment area of 1000 m by 1000 m. With 802.11g at 6 Mbps, transmission range is rather small and fairly high node density is required for the network to be connected. All results reported are averaged from 1000 simulation runs where, for each simulation run, 2000 nodes were randomly placed as specified above, one of those nodes was selected at random as a multicast source, and a varying number of distinct multicast destination nodes were selected at random from the remaining nodes. The simulation parameters are summarized in Table 1. The simulations reported in Sections 6.2.1 and 6.2.2 were carried out using a custom TDMA MAC layer we added to ns-3. All other simulations were performed using the standard CSMA/CA MAC for 802.11.

6.2. Interference-aware multicast routing trees

We compare the IAST and FTM multicast algorithms against three existing tree-based structures: MNT [17], SPT [18], and COMB [23]. For the IAST algorithm, we use GeoSteiner [38] to find a Euclidean Steiner Tree.

Table 1
Simulation parameters used to evaluate multicast routing structures.

Parameter	Value
Deployment area	1000 m by 1000 m
Number of nodes	2000
Mobility model	constant position
Propagation loss model	log-distance
Path-loss exponent	3
Transmission power	40 mW
Thermal noise	290K
Device	IEEE 802.11g
Default transmission mode	6 Mbps

6.2.1. Low intensity multicast

We begin our evaluation with a low intensity multicast scenario. We varied the number of multicast destination nodes from 10 to 100. To prevent multiple nodes from forwarding packets at exactly the same time, we inserted random delays between 0 μs and 1000 μs before nodes forward packets to their children. We measured the delay between the time when the source node transmits the packet and the time when all destinations have received the packets. The simulation results are reported in Fig. 8.

As seen in Fig. 8, the IAST algorithm has the lowest delivery delay among all algorithms, even though SPT has the shortest distance between the source and every destination. SPT has a larger number of nodes and this creates more contention in the network, which means that nodes have to delay their transmissions when the channel is sensed as busy. MNT and COMB both have even longer delays, because they have both longer paths and higher contention.

6.2.2. General multicast with scheduled transmissions

In this section, we consider a general multicast application where the application generates packets faster than the time it takes to deliver packets to all destinations. Thus, we use a scaling factor and scheduling in the algorithms.

We first evaluate the scaling factor since it is a significant parameter affecting IAST and FTM performances. In this simulation, the number of multicast destinations is fixed at 10. We vary the scaling factor from 0.3 to 1.0 and collect the schedule lengths returned by IAST algorithm. We did not use the scaling factor below 0.3 since the network became disconnected in some simulations. The simulation results are reported in Fig. 9.

Fig. 9 confirms that using different scaling factors affects IAST performance. If the scaling factor is too small, the extra nodes included in the multicast tree outweigh the gain of spatial reuse. If the scaling factor is too large, spatial reuse is not possible.

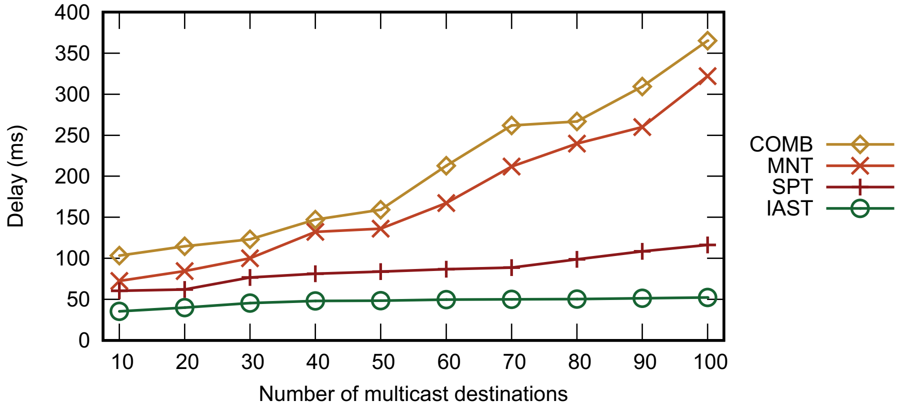


Fig. 8. End-to-end delivery delay of different multicast routing structures.

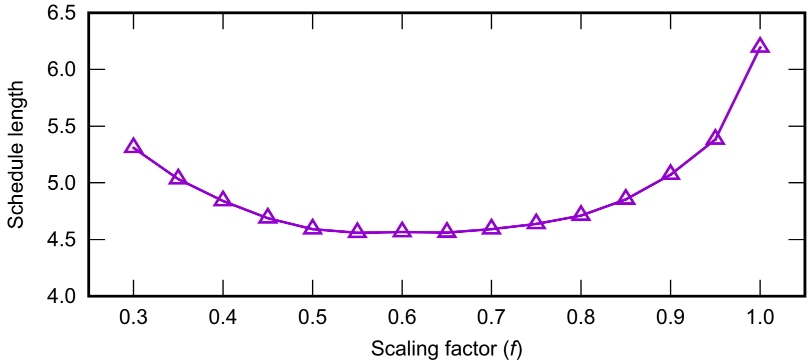


Fig. 9. IAST schedule length with different scaling factors.

Recall from Section 4.2.1 that there is a scaling factor that achieves the optimal schedule length and reducing the scaling factor below that value will not improve schedule length further. From Fig. 9, it appears that the optimal schedule length is achieved at around $f = 0.6$ for these network settings. Note, however, that the schedule length is fairly robust to the value of f and that in the interval from $f = 0.5$ to $f = 0.7$, the schedule length is very close to the optimal value. We have evaluated the scaling factor over a range of network settings and consistently found it to be stable in the range from 0.5 to 0.7. Based on these results, we have set the scaling factor to 0.7 in the remaining simulations.

