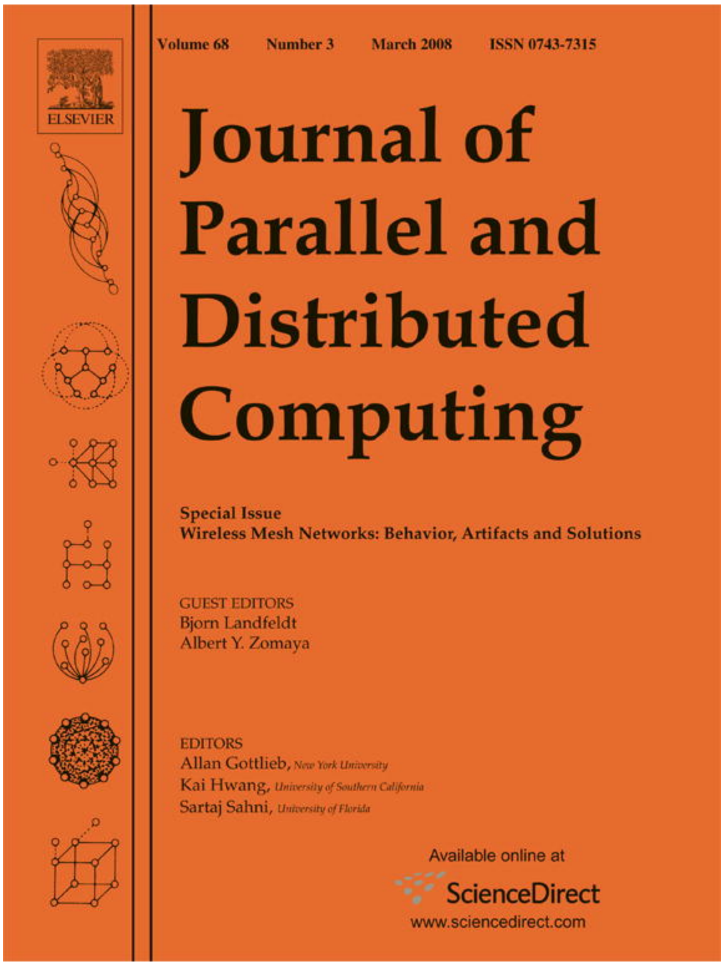


Volume 68 Number 3 March 2008 ISSN 0743-7315

Journal of Parallel and Distributed Computing

Special Issue
Wireless Mesh Networks: Behavior, Artifacts and Solutions

GUEST EDITORS
Bjorn Landfeldt
Albert Y. Zomaya

EDITORS
Allan Gottlieb, *New York University*
Kai Hwang, *University of Southern California*
Sartaj Sahni, *University of Florida*

Available online at www.sciencedirect.com
ScienceDirect
J. Parallel Distrib. Comput. 68 (2008) 372 – 386
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

An interference-aware fair scheduling for multicast in wireless mesh networks

Dimitrios Koutsonikolas, Saumitra M. Das*, Y. Charlie Hu

School of Electrical and Computer Engineering, Purdue University, MSEE 212, 1285 Northwestern Ave, West Lafayette, IN 47907, USA

Received 16 November 2006; accepted 7 May 2007
Available online 26 May 2007

Abstract

Multicast is a fundamental routing service in wireless mesh networks (WMNs) due to its many potential applications such as video conferencing, online games, and webcast. Recently, researchers proposed using link-quality-based routing metrics for finding high-throughput paths for multicast routing. However, the performance of such link-quality-based multicast routing is still limited by severe unfairness. Two major artifacts that exist in WMNs are fading which leads to low quality links, and interference which leads to unfair channel allocation in the 802.11 MAC protocol. These artifacts cause the multicast application to behave unfairly with respect to the performance achieved by the multicast receivers.

In this paper, we design a MAC layer solution to improve the fairness of multicast service in WMNs. Our proposed MAC layer takes into account the interference among multicast forwarding nodes and assigns them transmission time slots while maximizing the spatial reuse for high throughput. Detailed simulations and testbed experiments show that our solution significantly increases fairness as well as throughput compared to the 802.11 protocol.

Keywords: Multicast; Wireless mesh networks; Interference

1. Introduction

Wireless mesh networks (WMNs) have been proposed as an efficient solution for ubiquitous last-mile broadband access. The deployment and use of WMNs has increased significantly and several cities have planned and/or deployed WMNs [1,5,24,34,35]. To compete with existing broadband technologies such as cable and DSL and become a viable solution, WMNs have to meet two performance criteria: to provide *high throughput* in order to meet the ever-growing demand of network applications (e.g., online gaming, video conferencing, webcast), and to guarantee *fairness* among different clients who usually pay the same flat rate for their subscriptions.

In a typical WMN environment, two major factors can severely affect both throughput and fairness of the whole network: *fading* and *interference*. Fading is the random attenuation of the signal due to reflections, scatterings and multipath propagation. Fading leads to inherently low quality links (even if the two end nodes are within transmission range) which can incur random packet losses. Interference leads to both unfair channel allocation in the 802.11 MAC protocol, when nodes defer their transmissions after sensing other transmissions, and packet drops, when the noise due to signals and multipaths from other transmitters decreases the SINR below the packet reception threshold. Hence, any practical solution that provides high throughput or fairness in a WMN has to deal with these two factors.

Because of its random nature, fading is very difficult to model or even to measure. In contrast, interference by itself is relatively easier to model and many heuristics have been proposed [8,11,17,18,36] in isolation of fading. However, in a realistic environment, interference is exacerbated by fading and its behavior also becomes random. In this case, simple heuristics either underestimate or overestimate the effects of interference [28]. As a result, it is difficult to develop theoretical solutions that ignore fading to work well in a real environment.

To overcome the limitation of theoretical solutions, a plethora of experimental approaches has been proposed towards

---

* Corresponding author. Fax: +1 765 494 3371.
E-mail address: smdas@purdue.edu (S.M. Das).

0743-7315/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2007.05.007

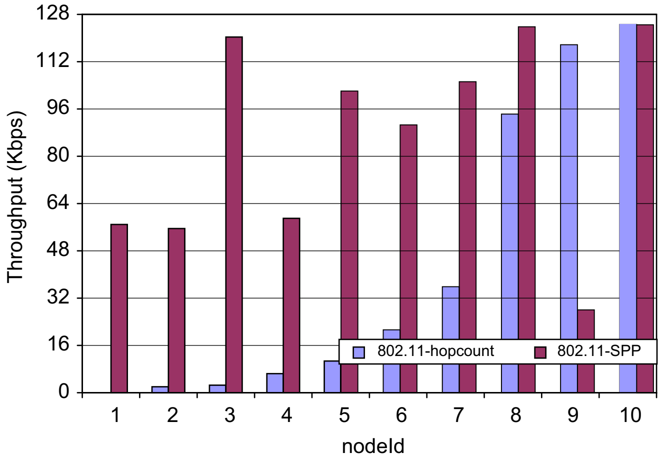


Fig. 1. Throughput comparison of two multicast trees with 10 multicast members based on hop-count and SPP metrics in a 50-node WMN. The source sending rate is 128 Kbps.

providing high throughput and fairness in realistic environments, in particular for unicast protocols. For example, link-quality-based metrics for high-throughput unicast routing were proposed in [8,11,12], opportunistic routing to exploit long, lossy links was proposed in [3] and an overlay MAC layer solution for fairness was proposed in [30].

Multicast is another fundamental routing service in multihop mesh networks. It provides an efficient means of supporting collaborative applications such as video conferencing, online games, webcast and distance learning, among a group of users [25]. In spite of its significance, there has been little work on multicast routing in wireless networks. Moreover, there is no experimental work on providing fairness among multicast members in a multihop wireless network.

Most recently, the authors of [32] studied link-quality-based routing metrics for finding high-throughput paths for multicast routing. Although these metrics have been shown to improve throughput compared to the widely used hop-count metric, one significant problem—unfairness among different members of a multicast session[1]—remains unsolved. As an illustration, Fig. 1 plots the throughput achieved by the 10 members of a multicast session using trees constructed using hop-count and success probability product (SPP), respectively. SPP selects the path with the highest probability of packet delivery ratio (PDR) and it was shown in [32] to offer the highest throughput among various link-quality metrics for multicast routing. With the hop-count metric, five members totally starve, achieving throughput between 0 and 10 Kbps. Although SPP offers significant throughput improvement for 9 out of 10 members, significant unfairness remains: some receivers have throughput as high as 122 Kbps (close to optimal) and others have throughput as low as 30 Kbps.

SPP considers the use of longer paths consisting of shorter links, since long links are known to suffer from heavy packet losses. However, adding more nodes in a path increases the number of transmissions, and consequently the probability of collisions, if these transmissions are not scheduled properly. This does not only limit the gain that could be achieved by link-quality-based metrics, but it can also reduce throughput compared to hop-count metric in some cases. For example, in Fig. 1, throughput is significantly reduced for receiver 9. In other words, the unfairness problem cannot be solved in the context of the currently used MAC protocols (802.11). Hence, there is the need for designing new efficient MAC protocols that will schedule contending transmissions to avoid collisions.

In this paper, we propose a MAC layer solution to improve the fairness of multicast service in WMNs while exploiting spatial reuse for high throughput. Since fading is very difficult to model or to measure, our solution focuses on addressing the unfairness caused by interference. Our TDMA-like MAC layer, called interference-aware fair scheduling (IAFS) for multicast, consists of a measurement-based interference model and an interference-aware scheduler that assigns transmission time slots to forwarding nodes in the multicast tree(s) while maximizing the spatial reuse for high throughput. Note that, although packet scheduling in wireless networks has been extensively studied (mainly for unicast [15,20,21,33] but also for broadcast/multicast [2,6,7,14]) all these works focus on theoretical solutions, making assumptions such as the "unit disk graph" model, or "2-hop interference" model, which cannot model reality. Hence, their results can only serve as upper bounds for practical cases. Also, for the case of multicast none of the existing works focuses on fairness. Since the problem of packet scheduling is known to be NP-hard, our goal in this paper is not to propose another approximation algorithm, but rather to design a scheme to improve multicast fairness and throughput that will work well in a realistic environment. To our best knowledge, we are the first to incorporate measurement-based interference awareness in a scheduling algorithm and study fairness in multicast in a realistic environment.

