Distributed Fair Scheduling for Wireless Mesh Networks Using IEEE 802.11

Janghwan Lee, *Member, IEEE*, Hyunsoo Yoon, *Member, IEEE*, and Ikjun Yeom, *Member, IEEE*

***Abstract*—In IEEE-802.11-based wireless mesh networks (WMNs), unfair bandwidth sharing may arise, because the carrier sense multiple access with collision avoidance protocol is designed to provide per-station fairness only in one hop. As the hop count from a mobile client to the gateway node increases, the throughput of the node drastically decreases. In this paper, we propose a fair bandwidth allocation scheme for multiradio multichannel WMNs. This scheme provides fair bandwidth sharing among the nodes in a WMN, regardless of their hop distance from the gateway node. To achieve fairness, we first estimate the number of active nodes attached to each router and calculate the effective weights of routers based on the estimation. Then, we differentiate their contention window using their weights. For this method, we derive a multihop packet collision model. The proposed scheme is fully distributed and does not require any global information. Through an extensive simulation study, we show that our scheme ensures per-node fairness without loss of the total aggregate throughput.**

***Index Terms*—Fairness, IEEE 802.11, wireless mesh network (WMN).**

I. Introduction

A WIRELESS MESH NETWORK (WMN) is a wireless ad hoc network that consists of three types of nodes: 1) client nodes; 2) router nodes; and gateway nodes. To provide a last-mile broadband Internet access, intermediate wireless routers in a WMN relay packets from client nodes to gateway nodes that are connected to the wired Internet. Recently, the WMN technology has emerged as an access network not only in urban centers but in rural areas as well, because it is easy to deploy and manage. Some testbeds have already been deployed and studied in university campuses and metropolitan areas [2], [3].

There are several candidate network technologies for implementing a WMN, such as IEEE 802.11 and IEEE 802.16. Among these technologies, IEEE 802.11 is a promising candidate due to its low deployment and communication costs. IEEE 802.11 was initially developed to provide a wireless access link by one-hop communications, and several problems have arisen in extending it for multihop communications. One of the main problems in WMNs based on IEEE 802.11 is unfair bandwidth sharing among flows with different hop distances to the gateway node. As the hop count from the gateway increases, the throughput decreases, because flows with large hop counts may experience more contention.

In this paper, we propose a new scheme to provide fair bandwidth sharing among nodes in a WMN.[1] The proposed scheme consists of two parts. The first part is for estimating the weight of each node, which is based on the amount of aggregate traffic that belongs to that node. Estimating the node weight is similar to the traditional flow measurement problem in the Internet area. Research works have been focused on estimating aggregate flow statistics on wired backbone links with reduced packet sampling. These techniques are not directly applicable to our scheme, because the number of flows in an access link is highly dynamic, and it is also required to estimate them with high precision. In this paper, we derive a simple technique for estimating the weight of nodes using the sending rates of leaf nodes.

The second part is for scheduling nodes to satisfy their weights, which are given by the results of the first part. Scheduling nodes with given weights is split into uplink and downlink cases. In [6] and [10], it has been shown that the downlink scheduling can effectively be handled with simple queue management schemes. In this paper, we focus on the uplink scheduling under the IEEE 802.11 distributed coordination function (dcf). In [9], an effective algorithm has been proposed to differentiate the contention window to achieve a given set of priorities among nodes in a single-hop network. We extend this algorithm for multihop environments.

This paper makes the following significant contributions. First, this paper proposes a new scheme for providing fair bandwidth sharing in a WMN without loss of utilization. The proposed scheme is highly scalable, because it is fully distributed and requires no global information. Second, to develop the scheme, this paper presents an efficient technique for estimating the weight of intermediate nodes based on their traffic. In a highly dynamic network, it is difficult to estimate the number of active nodes, because the temporal dormancy and the left of a node are hard to distinguish. The proposed technique utilizes the aggregate and individual sending rates of active nodes to estimate the weight of nodes without distinguishing the dormancy and the left of a node. Third, this paper considers a collision model in multihop environments. There have been several studies for modeling collision in a single-hop environment, but they

Manuscript received November 8, 2009; revised May 10, 2010 and August 9, 2010; accepted August 11, 2010. Date of publication September 13, 2010; date of current version November 12, 2010. This work was supported in part by the National Research Foundation of Korea under Grant 2009-0080319 through the Korean government (Ministry of Education, Science and Technology). A preliminary version of this paper was presented at the IEEE MASS 2008 [1]. The review of this paper was coordinated by Prof. H. Hassanein.

J. Lee and H. Yoon are with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305 701, Korea (e-mail: janghwan.lee@gmail.com; hyoon@cs.kaist.ac.kr).

I. Yeom is with the Department of Computer Engineering, Sungkyunkwan University, Suwon 440-746, South Korea (e-mail: ikjun@skku.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2076319

[1] In this paper, we assume that there exists one gateway node in a WMN. Traffic engineering and routing issues among multiple gateway nodes are beyond the scope of this paper.

are not suitable for a WMN. In this paper, we extend them for multihop environments.

The rest of this paper is organized as follows. In Section II, we review the related work and describe the motivation of our scheme. In Section III, we describe the formal notation of fairness objective. We describe our node number estimation scheme and scheduling algorithm in Section IV. In Section V, we evaluate our scheme through extensive *ns-2* [17] simulation. Finally, we summarize the result and discuss future research directions in Section VI.

II. Related Work

In recent years, several works have focused on the unfairness problem of multihop wireless networks. Gambiroza *et al.* proposed a centralized scheme in [5]. In this scheme, each mesh router collects information on the global topology, including link capacities and offered traffic, and then calculates the optimal sending rate based on the information. Then, each node in the network limits its ingress rate according to the given optimal rate. This scheme is effective in solving the unfairness caused by hidden and exposed terminal problems, because each router knows the global information. However, it is hard to apply for a large-scale WMN due to its high complexities, and offered traffic cannot simply be defined in real networks. Giannoulis *et al.* also proposed a centralized scheme [20]. For a given mesh network topology, this scheme calculates the optimal radio path, source distribution, and channel assignment to maximize a target utility function by solving a nonlinear programming problem. This scheme provides an accurate solution for various fairness objectives, but the complexity of computing is an NP-complete problem. Thus, this scheme can be used only for stationary source distribution. However, the traffic of a WMN can be dynamic due to user mobility, and our scheme was designed to adapt to traffic variation.