We now consider the schedule length produced by different multicast routing structures, including our IAST and FTM structures. For these simulations, we varied the number of destinations from 10 to 100. There are multiple possible combinations between routing and scheduling algorithms. IAST-GP refers to IAST routing combined with the GreedyPhysical scheduling, IAST-TGP refers to IAST routing combined with Tree-Based GreedyPhysical. For MNT, SPT and COMB, we report the results from GreedyPhysical only since the difference between GreedyPhysical and Tree-Based GreedyPhysical are not statistically significant with 1000 simulations. The schedule lengths are reported in Fig. 10.

The results of Fig. 10 show that the schedule lengths of IAST and FTM are substantially shorter than SPT, MNT, and COMB. FTM schedules are approximately half the length of MNT schedules, 1/3 the length of COMB schedules, and less than 1/4 the length of SPT schedules for higher numbers of destinations. The shorter schedule lengths mean that IAST and FTM can support more transmissions than other structures within the same time period. As expected, SPT and MNT achieved their respective goals of shortest average path length and minimum number of forwarding nodes. However, multicast routing structures built by MNT and SPT mostly consist of black links that cannot be activated concurrently with any others. As a result, most of the forwarding nodes selected by MNT and SPT must be scheduled sequentially, which increases the overall schedule length. The presence of black links is even more problematic in SPT since the number of forwarding nodes is not taken into account, resulting in a very large number of forwarding nodes that must be scheduled sequentially. For COMB structure, the rigid structure of COMB results in longer schedule length.

We also evaluate the performance of all algorithms with varying network density. We kept the number of destinations at 10 and varied the number of nodes in the network from 500 to 3000. The minimum number of nodes could not drop below 500 since the network becomes disconnected in some simulations with our settings of 802.11g at 6Mbps. The schedule lengths are reported in Fig. 11.

As seen from Fig. 11, the schedule lengths of IAST and FTM algorithms increase as the node density decreases. This effect is particularly noticeable for IAST, because the closest nodes to ideal Steiner node locations can be quite far, meaning that the tree structures begin to deviate significantly from ideal Steiner trees. Since FTM is not built on Steiner trees, it is less susceptible to the lack of ideal node locations and its schedule length does not increase as dramatically for lower node densities. Even at the lowest node density, FTM's schedule length is about 17% shorter than SPT's, almost 34% lower than MNT's, and more than 55% lower than COMB's.

6.2.3. General multicast with contention-based access

Although in CSMA/CA networks, transmissions cannot be precisely scheduled, we hypothesized that the greater transmission concurrency facilitated by our interference-aware algorithms would still translate into better performance for the CSMA/CA case.

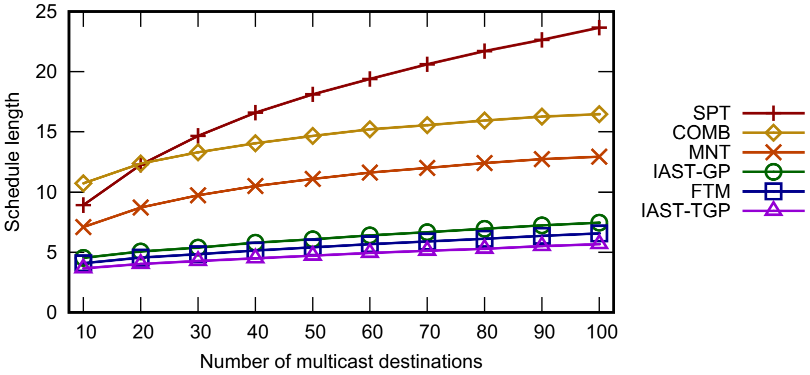


Fig. 10. Schedule length with varying number of destinations.

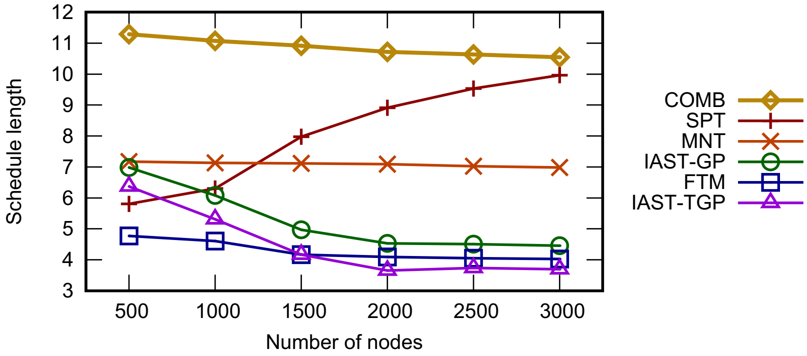


Fig. 11. Schedule length of different algorithms with varying node density.