We evaluate the performance of our solution via extensive simulations that consider fading and noise as well as via testbed experiments. Our simulation results show that our solution significantly increases fairness compared to the 802.11 protocol: (1) without fading, our solution achieves perfect fairness among multicast receivers; (2) with fading, our solution achieves close to perfect fairness among multicast receivers; and (3) our solution gives much better tradeoff between throughput and fairness, for example, when allowing weakly interfering nodes to transmit simultaneously by using a threshold-based interference model. Finally, our experiments on a 32-node WMN testbed confirm that our scheduling algorithm significantly improves both fairness and throughput of multicast receivers.

In summary, in this paper we make the following contributions:

- We introduce the problem of unfairness among multicast receivers in WMNs, which is very important from the

---

[1] In this paper, we are interested in fairness among the mesh routers whose clients are members of the multicast session, as we assume such routers will disseminate multicast packets to their clients via other fair means, such as MAC layer broadcast.

economical point of view for the wide deployment and adoption of WMNs.
- We point out the ineffectiveness of existing scheduling algorithms in realistic WMN environments and we identify the two main reasons that cause unfairness and throughput degradation in such environments, namely interference and fading.
- We propose a practical TDMA-based MAC layer solution to improve fairness of multicast service in WMNs while exploiting spatial reuse for high throughput.
- We make our solution *interference-aware* by using a *measurement-based* interference model, which is far more accurate compared to existing heuristics.
- We present detailed simulation results with realistic settings (noise and fading models) and show that our proposed solution significantly improves fairness among multicast receivers, while offering large gains in throughput and PDR compared to the widely used 802.11 protocol.
- We offer an application-layer implementation of our solution in a wireless testbed, which in spite of its limitations, verifies the simulation results.

The rest of the paper is organized as follows. Section 2 motivates our solution by elaborating on the two main causes for unfairness in WMNs. Section 3 formulates the problem and states our assumptions. Section 4 presents our proposed solution. Section 5 presents our detailed simulation results and Section 6 presents the experimental results on a 32-node WMN testbed. Finally, Section 7 discusses the related work and Section 8 concludes the paper.

2. Causes of unfairness in 802.11 WMNs

There are two major factors that contribute to low throughput and unfairness in 802.11 WMNs: fading and interference.

Fading is the random attenuation of the signal due to reflections, scatterings and multipath propagation. These phenomena are very common in WMNs, e.g., in urban environments with buildings, trees, traffic lights. Fading causes seemingly equivalent links (e.g., links of same length) to behave differently over time. Moreover, it does not affect links with different lengths in the same way; long links are usually affected much more than short links. For these reasons, a scheduling algorithm that achieves provable fairness in theory by ignoring fading is unlikely to achieve fairness in practice.

Interference between nodes can happen when they are in the transmission range of each other as well as when they are not. The later happens when (1) one potential transmitter senses the carrier of another transmitter, or (2) the noise due to signals from other transmitters decreases the SINR below the packet reception threshold. In the first case, the 802.11 MAC layer protocol will cause the node that sensed an ongoing transmission to backoff for a random amount of time and then to retry transmitting its packets. This can cause some nodes to never get a chance to transmit their packets. In the second case, the node simply cannot receive the packet properly. Hence, both cases can lead to severe unfairness and low throughput.

In a realistic environment, interference is exacerbated by fading and its behavior also becomes random, e.g., both conditions (1) and (2) above can be affected by multipaths. Hence, similar to fading, interference becomes difficult to model. Due to this difficulty, protocols proposed for WMNs have relied on simplistic models of interference and driven their algorithms with heuristic rules such as “everyone interferes with everyone else” and “nodes in twice the transmission range interfere”. Although easy to calculate, such heuristics can be far from accurate in modeling the actual interference, as was shown in [4]. The same work further showed that, although costly, it is feasible to characterize interference through measurements.

3. Problem formulation

In this section, we formulate the fairness scheduling problem for multicast in WMNs.

Application scenario: We consider a typical multichannel, multi-interface WMN with omnidirectional antennas, in which mesh routers are placed on the rooftops of clients or other infrastructure (e.g., streetlights), and are interconnected via wireless links. One or a few gateways are connected to the Internet and propagate traffic to and from clients. Clients can be mobile and each one is associated with a mesh router. We assume mesh routers communicate with their associated clients using different radios/channels from the ones they use to communicate with other mesh routers. The whole network is under the control of a single operator. We assume that the operator maintains traffic information about all clients and their associated mesh routers, the loss rate on each inter-mesh-router link of the WMN, as well as interference between all pairs of mesh routers or links. This information can be obtained by offline measurements, following a methodology similar to [28]. We discuss this methodology in detail in Section 4.2.

For this paper, we focus to the following application scenario. When there is an important event to be multicast (e.g., a football match), the operator advertises this event, and interested WMN clients subscribe for that event. The subscribed clients along with the gateway form a multicast group.

The operator organizes a multicast session for each new event in two steps: (i) it first creates a high-throughput multicast tree from the gateway to all subscribed clients, and (ii) it then schedules the transmissions of the forwarders of all multicast trees (all concurrent multicast sessions) for fairness and high throughput. Then the operator reserves one channel for this multicast session, which is different from the channel used for unicast sessions.

Subscription in general can take place before the event, or during the event. In this paper we assume that *all* clients subscribe *before* the event. Hence, the WMN operator can compute the schedule only *once*, at the beginning, and propagate it to the WMN routers before the event starts. This means that there is no control traffic during the event and hence no control overhead. The case of dynamic schedule updates, when some clients subscribe in the middle of an event is more complicated and is part of our future work. In such a case, a unicast channel could be used for control messages—notifying the operator of new client arrivals and propagation of the new schedule. The schedule also changes if routes change. In this paper we assume that the routing tree is also constructed only *once* in the beginning using offline measurements of link-quality-based metrics stored in a database. Measurements are repeated periodically, when there is no traffic on the network. We believe such an assumption is a valid one for WMNs. In general, with a planned WMN deployment routes are much more stable than in a mobile ad hoc network and there should not be any route breaks due to mobility or power drain.

Notations: We represent the mesh network as a graph $\mathcal{G}$ where the gateway G and the mesh routers are the vertices, i.e., the set of vertices are $\mathcal{V} = \{G, MR_i,\ i = 1, \ldots, n\}$, where n is the number of mesh routers in the mesh network. For each multicast session, the operator forms a tree $\mathcal{T}$ (we discuss the tree construction in Section 4.1) to connect the gateway (multicast source) to each mesh router that has any subscribed clients associated with it. We call these routers *receivers* in the rest of the paper. Mesh routers that forward data packets form the *forwarding group* (FG), a notion borrowed from ODMRP [19]. We denote the source of the tree by S ($S = G$), the set of receivers by $\mathcal{R}$, and the set of forwarding nodes by $\mathcal{FG}$. Also we denote by $\|\mathcal{R}\|$ and $\|\mathcal{FG}\|$ the number of nodes in $\mathcal{R}$ and $\mathcal{FG}$, respectively. Note that $S \in \mathcal{FG}$ and for each receiver R, $R \in \mathcal{FG}$ if R is not a leaf node. We assume that all links have the same capacity (in absence of fading).

Due to the broadcast nature of the wireless channel, a single transmission of a transmitting node can be received by all nodes that fall in the transmission range of that node. This property of the wireless medium is known as wireless multicast advantage (WMA) [4]. In most multicast protocols for wireless networks, members of the FG use MAC layer broadcast to exploit the WMA. Our architecture adopts this convention. We assume that the receiver routers also use MAC layer broadcast, and hence per-receiver-router fairness translates into per-client fairness, assuming the links between clients and their access routers are of similar quality.

Scheduling objectives: Our scheduling algorithm assigns transmission rights to the members of $\mathcal{FG}$. We define the *cycle* of a schedule to be the time needed to activate all members of $\mathcal{FG}$ once. The cycle is repeated for the whole multicast event. We call the time interval in which a node is activated within a cycle a *time slot*, denoted by t_s. We denote by T the total number of time slots in the cycle, and hence $cycle = T \times t_s$.

Our scheduling algorithm has two objectives. The first is *to ensure fairness among the multicast receiver routers*, i.e., every receiver achieves equal throughput, assuming all links are of equal quality. To achieve this, each member of $\mathcal{FG}$ needs to be activated for equal time (i.e., one slot) within a cycle. Since packets are broadcast at the MAC layer, each node in $\mathcal{FG}$ gets C data bytes from its parent and broadcasts the same C data bytes to its children. Hence, the throughput achieved by each tree node (FG nodes and receivers) is

$$\rho = \frac{C}{T \times t_s}. \qquad (1)$$

The second objective of our algorithm is *to maximize throughput by exploiting spatial reuse*. The idea is that when multiple forwarders do not interfere with each other, they can be activated in the same time slot. This in turn reduces T which improves ρ in Eq. (1).

To summarize, the task of the scheduling algorithm is to find a schedule s for the members of $\mathcal{FG}$ with cycle length T, such that: (1) each member is assigned exactly one slot, (2) no two members that interfere are assigned the same slot, and (3) T is minimized.

Note that in a realistic environment where not all links are of the same quality due to fading, the first objective of the scheduling algorithm is not feasible. In this case, our objective changes from achieving equal throughput for every receiver to reducing the gap between the maximum and minimum throughput achieved by any two receivers. As discussed in Section 2, due to its random nature, fading is very difficult to predict or model, and subsequently to be incorporated in a scheduling algorithm. Hence, our scheduling algorithm mitigates unfairness only due to interference. However, although our scheduling algorithm does not attempt to mitigate the unfairness *directly* caused by fading, i.e., by compensating lossy links due to fading, it addresses unfairness *indirectly* caused by fading, i.e., by using a measurement-based interference model which takes into account the effects of fading.