Several queue management schemes for fairness in a WMN has been proposed. It has been shown that weighted per-flow queue management schemes are effective in providing fairness in a WMN [6], [11]. However, per-flow queuing requires high implementation cost and may induce the scalability problem in large-scale WMNs. In addition, per-flow queuing itself without a Media Access Control (MAC) layer support may waste a significant amount of bandwidth when there are overinjected packets and fails to provide fair bandwidth sharing, because queuing cannot resolve the MAC-layer contention. We will look at this problem in Section V.

In summary, centralized schemes may provide accurate fairness, but their complexities make them hard to employ in large and dynamic networks. Decentralized-queuing-based schemes are simple and easy to deploy, but they may cause low utilization. Our proposed scheme provides accurate fairness with a distributed fashion while maintaining high utilization.

III. Network Model and Problem Formulation

A. Network Model

The traffic in a WMN is skewed such that most packets flow toward or from the gateway node. If we assume that there is no traffic between mesh nodes, except for the gateway, then we can model a mesh topology as a tree structure $T = (V, E)$. In the tree, we model the root node as the gateway node and other nodes as mesh routers. Let $V = \{1, \ldots, n\}$ be the set of mesh nodes and $E = \{(i, j) : i, j \in V\}$ be the set of links (see Fig. 1). Every link (i, j) has a weight $f_{(i,j)}$, which is defined as the fraction of the link capacity to the parent node.

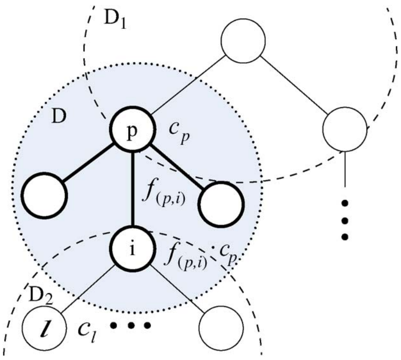


Fig. 1. Network model.

In this paper, we consider a multiradio multichannel environment, as described in [8]. Each node has two network interfaces that operate on orthogonal channels: 1) one channel for uplink and 2) another channel for downlink. With this model, we can eliminate interference over more than two-hop distance by assigning orthogonal channels. Similar assumptions have been made in [7] and [8]. In this model, the hidden terminal problem exists only in one-hop distance. We can separate each hop as an independent collision domain $D = (V_D, E_D)$, which is a subtree that consists of one parent as a root node, its child nodes, and the links between them (as illustrated in the shadow region in Fig. 1). If we assign different channels for collision domain D_1 and D_2, transmissions in the collision domain D are not interfered by transmissions in D_1 and D_2, because an interference range of a node is usually less than twice its transmission range [22]. To maximize spatial reuse, we consider employing the static channel assignment scheme in [7], [8], and [12], but the dynamic channel assignment schemes in [23] and [24] are also applicable to handle traffic dynamics.

While deriving our scheme, we assume that traffic is injected to a network only through leaf nodes. This assumption makes it simple to calculate a weight for each node, but it might be unrealistic, because mobile clients may directly be attached to an intermediate node. To cover this case with our assumption, we consider that each intermediate node has a virtual leaf node to host mobile clients. To emulate a virtual leaf node, an intermediate node has two separate queues. One queue is for packets from other intermediate nodes and/or leaf nodes, and the other queue is for packets from the mobile clients directly attached to it. Then, it calculates the weight for each queue and emulates virtual contention.

B. Problem Formulation

For each collision domain, there exist one parent node and a set of child nodes. Let the capacity of parent node p be c_p. If child node i shares c_p by its weight $f_{(p,i)}$, then the bandwidth received at node i is $f_{(p,i)}c_p$, where $\sum_{k\in E_D} f_k = 1$ for all D. Hence, the bandwidth of leaf node l becomes

$$c_l = C \prod_{(i,j)\in P_l} f_{(i,j)} \tag{1}$$

where C is the capacity of the root node, and P_l is the set of links on the path from l to the root node (the gateway node). On the current IEEE 802.11 MAC, all the child nodes in D share equal bandwidth. Then, c_l is calculated by

$$c_l = C \prod_{(r,j)\in P_l} \frac{1}{|V_{D_r}| - 1} \tag{2}$$

where D_r is the collision domain that contains node r as the parent node, and $|V_{D_r}| - 1$ is the number of child nodes in D_r. Based on (2), we can observe that, in the current IEEE 802.11 standard MAC protocol, the uplink bandwidth of a leaf node is determined by the hop distance and the number of nodes in collision domains along the path to the gateway node.

The objective of our scheme is to find a proper weight $f_{(i,j)}$ that ensures fairness and to schedule the nodes with their weights. Here, note that we consider the min–max fairness for the fairness model in this paper. Time-based fairness or proportional fairness is often used in wireless networks, particularly in one-hop multirate networks [16]. In this paper, however, we focus on the bandwidth allocation among the stationary mesh nodes with stable channel states. In this network model, we believe that the min–max fairness fits better than the proportional fairness.

Under the min–max fairness, we can classify nodes into two groups. The nodes in the first group require less bandwidth than their fair share, and the nodes in the second group require more bandwidth than their fair share. For simplicity of modeling, we assume that every node requires bandwidth more than its fair share. This assumption is reasonable when we consider Transmission Control Protocol (TCP) flows that observe their bottleneck at wireless links. Then, a fair share is simply defined as

$$c_i = c_j \quad \text{for all active leaf nodes } i, j \tag{3}$$

which means that every active node receives the same bandwidth allocation.

IV. Algorithm Description

Our algorithm consists of two parts. The first part is node weight estimation, and the second part is weighted scheduling among mesh nodes.

A. Node Weight Estimation

The weight of a node is proportional to the amount of traffic that flows through the node. In this paper, we define the amount of traffic as the number of active nodes, because our goal is to provide per-node fairness to all active nodes. A simple and intuitive method of estimating the number of active nodes that belong to an intermediate node is to count the number of nodes that have sent at least a packet within a given time interval using flow ID or Internet Protocol (IP) address. However, this method has four major limitations. First, it is hard to choose the proper time interval. With a short interval, it may underestimate the number of active nodes when the sending rate is low. With a long interval, it may count nodes that have already left the network (or the leaf node). Second, the estimated number may not be accurate in a highly dynamic network, because the estimation is based on the history of packets received. Third, every intermediate node has overhead to aggregate estimated node counts of its child nodes and to send up the value to its parent node with an explicit message. Forth, neither flow ID nor IP address can represent an active node, because each mesh node may have several mobile nodes that have different IP addresses and multiple flows. Furthermore, it can be overhead to maintain the global IP and flow ID information. On the other hand, our scheme is a simple distributed MAC-layer solution, which needs only one-hop information. In this section, we propose an accurate and fast scheme for estimating the number of active nodes.