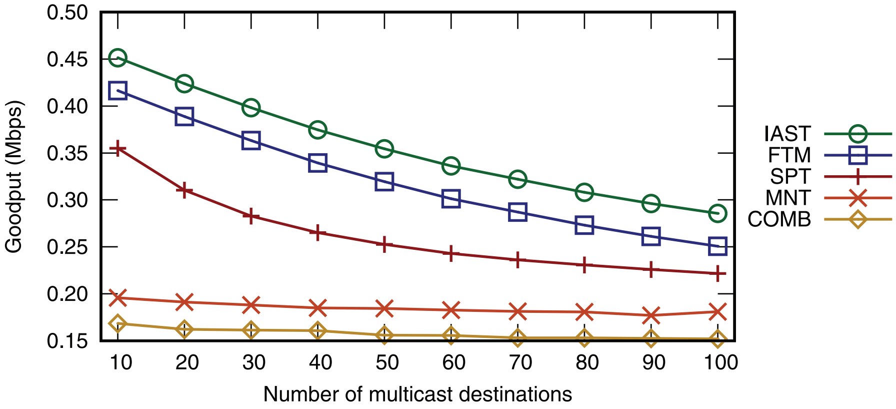


Fig. 12. CSMA goodput of different multicast routing structures.

To evaluate this, we performed ns-3 simulations using the standard ns-3 802.11 model. To prevent multiple nodes from forwarding packets at exactly the same time, we again inserted random delays before nodes forward packets to their children. The random delays were chosen uniformly between 0 μs and 1000 μs. We measured the maximum goodput and report the results in Fig. 12. Note that there is no scheduling in this CSMA/CA simulation, thus, only one variation of IAST is reported.

As seen from Fig. 12, IAST and FTM still outperform SPT and MNT even without explicit scheduling. In IAST and FTM, concurrent transmission is possible while black links in SPT and MNT are intolerant of even a small interference from concurrent transmissions. The random delays help mitigate the problem in SPT and MNT. Without random delays, SPT and MNT performances were even lower than the reported results. Thus, our interference-aware multicast routing structures permit higher levels of concurrency and this results in significant performance benefits even without explicit scheduling.

6.2.4. General multicast with contention-based access and bursty links

One of the advantages of the mesh-based multicast routing protocols over tree-based routing protocols is that multicast packets have more than one possible path to reach the destination. In tree-based multicast, multicast packets have only one possible path to reach the destination. Any failed transmission along the path will result in the destinations missing the multicast packet.

To illustrate the impact of failed transmissions on tree-based multicast, in this section, we evaluate the performance of one of our tree-based multicast protocols, namely IAST, when the wireless channel is bursty. When the wireless channel is bursty, transmission between two nodes have higher probability to fail than when the wireless channel is not bursty. As a result, wireless link

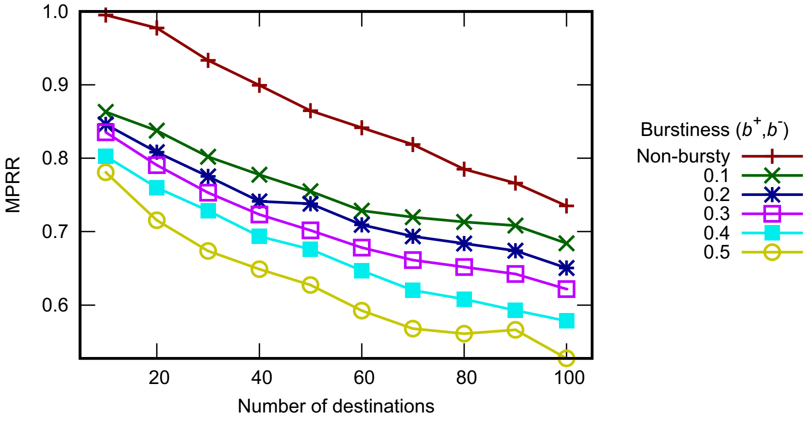
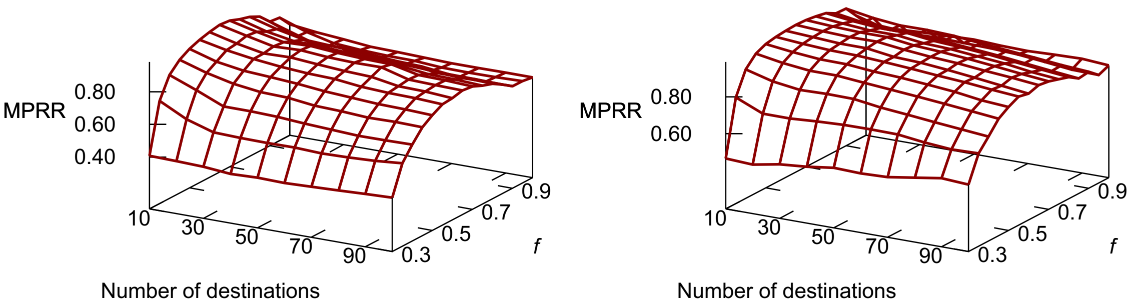


Fig. 13. Multicast packet reception ratio of the IAST protocol with varying wireless link burstiness.



(a) Overlay link extension

(b) Delaunay mesh extension

Fig. 14. Multicast packet reception ratio of OLE and DME with varying scaling factor (f).

burstiness will adversely affect the performance of tree-based multicast protocols.

To evaluate IAST's performance in this scenario, we performed ns-3 simulations using the standard ns-3 802.11 model. We used the ideal bursty link BPA function [39] with varying burstiness to simulate bursty links. We measured the multicast packet reception ratio and report the results in Fig. 13.

As seen from Fig. 13, IAST's packet reception ratio decreases as link burstiness increases. This result is expected since transmission between nodes are more prone to error when link burstiness increases. Since the IAST protocol uses a tree as the underlying routing structure, transmission failure will lower the multicast packet delivery rate at the destinations.