4. Architecture and scheduling algorithm

In this section, we present IAFS for multicast, a TDMA-like MAC solution to improve fairness and throughput of multicast in WMNs. We describe the four components of IAFS: the tree construction, the interference model, the scheduling algorithm, and the propagation of the schedule to the WMN nodes.

4.1. Tree construction

Constructing a good multicast tree is critical to the performance of any scheduling algorithm. In a realistic environment, if a tree consists mainly of lossy links due to fading, its performance will be poor, even with a perfect scheduling algorithm. In [32], the authors showed that SPP, which selects the path with the highest probability of packet delivery, gives the highest throughput among various link-quality-based metrics. Hence, in IAFS, we use SPP as the tree construction metric. SPP for a link is defined as the probability for a packet to be successfully transmitted from the sender to the receiver of that link, which can be easily calculated through offline measurements. The SPP for a whole path is equal to the product of the SPP values of the links constituting the path. A modified Dijkstra algorithm is used to find the paths with highest SPP value from the source to each receiver.

4.2. Interference model

In [28], Padhye et al. showed that it is difficult to accurately model interference among links and that simple heuristics (e.g., assuming that interference range is twice the

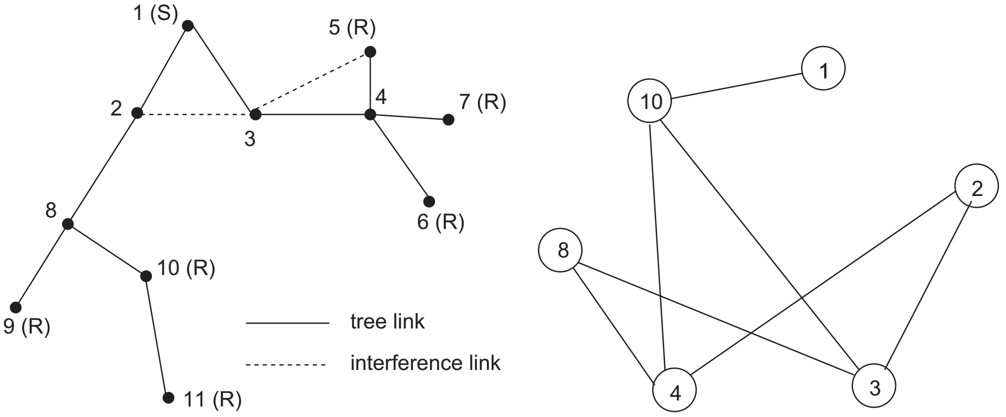


Fig. 2. Example physical tree formation and compatibility graph.

transmission range) fail to provide accurate results. They also proposed an empirical methodology to predict pairwise interference in a network of n nodes using $O(n^2)$ measurements. For two links L_{AB} and L_{CD}, they define "broadcast interference ratio" (BIR) as follows: $BIR = (R_{AB}^{AC} + R_{CD}^{AC})/(R_{AB} + R_{CD})$, where R_{AB}, R_{CD} are the packet delivery rates from A and C at B and D when only A or C broadcasts packets, respectively, and R_{AB}^{AC}, R_{CD}^{AC} are the packet delivery rates at B and D when both A and C broadcast packets simultaneously. The two links do not interfere when BIR is 1, and they interfere if $BIR < 1$.

In IAFS, since multicast forwarders use MAC layer broadcast to send data packets, the interference of interest is *between nodes*. This is in contrast to *between links* in [28], which is to be used for scheduling unicast. To measure the node interference, we perform all pairwise measurements similarly as in [28].[2] However, since we consider nodes and not links in our scheduling, as we explain in the next section, we cannot use the BIR definition as above. We define two nodes A and C to be interfering if C's transmission affects the reception of any of A's children and vice versa. To measure the effect of C's transmission on A's child B, we measure the PDRs from A at B when only A broadcasts packets and when both A and C broadcast packets, respectively, and denote them as R_{AB}^{A}, R_{AB}^{AC}. We then define *interference ratio* (IR) as $IR^{AC} = \min_{B \in \text{children}(A)} IR_{AB}^{AC}$ where $IR_{AB}^{AC} = R_{AB}^{AC}/R_{AB}^{A}$. Thus, node C is an interferer for link $A \to B$ (and it cannot transmit simultaneously with A), if $IR^{AC} < 1$.

Similarly to the original BIR definition, the IR can take any value between 0 and 1. This gives us two different interference models. A schedule that assigns two forwarding nodes the same time slot only if they do not interfere as defined above guarantees fairness among all multicast receivers, assuming all tree links are of equal quality. We call this interference model the *binary model*. The binary model is conservative and may lead to long cycles and hence reduced sender rates (inversely proportional to the cycle lengths) and throughput. In this paper, we also consider the *threshold-based model* which considers node C as an interferer for node A only if $IR^{AC} < IT$ where IT is a selected *interference threshold*. The threshold-based model is more aggressive in finding nodes that can transmit simultaneously. This can lead to a reduced cycle length, which in turn leads to an increased sending rate and potentially increased throughput. However, it can also lead to unfairness as weakly interfering nodes are now competing to transmit in the same slot. Thus the threshold-based model effectively trades reduced fairness for increased throughput. We experimentally study this tradeoff in Section 5.

4.3. Scheduling algorithm

Our scheduling algorithm is based on spatial TDMA, first proposed in [26], and consists of three phases: compatibility matrix (CM) and compatibility graph construction, clique enumeration, and clique selection, similar to [33]. In the following, we explain the three phases of the scheduling algorithm using a single multicast tree in Fig. 2(a) as an example. We will explain how the algorithm can be easily extended for multiple concurrent trees at the end of this section. In this figure, the tree connects the gateway (node 1) to the receivers (nodes 5, 6, 7, 9, 10, 11). The solid lines denote the tree links, while the dashed lines denote links (with two end nodes within transmission range of each other) that are not part of the tree. In this example, $S = \{1\}$, $\mathcal{R} = \{5, 6, 7, 9, 10, 11\}$, and $\mathcal{FG} = \{1, 2, 3, 4, 8, 10\}$, and the scheduling algorithm has to schedule transmissions of nodes 1, 2, 3, 4, 8, 10.

CM construction: As discussed in Section 4.2, since forwarding nodes in IAFS use MAC layer broadcast to send data, the interference of interest is between nodes. This is in contrast to between links, as in the original spatial TDMA algorithm [26] and its later variations (e.g., [21,33]) which assign transmission rights to links. Consequently, in our scheduling algorithm, we define the CM to describe if pairs of *nodes* can transmit simultaneously. Specifically, we define CM as

$$CM = [cm_{ij}], \quad 1 \leqslant i, j \leqslant \|\mathcal{FG}\|, \tag{2}$$

---

[2] Our measurement study [9] shows that multiway nodal interference, i.e., multiple pairwise non-interfering nodes may interfere with each other when transmitting simultaneously, is insignificant.

where

$$cm_{ij} = \begin{cases} 0 & \text{if nodes } i, j \text{ cannot transmit,} \\ & \text{simultaneously,} \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

Two nodes cannot transmit simultaneously if any of the following two conditions hold: (i) any child node of node i is within transmission range of node j (collision), or (ii) one of the two nodes is an interferer for any of the children of the other node (interference). This second condition can be mathematically expressed as: $CM[ij] = 0$ if $IR^{ij}_{i\,\mathrm{child}^k(i)} < IT$ for any k, where $\mathrm{child}^k(i)$ is the kth child of node i. The reason condition (1) (collision) is not treated as a special case of condition (2) is that our interference measurement uses 802.11 CSMA and hence the child node of i may still get an IR above zero, which can cause nodes i and j to be scheduled in the same slot (e.g., if $IR > IT$).

The CM for the tree in Fig. 2(a), assuming $IT = 1$, is

$$CM = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

where the rows correspond to nodes 1, 2, 3, 4, 8, and 10.

Using the CM, we can construct the compatibility graph shown in Fig. 2(b). In this graph, vertices correspond to nodes in $\mathcal{FG}$, and an edge between two vertices denotes that these two nodes can transmit simultaneously.

Clique enumeration: After the CM construction, we enumerate all possible cliques in the compatibility graph. Although the problem of clique enumeration is NP-hard [16], the relatively small size of a WMN makes it easy to solve. For the graph in Fig. 2(b), the set of all cliques is

$$\{\{1\}, \{2\}, \{3\}, \{4\}, \{8\}, \{10\},$$
$$\{1, 10\}, \{2, 3\}, \{2, 4\}, \{3, 8\}, \{3, 10\}, \{4, 8\}, \{4, 10\}\}.$$

Clique selection: We define a schedule as a set of cliques s that fulfills the following two conditions: (i) all nodes in $\mathcal{FG}$ are included in the schedule and (ii) each node in $\mathcal{FG}$ is included only once. Together, these two conditions ensure fairness among the receivers.

Given the list of cliques obtained in the previous step, we can enumerate the set $\mathcal{S}$ of all possible schedules. Each schedule $s \in \mathcal{S}$ corresponds to a cycle length $T_s = \|s\|$, where $\|s\|$ is the number of cliques in schedule s. To maximize spatial reuse, we need to select the schedule s^* that minimizes the cycle length:

$$T_{s^*} = \min_{s \in \mathcal{S}} T_s. \tag{4}$$

Finding the optimal schedule requires an exhaustive search of all possible schedules. To reduce the computation cost, we propose a simple heuristic. The basic idea is to incrementally select and add to our schedule cliques that include many FG nodes, so that the total number of cliques is minimized. However, a straight-forward implementation of the basic idea may not yield good schedules. For example, in Fig. 2(b), two possible schedules that may result from simply selecting cliques based on their sizes are

$$s1 = \{\{1, 10\}, \{4, 8\}, \{2, 3\}\},$$

$$s2 = \{\{4, 10\}, \{3, 8\}, \{2, \}, \{1\}\},$$

which have different cycles. This example shows that arbitrarily breaking ties may not lead to a good solution. To address this issue, we propose a heuristic called least overlapped first (LOF). In LOF, each clique is assigned a *rank*, equal to the number of common nodes this clique has with all other cliques of the same size. Then at each step of the scheduling algorithm, we select the clique that has the smallest rank, among the cliques of the same size. The intuition behind this heuristic is that if we can schedule a large clique that does not have many common nodes with other cliques of the same size, it will be easier to find other cliques with the same size, and form a schedule with a small cycle.