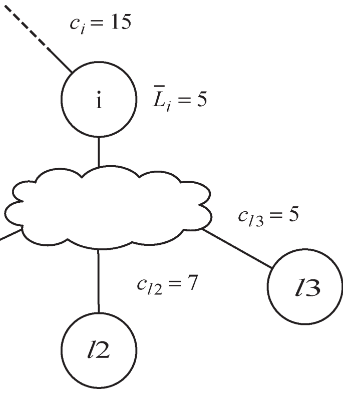


Fig. 2. Example of a weight estimation.

In this section, we describe how we can estimate the number of active leaf nodes in a WMN using the average sending rate of leaf nodes.

Let $D_r = (V_{D_r}, E_{D_r})$ be a collision domain that contains r as the root node. For each child node i in V_{D_r}, the root node r can estimate the number of active leaf nodes n_i based on

$$n_i = \frac{c_i}{\overline{L}_i}, \qquad i \in V_{D_r}/\{r\} \tag{4}$$

where c_i is the sending rate of child node i, and n_i and $\overline{L}_i$ are the estimated number of active leaf nodes and their average sending rate in the subtree, which has node i as the root node, respectively. If node i is a leaf node, $\overline{L}_i = c_i$, and n_i is 1. Intuitively, we can comprehend (4) that we can get the number of active leaf nodes by dividing the aggregate sending rate of leaf nodes by the average of their individual sending rate. One example of estimation is illustrated in Fig. 2. In this case, the average of c_{l1}, c_{l2} and c_{l3} is 5, and their aggregate throughput is 15. Then, the number of active nodes is calculated by $15/5 = 3$.

In (4), c_i can simply be measured by each intermediate node. An intermediate node just measures the input rate of each child node using rate estimation schemes such as exponential weighted moving average or the time-based rate estimation algorithm.

For $\overline{L}_i$, each leaf node piggybacks its sending rate $leaf\_rate$. In an intermediate node, one simple way of estimating $\overline{L}_i$ is to calculate the moving average of $leaf\_rate$ as follows:

$$\overline{L}_i \longleftarrow \alpha\overline{L}_i + (1-\alpha)leaf\_rate, \qquad (0<\alpha<1) \tag{5}$$

where $leaf\_rate$ is the temporal sending rate of a leaf node and marked on the packet, and $\overline{L}_i$ is the average value of $leaf\_rate$. However, (5) does not produce an accurate average value of the sending rates of leaf nodes, because a leaf node with a higher sending rate sends more packets in a given time interval, and thus, the average value in (5) is biased to the higher sending rate. To compensate for different incoming rates, we differentiate the number of calculations in (5) inversely proportional to $leaf\_rate$ as follows:

$$\begin{aligned} \overline{L}_i^{(1)} &\longleftarrow \alpha\overline{L}_i + (1-\alpha)leaf\_rate \\ \overline{L}_i^{(2)} &\longleftarrow \alpha\overline{L}_i^{(1)} + (1-\alpha)leaf\_rate \\ &\vdots \\ \overline{L}_i^{(n)} &\longleftarrow \alpha\overline{L}_i^{(n-1)} + (1-\alpha)leaf\_rate \\ &\left(n = \frac{\overline{L}}{leaf\_rate>0}\right). \end{aligned} \tag{6}$$

Equation (6) is $\overline{L}/leaf\_rate$ times of compositions of the moving averaging function and can be reduced to a closed form as follows:

$$\begin{aligned} \overline{L}_i \leftarrow \alpha^n\overline{L}_i + (1-\alpha)leaf\_rate\sum_{k=0}^{n-1}\alpha^k \\ = \alpha^n\overline{L} + (1-\alpha^n)leaf\_rate. \end{aligned} \tag{7}$$

Equation (7) compensates for the difference in packet frequency between high $leaf\_rate$ nodes and low $leaf\_rate$ nodes on estimating the average sending rate $\overline{L}_i$, where α determines the window size of the moving averaging function.[2]

Once an intermediate node gets to know n_i based on (4) and (7), the weight $f_{(r,i)}$ of node i is given by

$$f_{(r,i)} = \frac{n_i}{\sum_{k\in V_{D_r}/\{r\}} n_k} \qquad i\in V_{D_r}/\{r\}. \tag{8}$$

Theorem 1: If every node allocates the capacity to its child nodes according to (8), then (3) is satisfied.

Proof: We use structural induction on a tree topology.

- *Basis*. For every leaf node i, consider its parent node r and collision domain D_r that contains r as the root node. Then, $\overline{L}_i = c_i$. Hence, based on (4) and (8), we have

$$f_{(r,i)} = \frac{c_i/c_i}{\sum_{k\in V_{D_r}/\{r\}} c_k/c_k} = \frac{1}{|V_{D_r}|-1}.$$

This equation indicates the trivial fact that all the leaf nodes that share the parent node receive the same weight.

- *Induction*. For every child node i of node r, assume that the capacities of all leaf nodes that belong to i are the same as the average value $\overline{L}_i$ (induction hypothesis). Then

$$\begin{aligned} f_{(r,i)} &= \frac{c_i/\overline{L}_i}{\sum_{k\in V_{D_r}/\{r\}} c_k/\overline{L}_k} \\ c_i &= c_r f_{(r,i)} = c_r\frac{c_i/\overline{L}_i}{\sum_{k\in V_{D_r}/\{r\}} c_k/\overline{L}_k} \end{aligned}$$

from which

$$\overline{L}_i = c_r\frac{1}{\sum_{k\in V_{D_r}/\{r\}} c_k/\overline{L}_k}.$$

The bandwidth received at the leaf nodes that belong to any collision domain D_r is the same value, independent of node i. Therefore, we conclude that

$$c_i = c_j \quad \text{for every all active leaf node } i \text{ and } j.$$

■

B. Scheduling Nodes

There exist several schemes for scheduling nodes with a given set of weights in IEEE 802.11 wireless local area network (WLAN) [9], [13]. Most of them have been developed based on the assumption that all the contending nodes are within a carrier sensing range so that, when a node transmits a packet, other nodes can sense the transmission. However, a WMN should maximize its coverage and place mesh router as far apart as possible, and mesh nodes may not be within a transmission range.