6.3. Interference-aware multicast routing meshes

We evaluate our algorithms in four sets of simulations. First, we start by evaluating the scaling factor to be used by the overlay link extension algorithm and Delaunay mesh extension algorithm. Second, we evaluate our proposed multicast mesh structures and other multicast routing structures in a network with bursty wireless links. We study the impact of the level of burstiness in the third set of simulations. Finally, we evaluate the performance of our algorithms in a network where nodes randomly drop multicast packets.

6.3.1. Tuning scaling factor and merging distance

We first evaluate the scaling factor since it is a significant parameter affecting both OLE and DME. In this simulation, one node is randomly selected as a multicast source. The source node sends multicast packets at the rate of 10 per second. We vary the scaling factor from 0.3 to 1.0 and collect the multicast packet reception ratio of both OLE and DME. We did not use the scaling factor below 0.3 since the network became disconnected in some simulations. The simulation results are reported in Fig. 14.

As seen from Fig. 14, the choice of scaling factor affects the performance of OLE and DME algorithms. If the scaling factor is too small, the extra nodes included create more interference that can outweigh the gain of spatial reuse. If the scaling factor is too large, the links are prone to interference from other concurrent transmissions. The optimal scaling factor is also dependent on the number of multicast destinations in the network as the number of nodes in the multicast trees changes. However, the figure shows that performance is quite stable across a fairly wide range of scaling factors, e.g. 0.5 to 0.7. Based on this analysis, we have set the scaling factor to 0.7 in the remaining simulations. While we do not have a strong intuition as to why performance is stable over this range of scaling factors in this case, we assume that there is a similar phenomenon to the scaling factor analysis we did for tree-based multicast in Section 4.2.1 that was further discussed in Section 6.2.2.

Next, we evaluate the merging distance parameter of DME-merge algorithm. In this simulation, we use the scaling factor of 0.7 and vary the merging distance of DME-merge from 10 to 70 m. Multicast packet reception ratios of DME-merge are reported in Fig. 15.

Fig. 15 confirms that the choice of merging distance affects the performance of DME-merge algorithm. When the merging distance is small, only a few Delaunay nodes are merged together, the mesh structure of DME-merge is still similar to the mesh structure of DME. However, if the merging distance is large, DME-merge will aggressively merge Delaunay nodes. This aggressive merging can result in reduced performance. When DME-merge merges two Delaunay nodes, the resulting Delaunay node is responsible for all links of the two original Delaunay nodes. Thus, the more

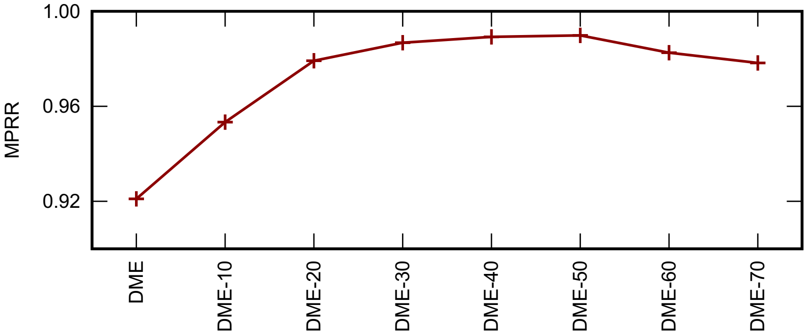


Fig. 15. Multicast packet reception ratio of DME-merge algorithm with different merging distance.

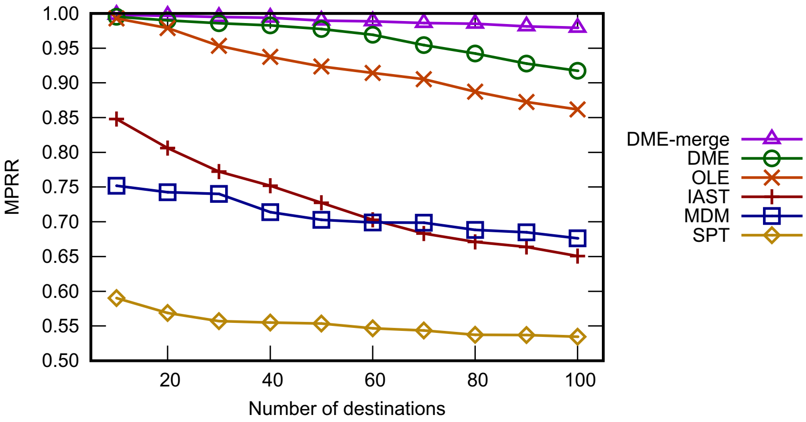


Fig. 16. Multicast packet reception ratio of different mesh multicast structures.

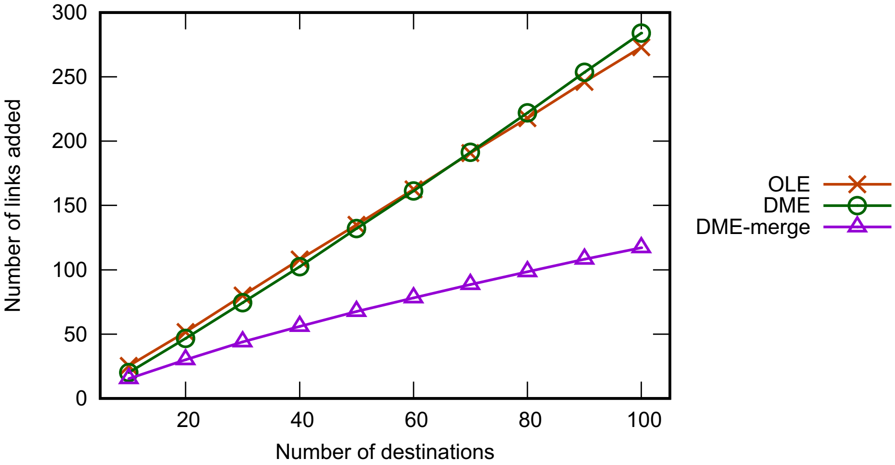


Fig. 17. Number of links added to the multicast tree from different algorithms.