The steps of the LOF algorithm are as follows:

1. set $s = \{\}$;
2. while not all nodes in $\mathcal{FG}$ are included in s
   (a) search for the clique CL_i that includes the maximum number of nodes among the cliques that do not intersect with the members of s. If there are more than one cliques with the same number of nodes, select the one with the lowest rank.
   (b) add CL_i to s.

The above LOF algorithm generates schedule $s1$ in the example of Fig. 2. In this example, clique $\{1, 10\}$ has rank 2 as it overlaps with cliques $\{3, 10\}$ and $\{4, 10\}$, cliques $\{2, 3\}, \{2, 4\}, \{3, 8\}, \{4, 8\}$ have rank 3, and cliques $\{3, 10\}, \{4, 10\}$ have rank 4. Hence, LOF will first select the clique with the smallest rank, $\{1, 10\}$, and then two cliques with rank 3, resulting in schedule $s1$ with a cycle of three slots.

The main idea of the scheduling algorithm is common in many different scheduling algorithms, e.g., [14,20–22], although details may differ. In most of these works, the term *flow contention graph* or *conflict graph* is used, which is the complement term of *compatibility graph* in our paper, and the term *independent set* is then used instead of the term *clique*, which contains nodes that are not connected in the conflict graph. Then finding the minimum number of cliques to cover the compatibility graph is equivalent to finding the minimum number of independent sets to cover the conflict graph. We point out that the two approaches are essentially equivalent. We could have easily used the conflict graph-related terminology as well.

Scheduling multiple trees: The basic algorithm can be easily extended to schedule multiple multicast trees. In this case we simply enumerate each forwarder in each multicast tree separately in the CM construction. If the same node appears as forwarder in k multiple trees, it will appear as k separate nodes in the CM, and be assigned k time slots in a cycle, one for each of its tree appearance.

4.4. Schedule propagation

After computing the schedule, the network operator has to propagate it to the *FG* nodes. The schedule is propagated along the same tree the operator formed for the multicast session. However, since we need reliable delivery of the schedule to the forwarding nodes, each forwarding node unicasts it to its child nodes using hop-by-hop TCP sessions. The file of the schedule size is very small, containing only information of which nodes transmit at each slot of the cycle (and for which multicast session as there can be multiple concurrent ones).

4.5. Clock synchronization

There exist several accurate algorithms for clock synchronization in multihop wireless networks [13,31]. However, IAFS does not require very precise clock synchronization since it can use a relatively large slot time. In our implementation, we synchronize all mesh routers with a chosen leader router near the center of the network. We measure the one-way latency between a sender router and a receiver router and use it to adjust the clock skew at the receiver.

5. Simulation evaluation

5.1. Methodology

We used the Qualnet [29] simulator in our simulation study. We simulated a network of 50 static nodes placed randomly in a $1000\,\text{m} \times 1000\,\text{m}$ area, and we simulated 10 different topologies. The radio propagation range was 250 m and the nominal bit rate was 2 Mbps (the data rate used for broadcast in 802.11 MAC protocol). Qualnet can simulate a very realistic physical model, which is very important for our study, since we are interested in a realistic environment. In our experiments, we used the two-ray propagation model, along with thermal noise and Rayleigh fading. The noise factor was set to 7 dB. The Rayleigh fading model is appropriate for modeling environments with many reflectors, e.g., trees and buildings, where the sender and the receiver are not in line-of-sight of each other. Such environments are common in WMNs.

For each topology, we first measured interference, following the methodology discussed in Section 4.2 (similar as in [28]). In all measurements, each node broadcasts 512-byte packets for 30 s.

We performed the performance evaluation in four steps. In the first step, we compared the performance of our scheduling algorithm using the binary interference model and of 802.11 in an ideal environment, i.e., without fading. The results would show how well our algorithm addresses unfairness of 802.11 due to interference alone. In the second step, we repeated the experiments in a realistic environment, i.e., with fading present, again using the binary interference model. In the third step, we compared how well the two approaches trade off fairness for throughput (PDR) in a realistic environment. For this, we used the threshold-based interference model in the schedule algorithm and vary IT from 0 to 1 (the binary model). Finally, in step four, we evaluated the two approaches for multiple multicast trees.

Note that there is no other practical scheduling algorithm for addressing fairness for multicast in multihop wireless networks. We did not compare our algorithm against any of the existing algorithms for unicast, because we think such a comparison would be unfair, since the objectives are different. Instead, we compared our algorithm with the well-known 802.11 protocol. The 802.11 technology has been widely used in WMNs, and although it is known that it is unfair, we want to see if our solution can offer significant benefits against it, to justify its deployment in future WMNs. Also, note that 802.11 for multicast does not use any control packet, as opposed to unicast, i.e., there are no RTS, CTS or ACK packets to interfere with data transmissions. Hence the comparison is fair, since we also use a single channel to transmit data and we have no control messages. We note again that the schedule is propagated *at the beginning* and there are no schedule updates during the multicast session. For both protocols we assume that the channel is used exclusively for the multicast data, no other transmissions are present.

We used one multicast group with 10 receivers in the first three experiments and two multicast groups with 5 receivers each in the fourth experiment. In each experiment, we ran IAFS and 802.11 on the same SPP-based tree(s). All multicast sources sent CBR traffic consisting of 512-byte packets. In IAFS, the source sending rate $\mathcal{R}$ for each scenario is determined by the schedule cycle length for that scenario, as $\mathcal{R} = 1/(T \times t_s)$, where T is the cycle length and the duration of each slot t_s equals 2.5 ms. We used the same sending rate for 802.11 for a fair comparison.

We use the following metrics in comparing IAFS and 802.11. (1) *Scheduling cycle length:* the number of transmission slots in a cycle. The cycle length in turn dictates the multicast sending rate. (2) *Throughput* (*Kbps*): the number of packets delivered to each receiver divided by the duration of the multicast session. (3) PDR: the number of packets received by each receiver divided by the number of packets sent by the multicast source. PDR equals the throughput divided by the sending rate of the source. (4) *Fairness*: fairness is characterized using the notion of Fairness Index γ defined in [10]. Although this metric is used in [10] to characterize spatial bias across comparable nodes in terms of hop-count and contention, the definition itself is not related to spatial bias and it simply gives an idea of the difference between throughputs received by different receivers. Let the throughput of receiver i be denoted as T_i. To characterize the average fairness over all receivers, we define $\gamma_{\text{avg}} = \textit{average}(\frac{\max(T_j,T_k)}{\min(T_j,T_k)}),\ \forall j,k \in [1..N]$ and to characterize the worst case fairness over all receivers, we define $\gamma_{\max} = \max(\frac{\max(T_j,T_k)}{\min(T_j,T_k)}),\ \forall j,k \in [1..N]$ where N is the number of receivers. Ideally, $\gamma_{\text{avg}} = \gamma_{\max} = 1$, but it is generally larger

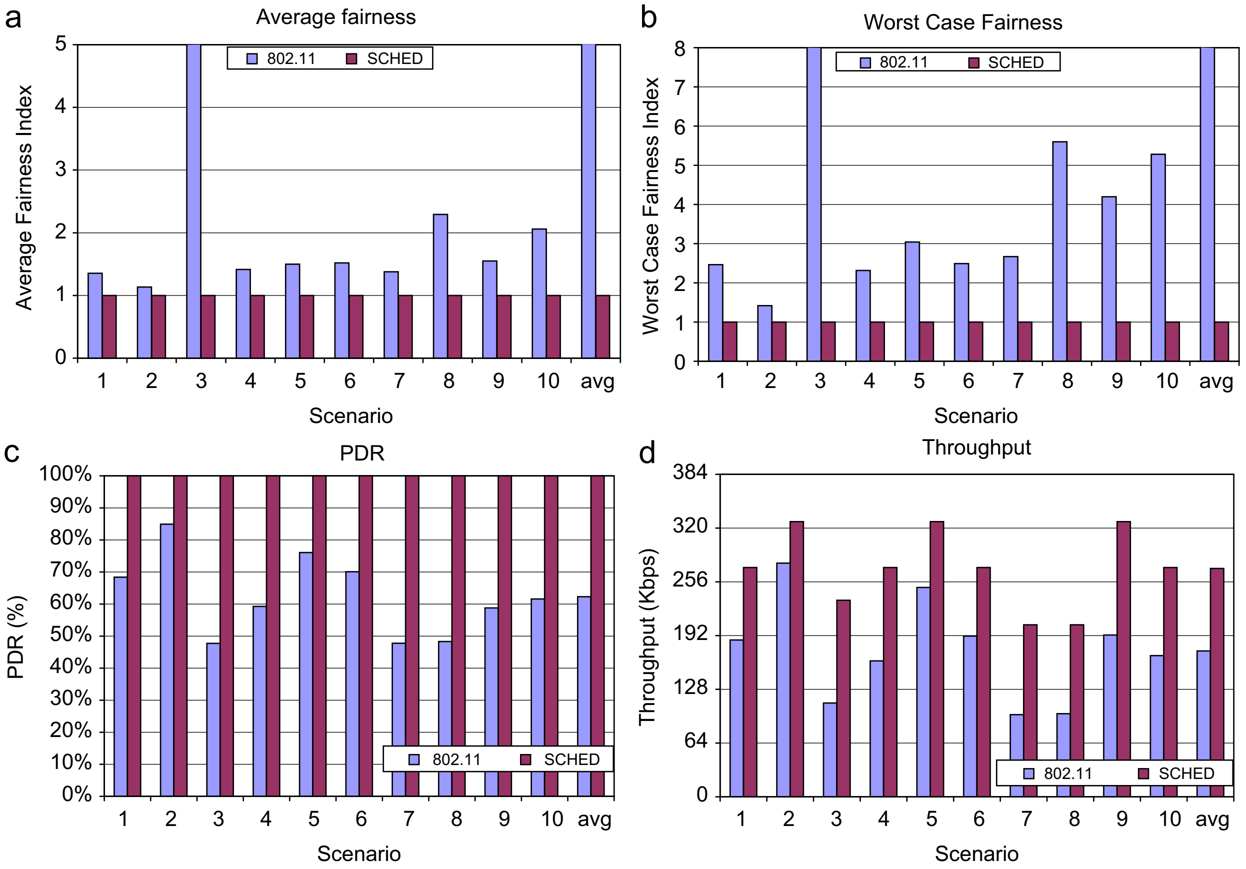


Fig. 3. Average fairness index, worst case fairness index, average PDR and average throughput for the IAFS scheduler (SCHED) and 802.11 in an ideal environment (without fading). The sending rate for each scenario is determined by the cycle length.

than 1 due to unfairness. The larger the γ value is, the more unfair the protocol is.