Fig. 3 illustrates two different collision models within and beyond a transmission range. The IEEE 802.11 MAC operates in a discrete time slot, and the length of each slot is usually set to 9 μs. In the model within a transmission range [see Fig. 3(a)], suppose that node 1 chooses a slot [see slot b in Fig. 3(a)] to send data packet. If another node (node 2) chooses the slot prior to slot b (slot a), then node 2 acquires the channel and can send its packet. If node 2 chooses the next slot of node 1's transmission (slot c), node 2 can detect the busy channel and postpones its transmission. The only case that collision occurs is when both nodes choose the same slot. In the model beyond a transmission range [see Fig. 3(b)], however, nodes may collide in the consecutive slots from the first slot of a request-to-send (RTS) mechanism (slot d) to the first slot of a clear-to-send (CTS) mechanism (slot e), because node 2 cannot detect the transmission of node 1 [15].

To model the collision beyond a transmission range, we extend the previous model for the collision within a transmission range in [9] and [14]. When a node experiences collision, it chooses a random backoff time between 0 and the contention window size (CW) and counts down to 0 to start transmission. Let $b(t)$ be the backoff time counter of a given node

[2] Packets can be lost while transmitted in a WMN. Then, the value of $leaf\_rate$ does not reflect the correct transmission rate. To deal with the change in transmission rate due to packet losses, each intermediate node adjusts $leaf\_rate$ to $(1-p_d)leaf\_rate$, where p_d is the loss rate of the node.

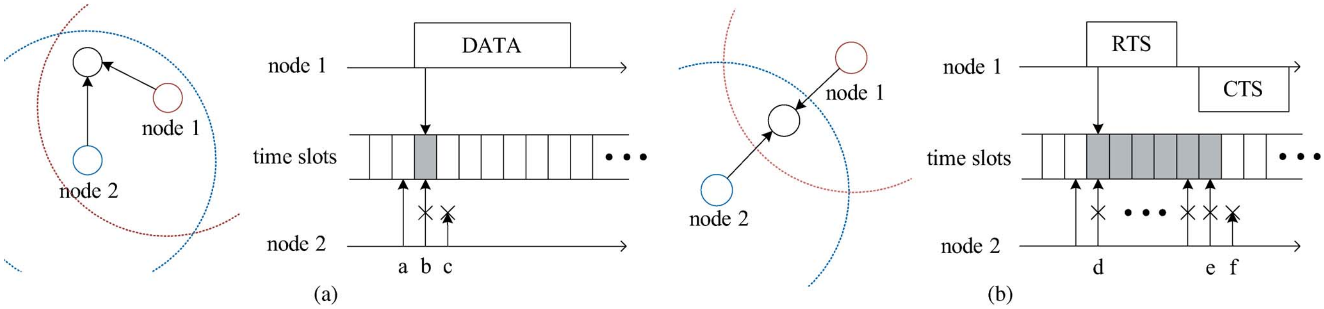


Fig. 3. Collision models within and beyond a transmission range. (a) Within a transmission range. (b) Beyond a transmission range.

at time t. Then, the stationary distribution of the chain b_k is given by

$$b_k = \lim_{t\to\infty} P\left\{b(t) = k\right\}. \tag{9}$$

At a given node, when its contention window decreases to 0, the node attempts to transmit its packet. It may succeed to transmit, or it may observe collision and pick a new number between 0 and $CW - 1$ for contention window, *where CW is a contention window size*. If the node picks a value greater than k, the contention window comes back to k, with a probability of 1. Hence, we have

$$b_k = \frac{CW - k}{CW} b_0. \tag{10}$$

Here, note that

$$\sum_{k=0}^{CW-1} b_k = b_0 + \sum_{k=1}^{CW-1} \frac{CW - k}{CW} b_0 = 1. \tag{11}$$

Based on (11), we have

$$b_0 = \frac{2}{CW + 1} \tag{12}$$

as shown in [14]. Intuitively, (12) means that a node attempts to transmit a packet every $(CW + 1)/2$ slots, on the average, because the contention window is uniformly distributed in the range $[0, CW - 1]$. We define the probability τ_i that node i attempts to transmit on given slot as

$$\tau_i = b_0 = \frac{2}{CW_i + 1}. \tag{13}$$

Then, the probability that node i successfully transmits on a given slot $\mathcal{T}_i$ is

$$\mathcal{T}_i = \tau_i \prod_{\forall j\neq i} (1 - \tau_j)^s \tag{14}$$

where s is the number of consecutive slots that are vulnerable to collision, as shown in Fig. 3(b). Equation (14) represents the situation in which node i transmits and that all the other nodes are silent during s number of consecutive slots. Finally, to schedule the nodes to satisfy their weights f_i, we define the following constraint:

$$\frac{\mathcal{T}_i}{f_i} = \frac{\mathcal{T}_j}{f_j} \Longleftrightarrow \frac{\tau_i}{f_i(1 - \tau_i)^s} = \frac{\tau_j}{f_j(1 - \tau_j)^s}. \tag{15}$$

Equation (15) makes the probability of successful transmission proportional to the weights of all the child nodes i, j in a collision domain.

Because (15) gives us relative values of contention windows, there is an issue of base contention window selection to maximize the channel utilization. It is well known that the optimal base contention window size is a function of the number of contending nodes, and we can approximate by linear function $CW_1 = \rho|V_D|$, where V_D is a set of nodes in a collision domain D [9]. ρ is a function of collision length, and we can determine based on the parameters in the IEEE 802.11 standard [19].

Without loss of generality, we assume that f_1 is 1 and rewrite (13) and (15) as

$$f(CW_i) = \frac{CW_1 + 1}{2\left(1 - \frac{2}{CW_1+1}\right)^s} - \frac{CW_i + 1}{2f_i\left(1 - \frac{2}{CW_i+1}\right)^s} = 0. \tag{16}$$

Based on (16), we can find a proper CW_i for node i to meet its f_i. However, (16) does not yield a closed-form solution, because the exponent s is greater than 3. According to the parameters in the IEEE 802.11b standard, s is evaluated as more than 17, including the preamble, RTS header, and *short interframe space* (SIFS) [15]. To solve it, instead, we can use numerical approximation such as the Newton method. Let n's iteration of CW_i be $CW_i^{(n)}$, and we approximate CW_i using the following procedure:

$$CW_i^{(n+1)} = CW_i^{(n)} - \frac{f\left(CW_i^{(n)}\right)}{f'\left(CW_i^{(n)}\right)} \tag{17}$$

where $f'(CW_i)$ is the derivative of f to CW_i.