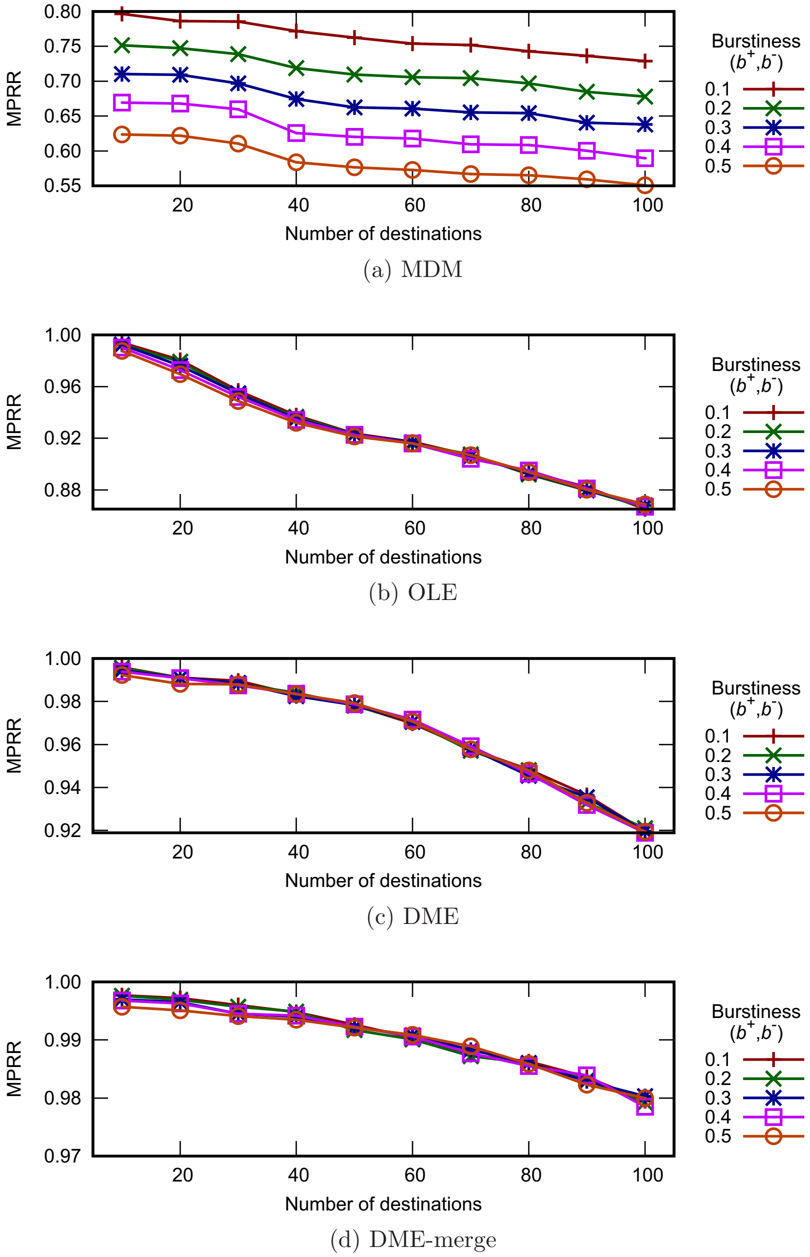
Delaunay nodes that are merged into one, the more links the new Delaunay node must handle. In an extreme case, this can create a Delaunay node that is connected to all other overlay nodes, while having only one incoming path to the Delaunay node.

Again, performance is quite stable across a fairly wide range of merging distances, e.g. 30–50. Based on these results, we have set the merging distance to 50 in the remaining simulations.

6.3.2. Mesh multicast with bursty links

In this simulation, we evaluate the performance of different multicast routing structures when the wireless links exhibit bursty behavior. Wireless communications in this simulation are modeled with ideal bursty links [39]. The number of multicast destinations was varied from 10 to 100. A single multicast source was randomly selected among the remaining nodes. The source node sends multicast packets at the rate of 10 per second for 600 s. We have implemented another mesh multicast routing structure called MDM for comparison [19]. We measured the multicast packet reception ratio (MPRR) and report the results in Fig. 16.

As seen in Fig. 16, the performance of the shortest path tree is the lowest among the multicast routing structures. Since the shortest path tree does not consider interference when building a tree, it is more prone to interference and bursty links than other structures. The tree structure also makes it vulnerable to even a single transmission failure as the tree will be disconnected. The figure also shows that our interference-aware Steiner tree provides



(a) MDM

(b) OLE

(c) DME

(d) DME-merge

Fig. 18. Multicast packet reception ratio of multicast routing structures with different burstiness.

improvement over the shortest path tree since IAST takes interference into account when building a tree. As a result, IAST is less prone to interference and bursty links than the shortest path tree. Still, a single transmission failure will disconnect the multicast tree of IAST. MDM also provides improvement over the shortest path tree by including multiple paths to deliver multicast packets to the destinations. This robustness increases the performance of the multicast structure since a single transmission failure will not result in receivers missing the multicast packets. However, since MDM does not take interference into account when building a mesh, it is still prone to transmission failure along the mesh.