5.2. Results

Ideal environment: We first compare IAFS with 802.11 in an ideal environment, i.e., without fading and thermal noise. Figs. 3(a)–(d) show the results for 10 different scenarios. The last bar of each figure shows the average value of the corresponding metric over the 10 different scenarios.

Figs. 3(a) and (b) show that both the average and the worst case fairness index are equal to 1 under IAFS for all 10 scenarios, i.e., IAFS always achieves perfect fairness in an ideal environment. In contrast, the same figures show that 802.11 suffers from severe unfairness, even in an ideal environment. Specifically, in one scenario (scenario 3), both fairness indices go to infinity, suggesting that at least one receiver did not receive any packets (zero throughput); and in two other scenarios (scenarios 8 and 10), the worst case fairness index is larger than 5, suggesting that the throughput obtained by some receiver is more than 5 times that obtained by some other.

The fairness properties of the two protocols have a direct impact on the PDR and subsequently the throughput obtained by the multicast receivers. Figs. 3(c) and (d) show that while IAFS achieves 100% PDR in all 10 scenarios, i.e., all receivers received throughput equal to the source sending rate, the PDR for 802.11 can be lower than 50% in some scenarios (scenarios 3, 7, and 8). Averaging over 10 scenarios, the PDR and the throughput under IAFS are 38% and 57% higher than under 802.11, respectively.

In summary, since the experiments here were done in an environment without fading, the only reason for packet losses is interference. Hence interference is an important limiting factor resulting in low throughput and unfairness in multicast over the 802.11 protocol. In contrast, IAFS using a binary interference model always provides an interference-free schedule of transmissions, resulting in perfect fairness among all multicast receivers and achieving 100% PDR.

Realistic environment: Next, we compare IAFS with 802.11 in a realistic environment, with Rayleigh fading and thermal noise. Figs. 4(a)–(d) show the average and worst case fairness indices, and the average PDR and throughput, respectively, over 10 receivers for the 10 different scenarios. We make the following observations.

First, IAFS cannot achieve perfect fairness anymore, and the two fairness indices are larger than 1 in all the 10 scenarios. This is because fading causes random packet losses, causing some links to be lossier than others. Although fading is implicitly taken into account in the SPP-based tree construction which selects good-quality links, bad links cannot be com-

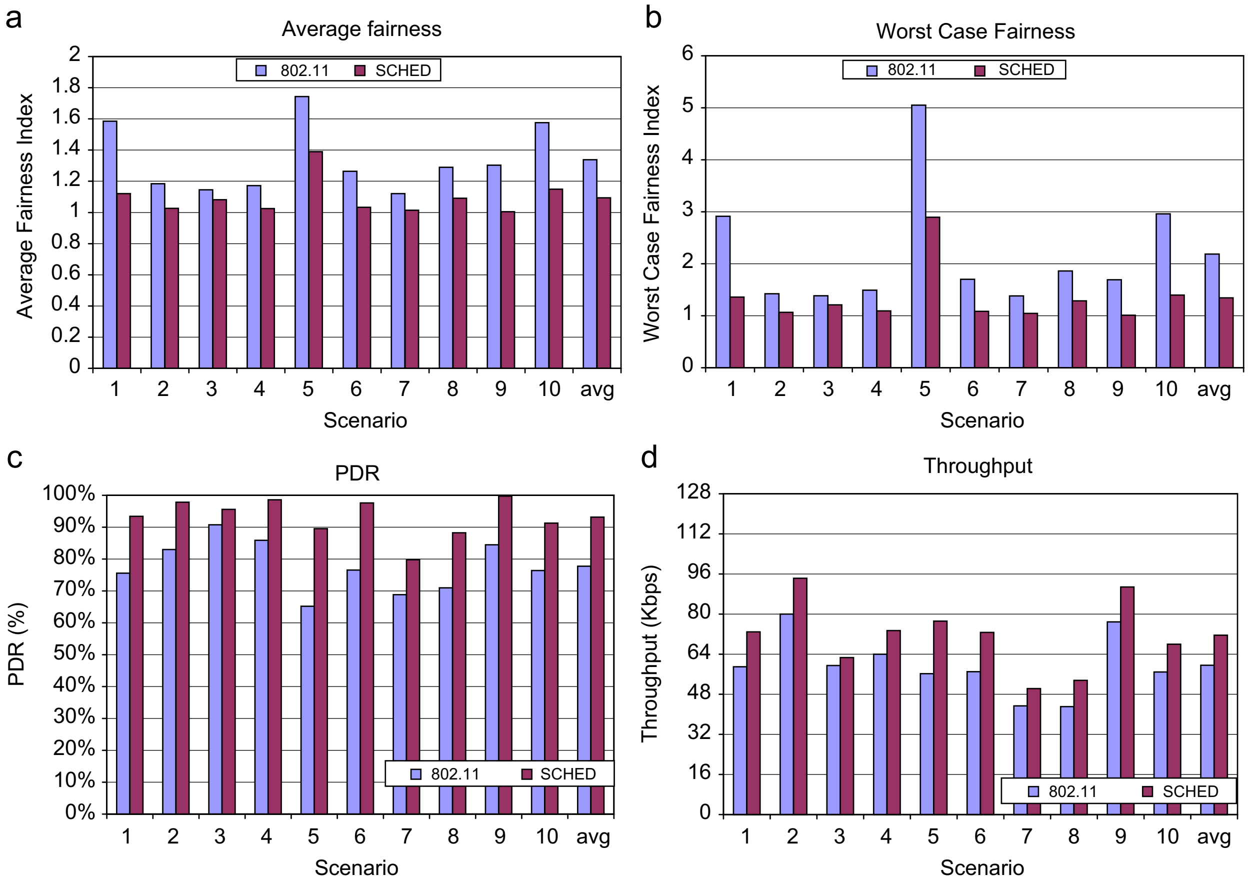


Fig. 4. Average fairness index, worst case fairness index, average PDR and average throughput for the IAFS scheduler (SCHED) and 802.11, in a realistic environment (with fading). The sending rate for each scenario is determined by the cycle length.

pletely avoided. For example, if a receiver can only be reached through some long, lossy link, SPP has no other choice than selecting that link. Such a receiver will always experience packet losses and receive lower throughput than some other receivers.

Second, although scheduling cannot offer perfect fairness in a realistic environment, it still achieves fairness close to perfect in 9 out of 10 scenarios. Other than scenario 5 which has a worst case fairness index close to 3, the average fairness index remains lower than 1.1 and the worst case fairness index remains lower than 1.4 in the rest 9 scenarios. Furthermore, in all 10 scenarios, IAFS still offers higher fairness than 802.11, which can have an average fairness index larger than 1.7 and a worst case index larger than 5 (scenario 5).

Third, IAFS cannot achieve 100% PDR in a realistic environment, but it still achieves 16% higher PDR and 20% throughput than 802.11 averaging over the 10 scenarios.

Fourth, with fading, the sending rate is much lower (the cycle length is much longer) than without fading and hence the throughput also becomes much lower. The throughput averaged over all scenarios is reduced by 73% for IAFS and 65% for 802.11. This is because interference is exacerbated by fading, for example, we found with a transmission range of 250 m, nodes can interfere even at distances larger than 800 m with fading.

We notice that 802.11 performs better in realistic scenarios than in ideal ones. This somewhat surprising behavior is explained as follows: in ideal scenarios, since there is no fading or noise, the only reason for packet loss is interference. Although the interference range is smaller than with fading and more predictable, two nodes that are in interference range of each other will *always* interfere, and hence at least one of them will experience severe packet loss. The 802.11 cannot identify interfering nodes, hence its performance is very poor. On the other hand, when fading is present, although interference range can be larger, the phenomenon is not permanent due to random signal variations over time. Two nodes can be in interference range for some time interval and out of interference range for some other interval. Hence packet loss due to interference is not as severe as in an ideal environment. Now fading is the main reason for packet loss, and this reduces the gains that IAFS can achieve. However, as we noted in previous paragraph, these gains are still important and this is what is expected in a realistic environment. We note again that evaluation in such realistic environments is what is missing in previous works on scheduling, which show different scheduling algorithms to have excellent performance in ideal environments.

Fairness-throughput tradeoff: In this experiment, we compare the two protocols' ability to trade off fairness for through-

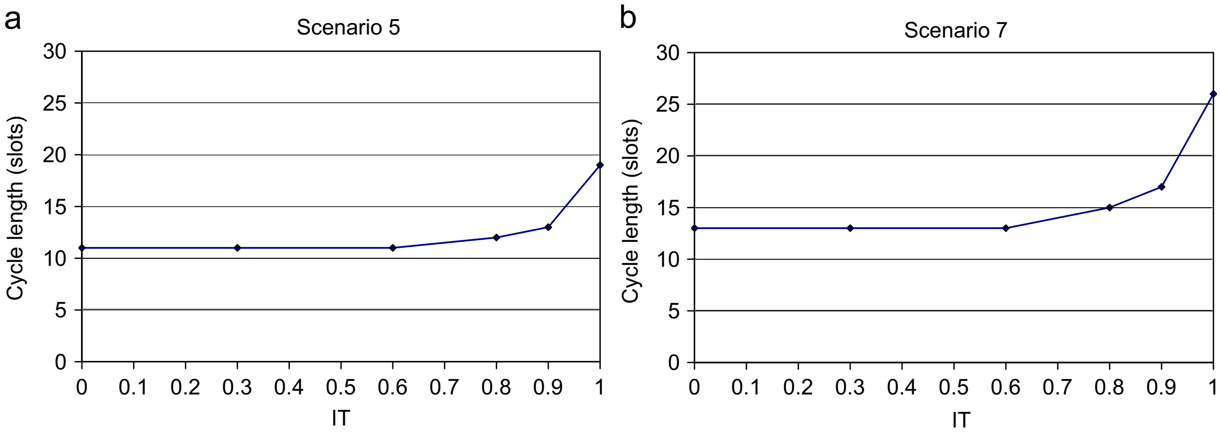


Fig. 5. Cycle length as a function of *IT*. The cycle length determines the corresponding sending rate.