To implement the numerical method at mesh routers, we need to analyze the stability of the numerical solution and the complexity of the method. We can rewrite the target equations (13) and (15) as

$$\gamma = \frac{\frac{2}{CW_i+1}}{f_i\left(1 - \frac{2}{CW_i+1}\right)^s} \tag{18}$$

where γ is a constant value and means the probability of successful transmission of a base node, with one as its weight. For positive CW_i, the right side of (18) is a positive monotonic decreasing function, and γ is a positive value. Therefore, we can prove that there exists a unique solution by the intermediate

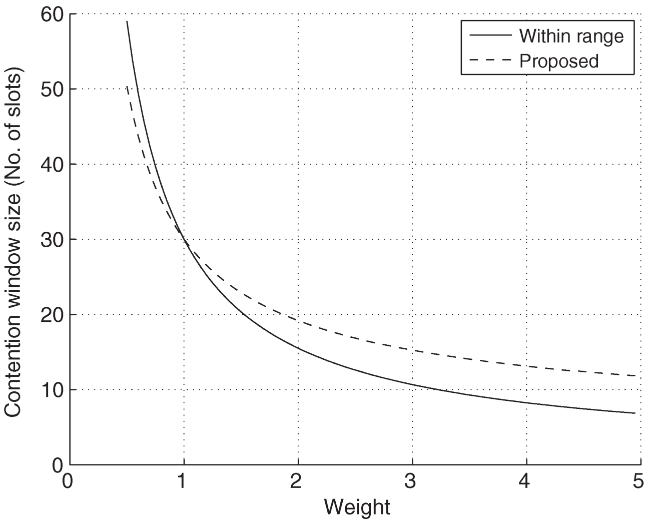


Fig. 4. Contention window size according to the weighted scheduling model when the base contention window is 30.

value theorem, and we can obtain the solution from (18) using a stable numerical method such as the bisection method. It is not easy to directly analyze the complexity of the approximation method, but we may derive the upper bound for it using a simple binary search algorithm. If we use a trivial binary search algorithm to find the solution, when the maximum contention window size is 1024 slots, the maximum iteration number to converge to an integer solution is $\log_2 1024 = 10$. In practice, we can optimize the algorithm using the Newton method, which more quickly converges than the binary search. Thus, we can expect that the overhead of the approximation algorithm is not significant. Throughout our simulation, it is observed that the average iteration count of the approximation method is 3.85, and the maximum value is 8.

Based on the result of the numerical method, we present Fig. 4 to show the contention windows size functions with respect to their weights. In the figure, the solid line is the contention window size, without considering the out-of-range collision proposed in [9] [see Fig. 3(a)], and the dashed line is the contention window of our proposed scheme based on (12) and (14) [see Fig. 3(b)]. If nodes are placed within a carrier-sensing range, according to [9], the contention window size of a node is inversely proportional to its weight as a function of

$$CW_i = \frac{CW_1 - 1}{f_i} + 1. \tag{19}$$

It is easy to calculate, but we cannot directly apply this approach to our WMN. In the case that nodes are placed beyond a carrier-sensing range, the contention window size is not exactly inversely proportional to the weight of a node. Fig. 4 shows that the slopes of the curves are more gentle than (19).

The scheduling method that we have discussed so far is derived based on the assumption that the packet length in a WMN is constant. Now, we consider a more general WMN with variable packet lengths. In such a WMN, fairness in the channel access opportunity does not mean fairness in the throughput. For example, let the packet size of the node i be p and the packet size of node j be $2p$. Then, with the same contention window size, node j achieves twice as much throughput as node i, because node j transmits twice as much data for one transmission. Therefore, each node in a WMN has to adjust its CW based on its packet length. Let a base packet size in a WMN be P. If the packet size to transmit for node i is P_i, the node i should adjust its transmission opportunity $\mathcal{T}_i$ to $(P/P_i)\mathcal{T}_i$. Then, the node i uses the adjusted CW_i' from the following equation, instead of CW_i, for the transmission:

$$\frac{CW_i + 1}{2\left(1 - \frac{2}{CW_i+1}\right)^s} - \frac{CW_i' + 1}{2\frac{P}{P_i}\left(1 - \frac{2}{CW_i'+1}\right)^s} = 0. \tag{20}$$

Equation (20) has the same structure as (15), which means that every node instantly adjust its weights according to its packet size. The solution of (20) can also be obtained with numerical methods.

Algorithm 1 shows a pseudocode of the algorithm that we have discussed. Every intermediate node estimates node weights, calculates the contention window of child nodes, and periodically notifies them of weights through one-hop control messages. new_M is the function discussed in (7).

```
Algorithm 1. Estimate weight and schedule child nodes
//update $\overline{L}_i$ and $c_i$ when packet received
for all incoming packet $p$ from child node $i$ do
    $\overline{L}_i \Leftarrow new\_M(\overline{L}_i, p.leaf\_rate)$
    $c_i \Leftarrow (c_i avg\_interval + p.size)/(interval + avg\_interval)$
end for
//on every interval parent node updates weight of child nodes
for all child node $i$ do
    $CW_i \Leftarrow$ numerical solution of $(13)$ and $(15)$
    notify child node $i$ of new $CW_i$
end for
```

V. Performance Evaluation

In this section, we present the performance evaluation for the proposed scheme. To evaluate our scheme, we conduct extensive simulation on the *ns-2* simulator [17]. We modify three parts of the *ns-2* simulator: 1) the IEEE 802.11 MAC protocol; 2) the interface queue; and 3) the multi-interface support for mobile nodes. The transmission range is 200 m, and the interference range is twice the transmission range. The channel propagation model is a two-ray ground reflection model. The parameters for simulation are set based on the IEEE 802.11g standard [15]. We use 54 Mb/s as the channel bit rate and 6 Mb/s as the physical-layer convergence protocol (PLCP) data rate. The time slot is 9 μs, and the CW min and CW max are 15 and 1023 slots, respectively.