OLE and DME algorithms have higher multicast packet reception ratios than other multicast routing structures, including our previously proposed interference-aware Steiner tree algorithm. The additional paths included by extending the Steiner overlay tree allow the multicast packets to be delivered to the destinations even if some multicast packets were dropped on other paths to the receivers. Moreover, the paths built by both OLE and DME algorithms are interference-aware, which means that the nodes on the paths are less prone to transmission failure than other multicast mesh routing structures.

Among our three mesh algorithms, Delaunay mesh extension with node merging provides the highest multicast packet reception ratio, especially with a large number of multicast destinations. When the number of multicast destinations is large, the number of links that are introduced by the overlay link extension algorithm and Delaunay mesh extension algorithm increase quickly. The numbers of links added to the multicast tree after different tree extension algorithms were applied to the multicast tree are reported in Fig. 17.

As seen from Fig. 17, the number of links added to the multicast tree increases as the number of destinations increases. The added links can create more contention in the network and introduce more interference among the paths. This effect results in

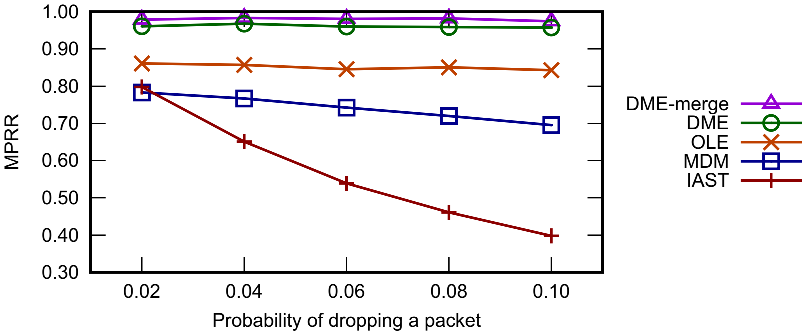


Fig. 19. Multicast packet reception ratio of different algorithms at varying faulty node probability.

lower multicast packet reception ratios of the overlay link extension algorithm and Delaunay mesh extension algorithm. For Delaunay extension algorithm with node merging, the algorithm tries to merge Delaunay nodes that are located close together, which reduces the number of new links in the final multicast structure. As a result, Delaunay extension algorithm with node merging does not suffer from this problem as much as the other two tree extension algorithms.

6.3.3. *Impact of link burstiness level*

In this set of simulations, we study the impact of the burstiness of the wireless links on the performance of multicast routing structures. Again, we varied the number of multicast destinations from 10 to 100 with one node selected randomly as a multicast source. We varied the wireless link burstiness by varying the parameter b^+ and b^- of the ideal bursty link model from 0.1 to 0.5. Multicast packet reception ratios of different multicast routing structures are reported in Fig. 18.

It is quite apparent from Fig. 18 that MDM performs much worse in the presence of bursty links than our proposed mesh structures. Even with a low level of burstiness, MDM's multicast packet reception ratio (MPRR) is only 0.7–0.8, depending on the number of destinations. For higher levels of link burstiness, MDM's MPRR drops down to below 0.6. All of our algorithms retain a MPRR of above 0.85 for all numbers of destinations and they are almost unaffected by the level of burstiness. DME-merge performs particularly well, having a MPRR of 0.98 or higher in almost all cases. The reason for MDM's poor performance with bursty links is that its mesh paths are quite long, making them very susceptible to link failures. In our mesh structures, redundant paths are quite short, connecting a pair of multicast destinations before merging back into the main path. This makes them much less susceptible to link failures.

6.3.4. *Mesh multicast with faulty nodes*

In this set of simulations, we look at another cause for transmission failure – the presence of faulty nodes in the network. For simplicity, we assume that wireless links are not bursty in this set of simulations. To simulate a faulty node, each node that is participating in the multicast routing structure randomly decides to drop a multicast packet instead of forwarding the packet to the next nodes. The probability of dropping a multicast packet was varied from 0.02 to 0.10. The decision to drop the packet is made independently for each multicast packet. In this set of simulations, the number of multicast destinations was kept constant at 50. Multicast packet reception ratios of different multicast routing structures are reported in Fig. 19.

As seen from Fig. 19, the performance of most multicast routing structures drops as the probability of faulty node increases. The performance drop is substantial for IAST since it relies on a tree as a multicast routing structure. A single faulty node along the tree will disconnect the subtree below the faulty node.

For MDM, the multicast packet reception ratio drops slightly from about 0.80 to about 0.70. MDM relies on mesh structure, which makes it more robust when a few nodes are faulty. However, the number of redundant paths of MDM is not large enough to handle a large number of faulty nodes, which results in slight drop in multicast packet reception ratio when the probability of a faulty node is high.

Our proposed mesh multicast routing structures can withstand a larger number of faulty nodes than MDM and IAST as can be seen by the almost constant multicast packet reception ratios even at a high faulty node probability. The extra paths included by OLE, DME, and DME-merge make them more robust to faulty nodes than other multicast routing structures.