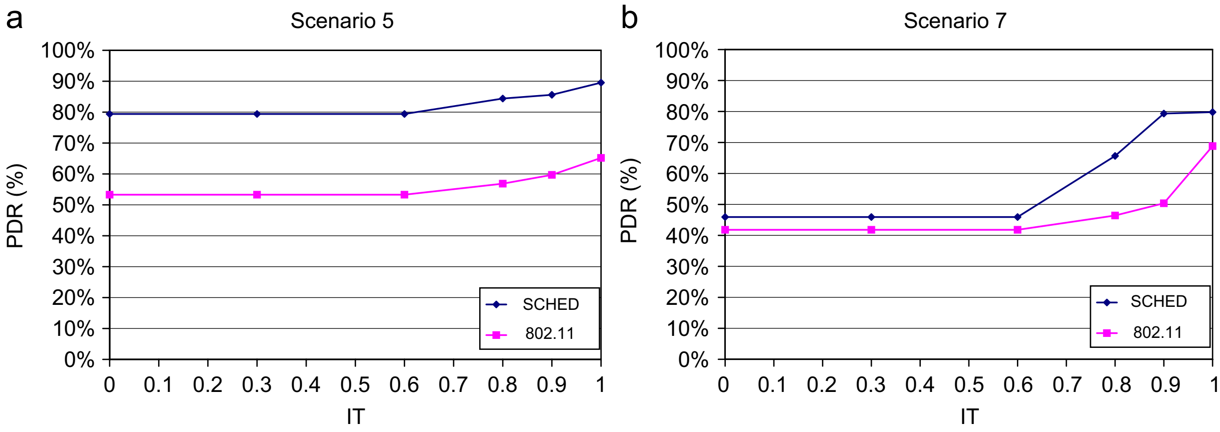


Fig. 6. Average PDR as a function of *IT*.

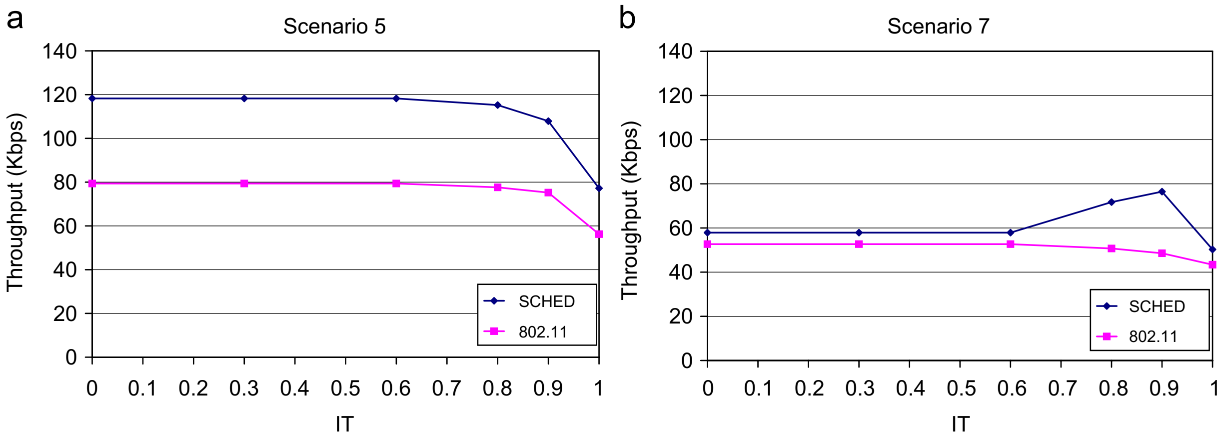


Fig. 7. Average throughput as a function of *IT*.

put in a realistic environment. As discussed in Section 4.2, the binary interference model is conservative and by controlling the IT, the threshold-based model allows weakly interfering nodes to transmit simultaneously, and effectively trades reduced fairness for increased throughput. To measure the tradeoffs, we ran IAFS with the threshold-based model and varied the *IT* from 0 to 1. Based on the results in the previous experiment, we selected two representative scenarios: scenario 5, which is the worst scenario in terms of fairness for both protocols, and scenario 7, which is the best scenario in terms of fairness for both.

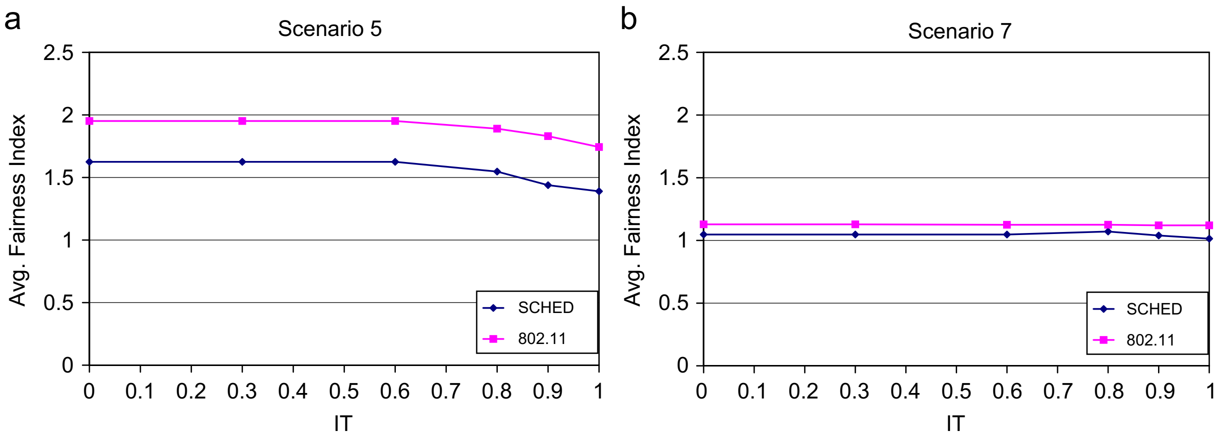


Fig. 8. Average fairness index as a function of *IT*.

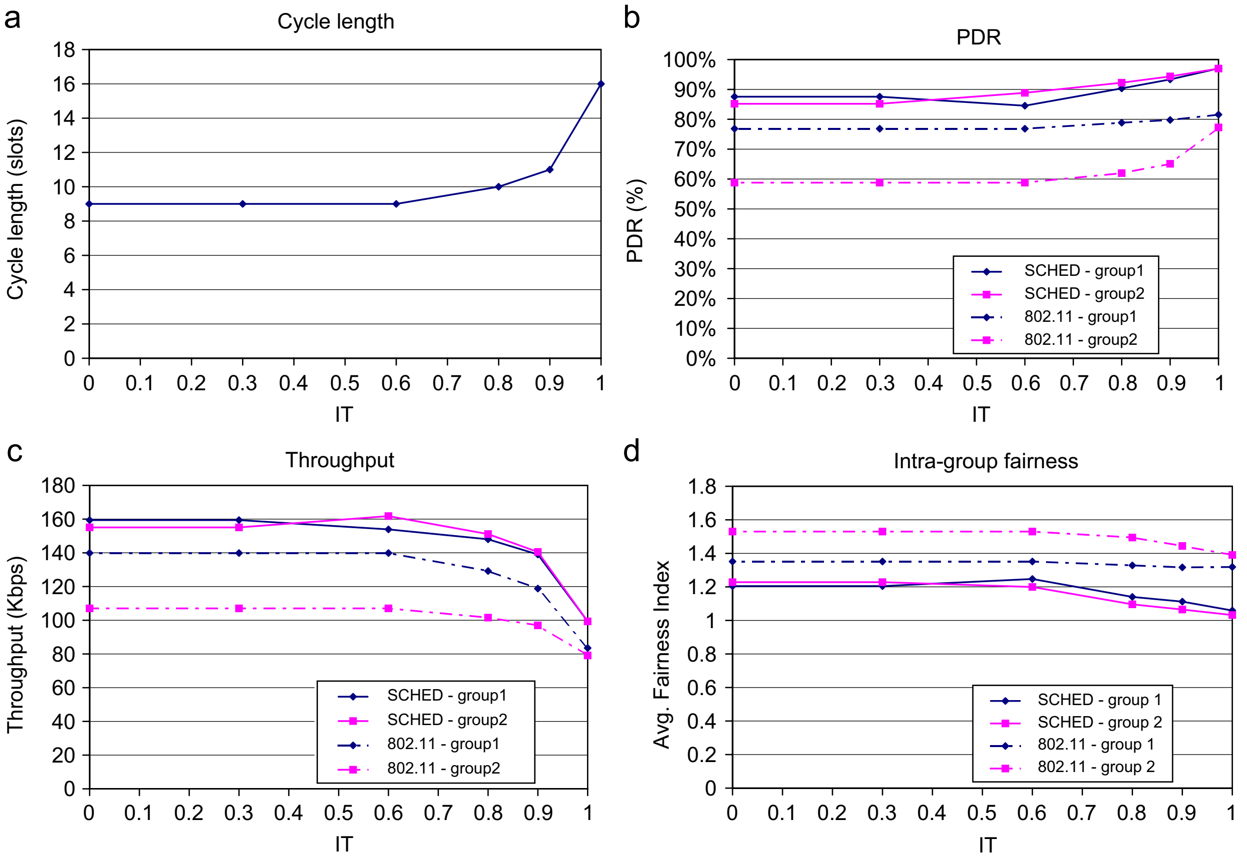


Fig. 9. Cycle length, average PDR per group, average throughput per group, and average fairness index per group under IAFS and 802.11 in scenario 1 with two multicast groups.