Throughout our simulation, we use three topologies, as depicted in Fig. 5. A triangle vertex is the gateway node connected to the wired Internet with a 100-Mb/s link, and white circles represent leaf nodes connected to client nodes. Black circles are intermediate nodes that relay packets from leaf nodes.

The following assumptions are made in our simulation. First, every topology has one gateway node, and all traffic flows from

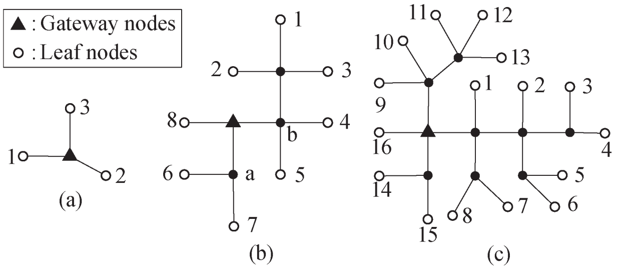


Fig. 5. Simulation topologies. (a) Four nodes (three leaf nodes). (b) Twelve nodes (eight leaf nodes). (c) Twenty-four nodes (16 leaf nodes).

leaf nodes to the gateway node. We assume that a topology with multiple gateways can be divided into several tree topologies with a single gateway. Second, mesh routers, including leaf and intermediate nodes, have two network interfaces that operate on orthogonal channels: 1) one interface for uplink and 2) another interface for downlink. With proper channel assignment, interference that is more than two hops does not exist. Third, the transmission range of a physical channel is 200 m, and mesh nodes are well placed to cover the maximum area without loss of connectivity. Fourth, mesh nodes are stationary, and the channel assignment does not change over time. Hence, we use static routing path and static channel assignment.

A. Weighted Scheduling

We have discussed two different scheduling schemes in Section IV-B. The first scheduling scheme is the distributed scheduling algorithm called *P-MAC*, which assumes that nodes are within a single shared transmission range [9]. The second scheduling scheme is our proposed scheme based on (13)–(15), which considers the collision model described in Fig. 3(b).

To evaluate the scheduling schemes, we conduct a simulation on *ns-2*. Based on our simulator parameters, we can induce s of (14). Because an RTS packet is sent at the rate of a *PLCP* data *rate*, s is calculated by

$$s = \frac{\frac{RTS\ length}{PLCP\ data\ rate} + SIFS}{time\ slot} = 7.259. \tag{21}$$

Note that s is an integer number of consecutive slots vulnerable to collision. Thus, we use 7 for s in our simulation.

For this simulation, we use a simple WMN topology with four nodes, as illustrated in Fig. 5(a). We assign a weight of one to nodes 1 and 2, and we change the weight of node 3 from two to five. There are five TCP flows from each leaf node toward the gateway node, and we measure the average throughput of these five flows. The duration of simulation is 100 s. Fig. 6 shows the normalized throughput of node 3 based on the average throughput of other nodes. It is observed that the previous differentiation scheme (P-MAC) tends to overfavor higher weight nodes, as illustrated in Fig. 4. It is shown that our scheme more accurately realizes the target throughput than P-MAC.

B. Node Weight Estimation

To estimate the number of active leaf nodes that belong to a node, we make some minimal modification to the existing IEEE 802.11 MAC. We implement the algorithm described in **Algorithm 1** on the *ns-2* simulator. In this section, we use topology (b) in Fig. 5 to evaluate the weight-estimation scheme. This topology consists of eight leaf nodes, three intermediate nodes, and one gateway node.

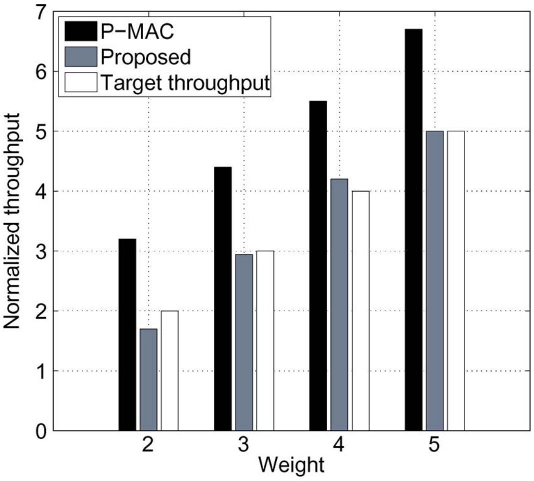


Fig. 6. Achieved throughput with various weights of node 3.

Fig. 7 depicts the weight estimation at the gateway node, the contention window size for the weighted scheduling, and the achieved throughput of leaf nodes. Initially, we inject five TCP flows to leaf nodes 4, 6, and 8. Then, we observe that the gateway node estimates the number of active nodes as one for all its child nodes (intermediate nodes a and b and leaf node 8) in Fig. 7(a). At 20 s, we start flows from leaf node 5 and observe that the estimated number of active nodes from node b becomes two. At 40 s, node 1 starts to send traffic, and the gateway node accurately detects it. At 60 s, node 5 stops sending, and simultaneously, node 7 starts to send traffic. Then, the estimated number of active leaf nodes from node b comes back to two, and the estimated number from node a increases by one. Throughout this result, we confirm that the proposed scheme for estimating the weight is quite accurate, although minor variations are observed in transient states due to the slow response of the rate estimator.

Based on this estimated weight of child nodes, the contention window for each child node is calculated to achieve fair bandwidth allocation, as shown in Fig. 7(b). The child nodes of the gateway node start with the same contention window, because they have the same number of active leaf nodes. At 20 s, the number of active leaf nodes that belong to node b becomes two, and the contention window for other child nodes increases so that node b can have more chances of sending its packets. At 40 s, the weight of node b again increases by one, and other child nodes increase their contention window more. At 60 s, the weight of nodes a and b become the same as 2, and the contention windows for them are also the same. As a result of this adaptation, it is observed in Fig. 7(c) that the achieved throughput of each leaf node is maintained to be the same, regardless of dynamic changes in the number of active nodes.