7. Conclusion

In this paper, we have studied the problem of interference-aware multicasting in wireless multihop networks. We first started by investigating multicast trees, which are the minimal structures to deliver multicast messages to all destinations. We classified nodes in multicast trees into different classes and showed, by analyzing an accurate physical interference model, that different classes require different routing strategies. Based on the analyses, we proposed two joint routing and scheduling algorithms for multicasting in wireless multihop networks. We evaluated the performance of different algorithms through simulation and showed that the proposed algorithms outperform other algorithms both in terms of schedule lengths and goodput.

Based on the results of the multicast tree study, we considered the problem of building an interference-aware mesh multicast routing structure. We proposed two algorithms to extend an interference-aware multicast tree to form an interference-aware multicast mesh based on interference analyses. We showed that the proposed interference-aware mesh structures provide higher reliability than our multicast tree structures and other mesh-based multicast structures, even with the presence of bursty wireless links or faulty nodes in the network.

Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under Grants CNS–1017248 and CNS–1319455. The NSF did not participate in the research in any way and the opinions expressed herein are solely those of the authors and not the NSF.

References

[1] D. Lertpratchya, D.M. Blough, G.F. Riley, Interference-aware multicast for wireless multihop networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference, 2014a, pp. 3034–3039.

[2] D. Lertpratchya, D.M. Blough, G.F. Riley, Interference-aware mesh multicast for wireless multihop networks, in: Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2014b, pp. 163–167.

[3] L. Junhai, Y. Danxia, X. Liu, F. Mingyu, A survey of multicast routing protocols for mobile ad-hoc networks, IEEE Commun. Surv. Tutor. 11 (1) (2009) 78–91.

[4] E.M. Royer, C.E. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '99, ACM, New York, NY, USA, 1999, pp. 207–218.

[5] L. Ji, M. Corson, Differential destination multicast-a MANET multicast routing protocol for small groups, in: Proceedings of Twentieth Annual Joint Conference on IEEE Computer and Communications Societies, in: INFOCOM 2001, 2, 2001, pp. 1192–1201.

[6] A. Penttinen, Efficient multicast tree algorithm for ad hoc networks, in: Proceedings of 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2004, pp. 519–521.

[7] J. Yi, C. Poellabauer, Real-time multicast for wireless multihop networks, Comput. Electr. Eng. 36 (2) (2010) 313–327.

[8] S.-J. Lee, W. Su, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, Mobile Netw. Appl. 7 (6) (2002) 441–453.

[9] R. Vaishampayan, J. Garcia-Luna-Aceves, Efficient and robust multicast routing in mobile ad hoc networks, in: Proceedings of 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2004, 2004, pp. 304–313.

[10] I. Er, W. Seah, Distributed Steiner-like multicast path setup for mesh-based multicast routing in ad hoc networks, in: Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006., vol. 2, 2006, pp. 192–197.

[11] D. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, H. Lee, Achieving minimum-cost multicast: a decentralized approach based on network coding, in: Proceedings of the 24th Annual Joint Conf. IEEE Computer and Communications Societies, in: INFOCOM 2005, 3, 2005, pp. 1607–1617.

[12] J.-S. Park, M. Gerla, D. Lun, Y. Yi, M. Medard, Codecast: a network-coding-based ad hoc multicast protocol, IEEE Wireless Commun. 13 (5) (2006) 76–81.

[13] J. Sanchez, P. Ruiz, I. Stojmenovic, GMR: Geographic multicast routing for wireless sensor networks, in: Proceedings of 3rd Annual IEEE Communications Society Conference Sensor and Ad Hoc Communications and Networks, vol. 1, 2006, pp. 20–29.

[14] D. Koutsonikolas, S. Das, Y. Hu, I. Stojmenovic, Hierarchical geographic multicast routing for wireless sensor networks, Wireless Netw. 16 (2) (2010) 449–466.

[15] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proceedings of IEEE Wireless Communications and Networking Conference, in: WCNC 2003, vol. 2, 2003, pp. 1118–1123.

[16] N. Meghanathan, Determining a sequence of stable multicast Steiner trees in mobile ad hoc networks, in: Proceedings of the 44th Annual Southeast Regional Conference, in: ACM-SE 44, ACM, New York, NY, USA, 2006, pp. 102–106.

[17] P. Ruiz, A. Gomez-Skarmeta, Approximating optimal multicast trees in wireless multihop networks, in: Proceedings of the 10th IEEE Symposium Computers and Communications, 2005, pp. 686–691.

[18] U.T. Nguyen, On multicast routing in wireless mesh networks, Comput. Commun. 31 (7) (2008) 1385–1399.

[19] X. Zhao, C.T. Chou, J. Guo, S. Jha, Protecting multicast sessions in wireless mesh networks, in: Proceedings of 31st Annual IEEE International Conference on Local Computer Networks, LCN 2006, 2006, pp. 467–474.

[20] Y. Zheng, U.T. Nguyen, H.L. Nguyen, Data overhead impact of multipath routing for multicast in wireless mesh networks, in: Proceedings of 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing (MUSIC), 2012, pp. 154–157.

[21] X.-Y. Li, S.-J. Tang, O. Frieder, Multicast capacity for large scale wireless ad hoc networks, in: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07, ACM, New York, NY, USA, 2007, pp. 266–277.