Fig. 5 shows that reducing the *IT* value reduces the cycle length and hence increases the sending rate. The largest reduction of the cycle length happens for both scenarios when *IT* is reduced from 1 (binary model) to 0.9. Hence, using the threshold-based model even with a conservative threshold can result in a large increase in the sending rate.

Fig. 6 shows that as *IT* is reduced and the cycle length shortens, the PDR is also reduced, and this reduction can be very large (e.g., PDR is reduced from 80% to 45% in scenario 7 under IAFS). This is because allowing some interfering nodes to transmit simultaneously leads to additional packet losses.

As *IT* decreases, the increased sending rate potentially increases the receiver throughput. Fig. 7 shows that for all *IT* values in the two scenarios, the throughput is higher than that obtained by the binary model ($IT = 1$). However, the increase is not proportional to the cycle length reduction, and in some scenarios, the throughput decreases as the cycle length decreases (e.g., scenario 7). This is explained by the reduction in PDR due to increased interference as *IT* is reduced.

Fig. 8 shows that fairness becomes worse as *IT* is reduced for the two scenarios. For scenario 5, the increase in the average fairness index is large, especially for *IT* values smaller than 0.8. For scenario 7 (the most fair scenario for both 802.11 and IAFS), fairness remains almost unchanged and close to optimal. However, this scenario suffers from very low PDR, which suggests that most links are equally bad and none of them has the potential of getting unfairly higher throughput than the others. In general, the reason for the worsened fairness as *IT* is reduced is that both allowing interfering nodes to transmit together in IAFS and the increased sending rate in 802.11 increase the probability of packet collisions.

Comparing IAFS and 802.11, we observe that IAFS always maintains better fairness than 802.11 and higher throughput and PDR. More importantly, the gap between the two protocols either increases or remains constant as *IT* decreases, suggesting that IAFS achieves better tradeoff between fairness and throughput than 802.11.

Due to the tradeoffs between fairness and throughput, the optimal choice of the IT value for IAFS will be application-specific.

Multiple multicast trees: Finally, we compare the two protocols when there are two concurrent multicast trees, each with five receivers. Here, we measure two types of fairness: *intra-group fairness*—fairness among the members of the same multicast group, and *inter-group fairness*—fairness between different groups. We pick scenario 1 and repeated the previous experiment varying the *IT* from 0 to 1. The results are shown in Figs. 9(a)–(d).

We observe that our algorithm can handle two multicast groups as efficiently as it handles a single group. Specifically, Figs. 9(b) and (c) show that IAFS always achieves higher PDR and throughput than 802.11 for both groups, and Fig. 9(d) shows that the average fairness index for each group stays below 1.23 under IAFS, but increases to above 1.57 for low *IT* values (high sending rates) under 802.11.

Furthermore, under IAFS, the PDR and throughput values for the two groups remain very close to each other, even for high sending rates. This implies that IAFS provides good inter-group fairness in addition to intra-group fairness. In contrast, 802.11 cannot offer inter-group fairness, as there is always a gap between the average PDR or throughput achieved by the two groups, and the gap increases as the sending rate increases. Hence, 802.11 also causes inter-group unfairness, in addition to unfairness among members of the same multicast group.

6. An implementation study

In this section, we compare the performance of a tree-based multicast protocol implemented on top of plain 802.11 and our scheduling algorithm. Our testbed, MAP [23], currently consists of 32 mesh routers spread out across four academic buildings on the Purdue campus (EE, MSEE, PHYSICS and ME). We use the Atheros 5212 based 802.11a/b/g wireless card on each router which is attached to a 2dBi rubber duck omnidirectional antenna with a low loss pigtail. Each mesh router runs Mandrake Linux 10.1 and the open-source *madwifi* drivers are used to enable the wireless cards.

The tree-based multicast protocol is implemented using broadcast sockets. Hence the 802.11 protocol will transmit each data packet exactly once. There is no RTS/CTS exchange prior to the transmission and no ACK/retransmission. For IAFS, we used application-layer scheduling. We schedule particular nodes to transmit in a time slot, but we do not have direct access to the MAC layer and hence cannot turn off some inefficiency that may occur due to backoff or carrier sense. In other words, although we schedule when each node should transmit a data packet according to our LOF heuristic and the measurement-based interference model, once a packet is in the wireless card's queue, we have no control on when the packet is really sent out. We attempted to disable the MAC layer using SoftMAC [27], but the current SoftMAC implementation we experimented with did not provide throughput similar to the original drivers. Thus the results reported here can be improved if we are given low level access to the radio card to turn off carrier sense and backoff and implement our solution at the MAC layer.

In both cases, the nominal bit rate of the wireless cards was set to 2 Mbps. The application data rate was again determined by the scheduling cycle and was the same for both protocols.

We experimented with different topologies and got similar results. Due to lack of space, we explain here the results by considering one particular example. In this example, we considered a multicast tree with node 11 of MAP as the source and nodes 3, 5, 7, 13, 14, 28 as the six group members (see Fig. 10 for the locations). We performed SPP measurements to build a multicast tree from source 11 to all these receivers. The *FG* nodes were 3, 5, 16, 24, 28, 29, 30. For the interference-aware scheduled multicast, we performed interference measurements following the methodology described in Section 4.2. The scheduler at the source node used these measurements along with the binary interference model to drive the scheduling algorithm, which had to schedule eight nodes (seven *FG* nodes plus the source node 11). The final schedule generated showed a clique of three nodes (28, 16 and 30) that could be scheduled in a single slot while the remaining nodes transmitted in separate slots due to the interference characteristics.

The results are shown at Figs. 11(a) and (b). Fig. 11(a) shows the throughputs achieved by each of the six receivers with 802.11 and IAFS. It shows that our scheduling algorithm achieves on average 38% higher throughput among the receivers than 802.11. Fig. 11(b) shows the average and worst case fairness index for each of the two protocols. IAFS has an average fairness index of 1.24 and a worst case fairness index of 1.69, while the respective values for 802.1 are 1.58 and 2.93. Hence, in spite of the implementation limitations, the results agree with the simulation results, showing that IAFS improves fairness and increases throughput compared to 802.11.

Fig. 10. Top view topology of the MAP testbed.

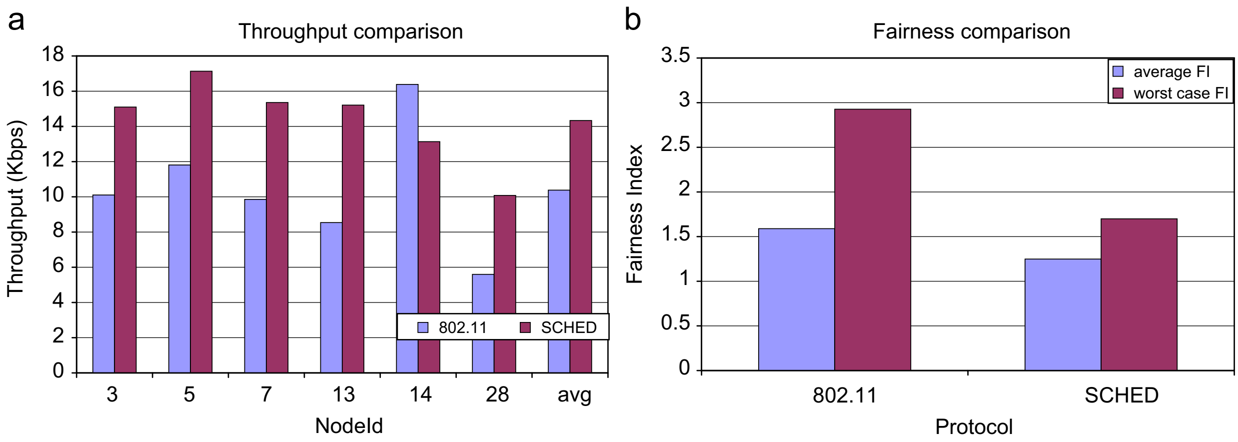


Fig. 11. Results from an implementation study.

7. Related work

There is a lot of work on scheduling for unicast in multihop wireless networks, e.g., [20–22]. In [20,21], Luo et al. formulate the problem of ad hoc fair queuing to maximize spatial reuse while ensuring fairness. In both these works, the focus is on a centralized algorithm, although a distributed protocol implementation is also provided. In [22], the same authors provide a distributed, localized, self-coordinating approach to the fair queuing problem. In these works, the term *flow contention graph* is used, which is the complement term of *compatibility graph* in our paper, and the term *independent set* is used instead of the term *clique*.

In [15], Gambiroza et al. propose a per-mesh-router fairness scheduling algorithm for unicast in WMNs. As opposed to it, our algorithm aims to provide per-client fairness for multicast clients. In [33], Salem et al. also propose a scheduling algorithm for per-client fairness in WMNs. This algorithm is also based on the CM matrix and uses the same three phases as ours: construction of the CM matrix and compatibility graph, clique construction and clique selection. In fact, their algorithm was used as a guideline for our algorithm. However, their algorithm is proposed for unicast and it is link-oriented, hence the details in the three phases are different from our algorithm. In addition, neither [15] nor [33] consider interference and fading in a realistic environment.

Compared to unicast, there is little work on scheduling for multicast in multihop wireless networks. The common feature in all the existing works is that they do not consider fairness.

In [14], Ephremides and Truong are the first to study the problem of scheduling broadcast transmissions in multihop wireless networks. Since we assume that multicast uses broadcast at the MAC layer, the problem of scheduling transmissions along a multicast tree can be considered as a special case of the problem of scheduling broadcast transmissions along the whole network. In this work, the authors prove that the problem of scheduling broadcast transmissions with the goal of *throughput optimization* is NP-hard, under the assumption of two-hop interference. They also propose a distributed approximation algorithm that also uses the notion of independent set, similar to many other works. As the authors point out, their algorithm optimizes throughput but it has the disadvantage of *unfairness*, which is the main goal of our work. Also, we are not interested in a distributed implementation, since the schedule is computed only once for each particular event that is to be multicast.