C. Fair Bandwidth Allocation

Now, we look at the throughput fairness of the proposed scheme. In topologies (b) and (c) in Fig. 5, we inject five TCP

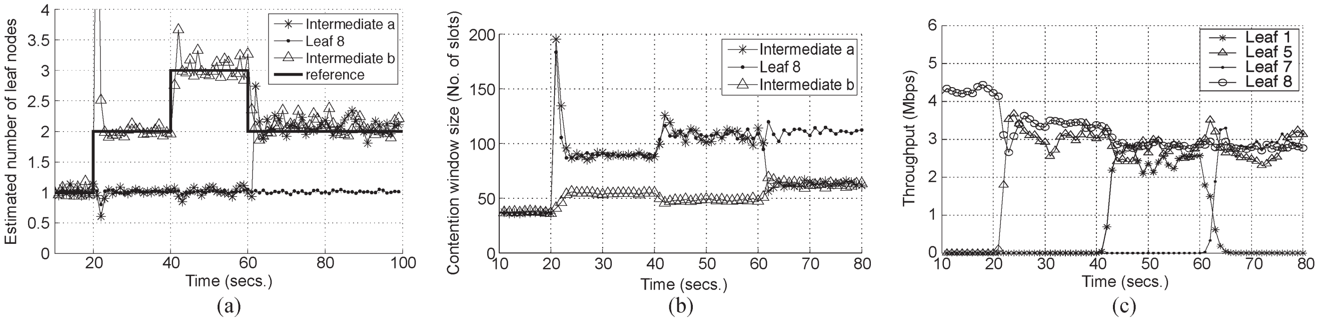


Fig. 7. Weight estimation and bandwidth sharing with dynamic join and leave of nodes in topology (b). (a) Estimated weight at the gateway node. (b) Contention window of child nodes. (c) Achieved throughput at leaf nodes.

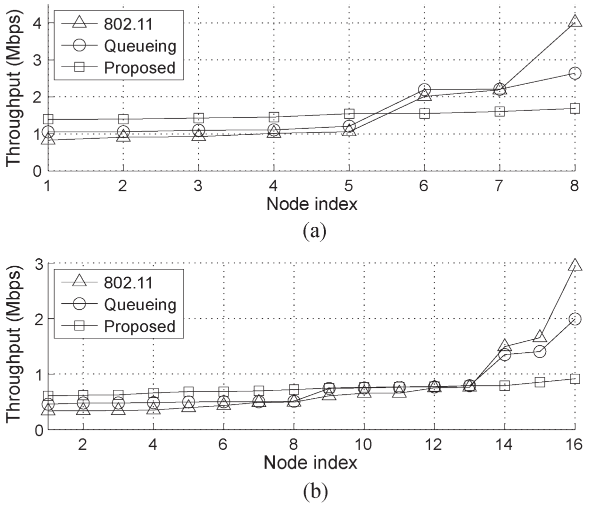


Fig. 8. TCP throughput of leaf nodes using IEEE 802.11, per-node queuing, and the proposed scheme. (a) TCP throughput at each leaf node in topology (b). (b) TCP throughput at each leaf node in topology (c).

flows to each leaf node. The simulation duration is 100 s. The average throughput achieved by leaf nodes is shown in Fig. 8.

In Fig. 8(a), the curve labeled as 802.11 is the throughput of each node using the current IEEE 802.11 standard, *Queuing* is the throughput of the per-node queuing scheme described in [6], and *Proposed* is the throughput of our proposed scheme. In IEEE 802.11, the following instances are observed: 1) Leaf nodes that belong to node b (node 1–5) achieve much less throughput than others, because five nodes share an intermediate node (node b); 2) nodes 6 and 7 achieve half the throughput of node 8, because two nodes share node a; and 3) node 8 achieves the highest throughput. The proposed scheme effectively eliminates unfair sharing and realizes fair bandwidth sharing. We can also observe that the queuing scheme cannot resolve the unfairness bandwidth problem, because the gateway node has no backlogged packets. This result can be also confirmed with a more complex topology [topology (c) in Fig. 8(b)].

We compare the result in terms of the *fairness index* [18]. A *fairness index* close to 1 indicates that bandwidth sharing is fair. The fairness index of the IEEE 802.11 standard, the queuing scheme, and our proposed scheme is 0.5832, 0.7716, and 0.9873, respectively. It is also confirmed that the proposed scheme achieves high-level throughput fairness.

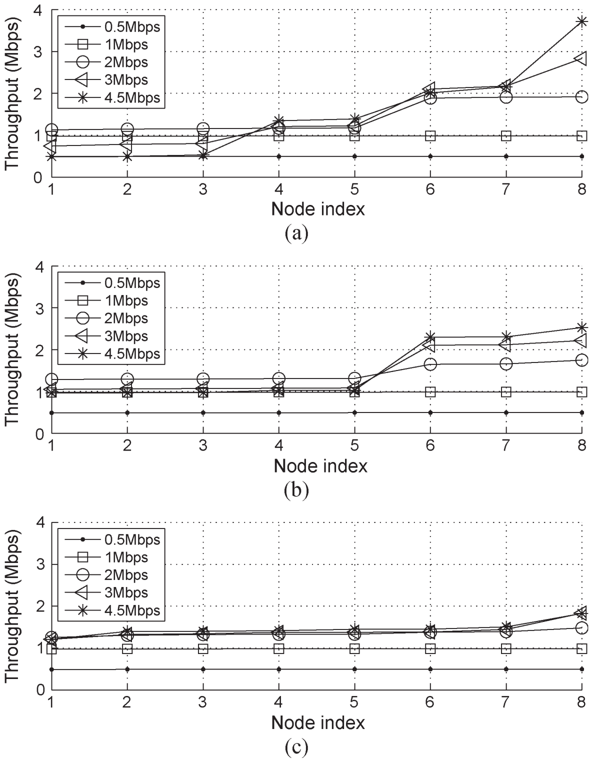


Fig. 9. Throughput of overinjected UDP traffic with IEEE 802.11 and the proposed scheme. (a) IEEE 802.11. (b) Per-node queuing scheme. (c) Proposed scheme.

User Datagram Protocol (UDP) traffic is unresponsive and does not respond to network congestion. When UDP traffic is overinjected to a network, the network suffers from a severe unfairness problem. Fig. 9 shows an unfair situation due to overinjected UDP traffic. In topology (b) in Fig. 5, we inject UDP traffic to all the leaf nodes with the same sending rate and increase the sending rate from 0.5 to 4.5 Mb/s. When the total amount of UDP traffic is less than the network capacity (rate less than 2 Mb/s), all schemes show similar results. As the sending rate increases, however, the nodes near the gateway node monopolize the channel resource in IEEE 802.11 [see Fig. 9(a)]. Fig. 9(b) also shows that the per-node queuing scheme cannot resolve the underlying MAC layer contention. In our proposed scheme, it is shown that the bandwidth achieved by each node converges to a fair share, regardless of the sending rate, and we can confirm that the proposed scheme also effectively deals with unresponsive traffic.