[22] G. Rodolakis, A. Laouiti, P. Jacquet, A.M. Naimi, Multicast overlay spanning trees in ad hoc networks: Capacity bounds, protocol design and performance evaluation, Comput. Commun. 31 (7) (2008) 1400–1412. Special Issue: Resource Management and routing in Wireless Mesh Networks.

[23] S. Shakkottai, X. Liu, R. Srikant, The multicast capacity of large multihop wireless networks, IEEE/ACM Trans. Netw. 18 (6) (2010) 1691–1700.

[24] J.E. Wieselthier, G.D. Nguyen, A. Ephremides, Energy-efficient broadcast and multicast trees in wireless networks, Mobile Netw. Appl. 7 (6) (2002) 481–492.

[25] Y. Wu, P.A. Chou, S.-Y. Kung, Minimum-energy multicast in mobile ad hoc networks using network coding, IEEE Trans. Commun. 53 (11) (2005) 1906–1918.

[26] M.X. Cheng, J. Sun, M. Min, Y. Li, W. Wu, Energy-efficient broadcast and multicast routing in multihop ad hoc wireless networks, Wireless Commun. Mobile Comput. 6 (2) (2006) 213–223.

[27] K. Chen, K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in MANET, in: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, in: INFOCOM 2002, vol. 3, 2002, pp. 1180–1189.

[28] J. Cha, J. Jeon, J. Kim, Y. Kwon, Location-based multicast routing algorithms for wireless sensor networks in presence of interferences, in: Proceedings of the IEEE International Conference on Communication Systems, 2010, pp. 41–45.

[29] J. Wang, M. Song, Y. Zhao, Interference-aware multicast in wireless mesh networks with directional antennas, in: Proceedings of the IEEE International Conference on Networking, Architecture, and Storage, 2010, pp. 171–180.

[30] K. Wang, C.-F. Chiasserini, R.R. Rao, J.G. Proakis, A joint solution to scheduling and power control for multicasting in wireless ad hoc networks, EURASIP J. Appl. Signal Process. 2005 (2005) 144–152.

[31] T. Erlebach, T. Grant, Scheduling multicast transmissions under SINR constraints, in: C. Scheideler (Ed.), Algorithms for Sensor Systems, Lecture Notes in Computer Science, 6451, Springer Berlin Heidelberg, 2010, pp. 47–61.

[32] P. Basu, C.-K. Chau, A. Bejan, R. Gibbens, S. Guha, Efficient multicast in hybrid wireless networks, in: Proceedings of the Military Communications Conference, 2015, pp. 306–311.

[33] B. Das, V. Bhargavan, Routing in ad-hoc networks using minimum connected dominating sets, in: Proceedings of the International Conference on Communications, 1997, pp. 376–380.

[34] P. Gupta, P. Kumar, The capacity of wireless networks, IEEE Trans. Information Theory 46 (2) (2000) 388–404.

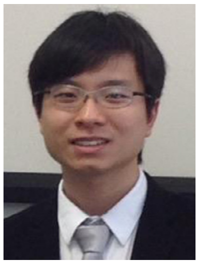
[35] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks, in: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '06, ACM, New York, NY, USA, 2006, pp. 190–201.

[36] D. Blough, G. Resta, P. Santi, Approximation algorithms for wireless link scheduling with SINR-based interference, IEEE/ACM Trans. Netw. 18 (6) (2010) 1701–1712.

[37] G. Brar, D.M. Blough, P. Santi, Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks, in: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, MobiCom '06, ACM, New York, NY, USA, 2006, pp. 2–13.

[38] D. Warme, P. Winter, M. Zachariasen, Exact algorithms for plane Steiner tree problems: A computational study, Adv, Steiner Trees 6 (2000) 81–116.

[39] D. Lertpratchya, G.F. Riley, D.M. Blough, Simulating frame-level bursty links in wireless networks, in: Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques, in: SIMUTools '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, Belgium, 2014, pp. 108–117.



Daniel Lertpratchya received the B.Eng. degree in computer engineering from Chulalongkorn University, Thailand in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology in 2009 and 2014, respectively. His research interests include computer networking, wireless networking, sensor networks, and computer network simulation. This work was completed while he was a Ph.D. candidate at Georgia Tech. He is currently with Apple Computer.

Douglas M. Blough received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in computer science from The Johns Hopkins University, Baltimore, MD, in 1984, 1986, and 1988, respectively. Since Fall 1999, he has been Professor of Electrical and Computer Engineering at the Georgia Institute of Technology, where he also holds a joint appointment in the School of Computer Science. From 1988 to 1999, he was on the faculty of Electrical and Computer Engineering at the University of California, Irvine. Dr. Blough's research interests include wireless multihop networks, and dependability and security of systems and networks. He has published over 120 articles on these topics in refereed journals and conferences.

Dr. Blough was General Chair for the International Conference on Dependable Systems and Networks (DSN) in 2014 and has been Program Chair for the International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2009), the International Conference on Dependable Systems and Networks (DSN 2000), and the Pacific Rim International Symposium on Dependable Computing (PRDC 1997). He was also Area Program Chair for ICDCS 2008 (Reliability and Dependability) and for MASS 2007 (Algorithms and Theory). He has served on Program Committees for these and many other conferences, including INFOCOM, Mobicom, and Mobihoc, among others. He is currently Associate Editor for IEEE Transactions on Dependable and Secure Computing and was a past Associate Editor for IEEE Transactions on Computers (1995–2000), IEEE Transactions on Parallel and Distributed Systems (2001–2005), and IEEE Transactions on Mobile Computing (2008–2013).