In [6], Chaporkar et al. propose an adaptive strategy for maximizing throughput in MAC layer multicast subject to maintaining the system stability. Their algorithm is localized allowing a source to decide when to transmit using simple computations based only on limited information about current transmissions in its neighborhood. They compare their policy against an offline algorithm that uses past, present and future global knowledge about the network state, and show that it achieves the same throughput. However, they assume that all the receivers are within one hop distance from the sender, hence their algorithm is not appropriate for multihop WMNs.

Two recent works are [2,7]. In [2], Bhatia and Li study the multicast throughput optimization problem in multihop wireless networks. They consider two different transmission models, namely broadcast-based and unicast-based multicast, and two multicast models, namely one multicast tree and many multicast trees per session. For each of the four combinations they look for the optimal tree and schedule to maximize throughput. However, they only provide theoretical results, showing that the problem in each case is NP-hard and proposing several approximation algorithms under different assumptions, which in many cases are very different from ours. They consider a futuristic MAC model, which supports multiple rates for multicast, while we consider a single rate network since current 802.11 supports only one rate for multicast. They assume the use of code division multiple access (CDMA), hence there is no interference among simultaneous transmissions of neighbor nodes. Finally, they assume an idealistic physical channel without fading; this allows them to use the assumption of *unit disk graph*, which is a common assumption in theoretical works in multihop wireless networks, but fails to model the real world. Because of all these assumptions, the results in this work, although novel, can only serve as upper bounds for a realistic study.

In [7] Chou et al. perform scheduling for low latency multicast/broadcast in WMNs. Similar to our work they also decouple tree formation from scheduling for practical cases, since the joint problem is NP-hard. However, there are many differences with our work. First of all the main goal of the scheduling algorithm is very different: their goal is to minimize broadcast latency; we seek to provide fairness among all multicast receivers, while exploiting spatial reuse for reducing the cycle length. Due to their goal, their algorithm requires a very fine-grained timing, since each node needs to know when it will receive a packet before it schedules a transmission for that packet. Similar to [2], they consider a futuristic MAC layer which supports multiple rates for multicast and the sender can specify only a subset of the nodes within transmission range to receive the transmitted packet. Instead we assume that only one rate is available, following the current MAC technology, and a broadcast is received by all nodes within transmission range. The authors propose two new algorithms for tree construction, exploiting their assumptions about the MAC layer. Instead we use SPP metric, which was shown in [32] to give the highest throughput among many link-quality-based metrics for multicast. Finally, they consider a distance-based interference model, where two nodes interfere if one of them is within a specific distance from any of the children of the other node. However, [28] showed that these heuristics do not reflect accurately the reality. Our work is the first to consider measurement-based interference for a scheduling algorithm.

8. Conclusion and future work

In this paper, we identified interference and fading as two major causes for unfairness and low throughput in 802.11 WMNs, and presented IAFS, a MAC layer solution that addresses unfairness of multicast service due to interference. IAFS consists of a measurement-based interference model and an interference-aware scheduler that maximizes the spatial reuse for high throughput. Since the measurement-based interference model takes into account the effects of fading, IAFS addresses unfairness indirectly caused by fading. Detailed simulations show that in a realistic scenario (with fading), IAFS achieves close to perfect fairness among multicast receivers when using the binary interference model, and the significant gain in fairness, higher throughput and PDR over 802.11 either increases or remains constant when allowing weakly interfering nodes to transmit simultaneously by using a threshold-based interference model. We are currently investigating how we can implement our solution at the MAC layer. In our future work, we plan to extend IAFS to support multirate multimedia multicast in WMNs and to study a unified fairness scheduler for both unicast and multicast traffic.

Acknowledgment

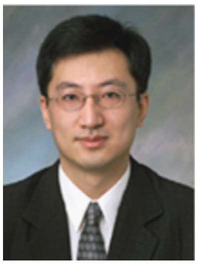
This work was supported in part by NSF Grants ANI-0338856 and CNS-0626703.

References

[1] Bay area wireless users group, ⟨http://www.bawug.org⟩.
[2] R. Bhatia, L. Li, Characterizing achievable multicast rates in multi-hop wireless networks, in: Proceedings of ACM Mobihoc, 2005.
[3] S. Biswas, R. Morris, ExOR: opportunistic multi-hop routing for wireless networks, in: Proceedings of ACM SIGCOMM, 2005.
[4] M. Cagalj, J.-P. Hubaux, C. Enz, Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution, in: Proceedings of ACM MobiCom, September 2002.
[5] Champaign-Urbana community wireless network, ⟨http://www.cuwireless.net⟩.
[6] P. Chaporkar, A. Bhat, S. Sarkar, An adaptive strategy for maximizing throughput in mac layer wireless multicast, in: Proceedings of ACM Mobihoc, 2004.
[7] C.T. Chou, A. Misra, J. Qadir, Low latency broadcast in multi-rate wireless mesh networks, IEEE J. Selected Areas Comm. (JSAC) 24 (11) (2006).
[8] D.S.J.D. Couto, D. Aguayo, J. C. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proceedings of ACM MobiCom, 2003.
[9] S.M. Das, D. Koutsonikolas, Y.C. Hu, D. Peroulis, Characterizing multi-way interference in wireless mesh networks, in: Proceedings of ACM WiNTECH 2006, September 2006.
[10] S.M. Das, H. Pucha, Y.C. Hu, Symmetrical fairness in infrastructure access in multi-hop wireless networks, in: Proceedings of ICDCS, 2005.
[11] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of ACM MobiCom, 2004.
[12] R. Draves, J. Padhye, B. Zill, Comparison of routing metrics for static multi-hop wireless networks. in: Proceedings of ACM SIGCOMM, 2004.
[13] J. Elson, D. Estrin, Time synchronization for wireless sensor networks, in: PDPS Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 2001.
[14] A. Ephremides, T.V. Truong, Scheduling broadcasts in multihop radio networks, IEEE Trans. Comm. 38 (4) (1990).
[15] V. Gambiroza, B. Sadeghi, E.W. Knightly, End-to-end performance and fairness in multihop wireless backhaul networks, in: Proceedings of ACM MobiCom, 2004.
[16] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, New York, 1979.
[17] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Trans. Inform. Theory 46 (2) (2000) 388–404.
[18] M. Kodialam, T. Nandagopal, Characterizing achievable rates in multihop wireless networks: the joint routing and scheduling problem, in: Proceedings of ACM MobiCom, 2003.
[19] S.-J. Lee, M. Gerla, C.-C. Chiang, On-demand multicast routing protocol, in: Proceedings of IEEE WCNC, September 1999.
[20] H. Luo, S. Lu, A topology-independent fair queueing model in ad hoc wireless networks, in: Proceedings of IEEE ICNP, 2000.
[21] H. Luo, S. Lu, V. Bharghavan, A new model for packet scheduling in multihop wireless networks, in: Proceedings of ACM MobiCom, 2000.
[22] H. Luo, P. Medvedev, J. Cheng, S. Lu, A self-coordinating approach to distributed fair queueing in ad hoc wireless networks, in: Proceedings of IEEE Infocom, 2001.
[23] Mesh@Purdue, ⟨http://www.engineering.purdue.edu/MESH⟩.
[24] MIT Roofnet, ⟨http://www.pdos.lcs.mit.edu/roofnet⟩.
[25] C. de Morais Cordeiro, H. Gossain, D.P. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, IEEE Network 17 (1) (2003).
[26] R. Nelson, L. Kleinrock, Spatial tdma: a colllision-free multihop channel access protocol, IEEE Trans. Comm. 33 (9) (1985).
[27] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, D. Grunwald, Softmac: a flexible wireless research platform, in Proceedings of HotNets IV, 2005.
[28] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, B. Zill, Estimation of link interference in static multi-hop wireless networks, in: Proceedings of IMC, 2005.
[29] QualNet, ⟨http://www.scalable-networks.com⟩.
[30] A. Rao, I. Stoica, An overlay mac layer for 802.11 networks, in: Proceedings of Mobisys 2005, April 2005.
[31] K. Romer, Time synchronization in ad hoc networks, in: Proceedings of ACM MobiHoc, 2001.
[32] S. Roy, D. Koutsonikolas, S.M. Das, Y.C. Hu, High-throughput multicast routing metrics in wireless mesh networks, in: Proceedings of ICDCS, 2006.
[33] N.B. Salem, J.-P. Hubaux, A fair scheduling for wireless mesh networks, in: Proceedings of WiMesh, 2005.
[34] Seattle wireless, ⟨http://www.seattlewireless.net⟩.
[35] Southampton wireless network, ⟨http://www.sown.org.uk⟩.
[36] K. Xu, M. Gerla, S. Bae, How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? in: Proceedings of Globecom, 2002.

Dimitrios Koutsonikolas is currently a Ph.D. student in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a B.Engg. degree from the National Technical University of Athens (NTUA), Greece. His research interests include wireless ad hoc, mesh and sensor networks. He is a recipient of the Ross Fellowship, Purdue University, 2004, and a Tellabs Fellowship, Center for Wireless Systems and Applications (CWSA), Purdue University, 2006.

Saumitra M. Das is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a M.S. degree from Carnegie Mellon University, USA and a B.Engg. degree from the University of Bombay, India. His research interests include cross-layer system design for multihop wireless networks, scalable routing strategies in wireless ad hoc networks, and mobile robotics.



Y. Charlie Hu is an Associate Professor of Electrical and Computer Engineering and Computer Science at Purdue University. He received his M.S. and M.Phil. degrees from Yale University in 1992 and his Ph.D. degree in Computer Science from Harvard University in 1997. From 1997 to 2001, he was a research scientist at Rice University. Dr. Hu's research interests include operating systems, distributed systems, wireless networking, and parallel computing. He has published over 100 papers in these areas. Dr. Hu received the NSF CAREER Award in 2003. He served as a TPC Vice Chair for ICDCS 2007 and the 2004 International Conference on Parallel Processing and a co-founder and TPC co-chair for the International Workshop on Mobile Peer-to-Peer Computing. Dr. Hu is a member of USENIX and ACM and a senior member of IEEE.