Through all these simulations, the aggregate throughput of our scheme maintains higher than 97% of the IEEE 802.11 standard, which is not a critical degradation, considering the realized throughput fairness.

VI. Conclusion

In this paper, we have proposed a scheme for realizing throughput fairness in a multihop WMN. The proposed scheme is highly scalable, because it is fully distributed and does not require any global information. The proposed scheme consists of two procedures: 1) distributed weight estimation and 2) channel scheduling. The weight estimation algorithm effectively estimates the number of active nodes using the sending rate of each node. The channel scheduling algorithm differentiates the contention window size of each child node. For accurate throughput sharing, we analyzed the collision model that nodes are beyond a transmission range. Through extensive simulations, we have observed that a significant improvement in throughput fairness can be realized by the proposed scheme.

References

[1] J. Lee and I. Yeom, “Achieving throughput fairness in wireless mesh networks based on IEEE 802.11,” in *Proc. IEEE MASS*, Atlanta, GA, Oct. 2008, pp. 507–508.

[2] I. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: A survey,” *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, Mar. 2005.

[3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, “Architecture and evaluation of an unplanned 802.11b mesh network,” in *Proc. MobiCom*, Cologne, Germany, Aug. 2005, pp. 31–42.

[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” in *Proc. ACM SIGCOMM*, London, U.K., Aug. 1994, pp. 212–225.

[5] V. Gambiroza, B. Sadeghi, and E. Knightly, “End-to-end performance and fairness in multihop wireless backhaul networks,” in *Proc. MobiCom*, Philadelphia, PA, Sep. 2004, pp. 287–301.

[6] N. Nandiraju, D. S. Nandiraju, D. Cavalcanti, and D. P. Agrawal, “A novel queue management mechanism for improving performance of multihop flows in IEEE-802.11s-based mesh networks,” in *Proc. IPCCC*, Phoenix, AZ, Apr. 2006, pp. 161–168.

[7] A. Brzezinski, G. Zussman, and E. Modiano, “Enabling distributed throughput maximization in wireless mesh networks—A partitioning approach,” in *Proc. MobiCom*, Los Angeles, CA, Sep. 2006, pp. 26–37.

[8] A. Raniwala and T. Chieh, “Architecture and algorithms for an IEEE-802.11-based multichannel wireless mesh network,” in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 2223–2234.

[9] D. Qiao and K. Shin, “Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF,” in *Proc. 10th IEEE Int. Workshop Quality Service*, Miami Beach, FL, May 2002, pp. 227–236.

[10] E. Park, D.-Y. Kim, C.-H. Choi, and J. So, “Improving quality of service and assuring fairness in WLAN access network,” *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 337–350, Apr. 2007.

[11] J. Jun and M. Sichitiu, “Fairness and QoS in multihop wireless network,” in *Proc. VTC—Fall*, Orlando, FL, Oct. 2003, vol. 5, pp. 2936–2940.

[12] B. Raman, “Channel allocation in 802.11-based mesh networks,” in *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–10.

[13] N. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless LAN,” in *Proc. MobiCom*, Boston, MA, Aug. 2000, pp. 167–178.

[14] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[15] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, P802.11, Nov. 1997.

[16] G. Tan and J. Guttag, “Time-based fairness improves performance in multirate wireless LANs,” in *Proc. USENIX Annu. Tech. Conf.*, Boston, MA, Jun. 2004, p. 23.

[17] *The Network Simulator—ns-2*. [Online]. Available: http://www.isi.edu/nsnam/ns/

[18] R. Jain, G. Babic, B. Nagendra, and C. Lam, “Fairness, call establishment latency, and other performance metrics,” ATM_Forum/96-1173, ATM Forum Document, Aug. 1996.

[19] D. Qiao and S. Choi, “Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation,” in *Proc. IEEE ICC*, Helsinki, Finland, Jun. 2001, vol. 7, pp. 1995–2000.

[20] A. Giannoulis, T. Salonidis, and E. Knightly, “Congestion control and channel assignment in multiradio wireless mesh networks,” in *Proc. IEEE SECON*, San Francisco, CA, Jun. 2008, pp. 350–358.

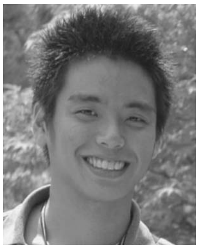
[21] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. W. Knightly, “Measurement and modeling of the origins of starvation in congestion controlled mesh networks,” in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1633–1641.

[22] J. Deng, B. Liang, and P. Varshney, “Tuning the carrier sensing range of IEEE 802.11 MAC,” in *Proc. IEEE Globecom*, Dallas, TX, Nov. 2004, vol. 5, pp. 2987–2991.

[23] K. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, “Interference-aware channel assignment in multiradio wireless mesh networks,” in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[24] J. So and N. Vaidya, “Multichannel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver,” in *Proc. MobiHoc*, Tokyo, Japan, May 2004, pp. 222–233.

[25] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services; RFC-2475,” Internet Request for Comments, vol. 2475, Dec. 1998.



Janghwan Lee (M’10) received the B.S. degree in computer science in 2004 from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, where he is currently working toward the Ph.D. degree with the Department of Computer Science.

His research interests include mobile ad hoc networks and wireless mesh networks.

Hyunsoo Yoon (M’08) received the B.S. degree from Seoul National University, Seoul, Korea, in 1979, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1981, and the Ph.D. degree from The Ohio State University, Columbus, in 1988.

He is currently a Professor of with the Department of Computer Science, KAIST. His research interests include mobile ad hoc networking, wireless sensor networking, and fourth-generation mobile communication networks.

Ikjun Yeom (M’02) received the B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in February 1995 and the M.S. and Ph.D. degrees in computer engineering from Texas A&M University, College Station, in August 1998 and May 2001, respectively.

From January 2002 to August 2008, he was an Assistant Professor with the Department of Computer Science Division Korea Advanced Institute of Science and Technology, Daejeon, Korea. He is currently an Associate Professor with the Department of Computer Engineering, Sungkyunkwan University, Suwon, Korea. His research interests include quality of service, Transmission Control Protocols, and mobile ad hoc networking.