

# Throughput Optimization and Fair Bandwidth Allocation in Multi-hop Wireless LANs

Qunfeng Dong

Department of Computer Sciences  
University of Wisconsin-Madison  
Madison, WI 53706, USA  
Email: qunfeng@cs.wisc.edu

Suman Banerjee

Department of Computer Sciences  
University of Wisconsin-Madison  
Madison, WI 53706, USA  
Email: suman@cs.wisc.edu

Benyuan Liu

Department of Computer Science  
University of Massachusetts at Lowell  
Lowell, MA 01854, USA  
Email: bliu@cs.uml.edu

**Abstract**—There is an inherent well-known conflict between fairness and throughput that arises in many networking scenarios. A number of researchers have studied this problem in the context of (single-hop) wireless local area networks (WLANs), where clients directly exchange traffic with access points (APs). More recently, researchers have proposed multi-hop extensions to WLANs where client traffic is forwarded via a series of client-client links. In this paper, we show that the objective of improving throughput without sacrificing fairness can be much better met in multi-hop WLANs. We decouple this objective into two separate but related problems. First, we need an algorithm to organize clients into a multi-hop structure such that fair bandwidth allocation within this structure leads to improved throughput. Second, we need algorithms for performing fair bandwidth allocation within the determined multi-hop structure. In this paper, we first design optimal fair bandwidth allocation algorithms for both max-min throughput fairness and max-min time fairness in multi-hop WLANs. Subsequently, design an efficient algorithm to find desirable multi-hop structures. With slight modification, our results in this paper can be generalized to other multi-hop wireless networks, such as the emerging wireless backhaul networks and wireless mesh networks. Our proposed solutions seamlessly integrate with legacy devices and hence are incrementally deployable. Simulation results demonstrate that our solutions can effectively improve throughput (by up to 114% or more) as well as network coverage while preserving fairness.

## I. INTRODUCTION

Wireless local area networks (WLANs) have mushroomed at *hotspots* like office buildings, libraries, coffee shops, airports, hotels, etc. In a typical deployment, a *client* equipped with an 802.11 interface communicates over the air to an *access point* (AP) or base station that is connected to a wired infrastructure. Unfortunately, wireless communication (between clients and APs) is very susceptible to signal quality degradation caused by fading, noise, interference, multi-path reflection, attenuation, and user mobility, etc. When the average received signal strength at the receiver is consistently below the threshold required for successful packet reception, the receiver experiences packet losses. To communicate more reliably, the sender can transmit at a lower *bit rate* (using a more resilient modulation scheme) so that the channel bit error rate (BER) is reduced. For example, 802.11b provides 4 bit rates (1Mbps, 2Mbps, 5.5Mbps, 11Mbps) and 802.11a provides 8 bit rates ranging from 6Mbps to 54Mbps. Many 802.11 vendors have implemented automatic scheme for such bit rate control [1]–

[3]. Previous studies [4], [5] have reported that rate diversity is prevalent in many in-door WLANs and exists even in a small room, which are most appropriate for hotspots.

Previous studies of corporate WLANs [6] and campus-wide wireless networks [7] have shown that WLANs often carry significant traffic and contain many APs that have a lot of busy or congested periods. When multiple clients contend for access to the same wireless channel, a channel allocation scheme is needed to distribute channel access time among competing clients, according to some fairness policy. The Distributed Coordination Function (DCF) MAC protocol used by 802.11 tends to give equal long-term channel access opportunities to all competing clients [8], [9]. In particular, each node has (approximately) the same number of opportunities to transmit a data frame, regardless of its bit rate and hence the amount of channel access time required to transmit a data frame. If clients transmit packets of similar sizes and experience similar packet loss rates, they achieve approximately the same throughput irrespective with their bit rates. This is referred to as *throughput-based fairness* [5]. Consequently, aggregate throughput and throughput of high bit rate clients may be dramatically brought down, because low bit rate clients will occupy more channel access time to transmit an equal amount of data. Such “performance anomaly” of 802.11 WLANs has been reported by Heusse *et al.* in [10].

DCF mainly affects the channel capacity allocation in the uplink direction. The packet scheduling mechanism at the AP dictates channel capacity allocation in the downlink direction. When there are multiple backlogged packets destined to multiple clients, the scheduling scheme must decide the order of transmission. Again, since the channel conditions at the clients vary, different bit rates are often used for different clients. Scheduling schemes in the literature provide throughput-based fairness that has been widely accepted in wireline networks and single-rate 802.11 WLAN formulations [11]–[13]. Like other existing scheduling algorithms in wireless networks [14]–[16], the scheduling scheme employed by APs of multi-rate WLANs also adopt this notion of fairness. Therefore, channel capacity allocation in the downlink direction is impacted in a similar undesirable way as in the uplink direction.

These inefficiencies lead to a number of potential problems. For example, in 54Mbps 802.11g networks that are deployed

alongside relatively slower 802.11b networks, 802.11g users may see far less performance improvement than expected and thus hesitate on upgrading to 802.11g. In order to address these inefficiencies, Tan and Gutttag [5] proposed *time-based fairness*, where each client is assigned an equal amount of channel access time, regardless of its bit rate. Clearly, time-based fairness protects high bit rate clients from drastic throughput degradation by reserving a fixed share of channel access time for them. However, this disadvantages low bit rate clients. In many cases, this is considered undesirable, too.

A key problem with throughput-based fairness and time-based fairness is that so far they are both constrained to maneuver with client-AP links, some of which may have a low bit rate. We refer to such WLANs as single-hop WLANs. Furthermore, in currently deployed WLANs where there are more than one AP, clients are typically restricted to use the link between themselves and the AP that has the strongest received signal strength indicator (RSSI). Bejerano *et al.* [17] propose to relax these constraints by allowing clients to carefully re-associate with others APs so that workload is more balanced among APs. However, although such load balancing techniques can effectively improve fairness, they are still restricted to use client-AP links. Therefore, achievable throughput improvement is limited.

To improve throughput, some recent work [18] made the observation that peer-to-peer links between nearby clients often possess much higher bit rates. By forwarding client traffic via high quality client-client links, multi-hop WLANs [18] have the potential to significantly improve client throughput. However, ad hoc mechanisms to choose multi-hop path can often hurt performance. In this paper, we address the problem of appropriate choice of multi-hop paths in multi-hop WLAN scenarios. We show that carefully choosing multi-hop paths can significantly improve client throughput without sacrificing fairness. To achieve this objective, two relevant problems need to be addressed. First, we need an algorithm to organize clients into a multi-hop structure such that fair bandwidth allocation within this structure leads to improved throughput. Second, we need algorithms to perform fair bandwidth allocation within the determined structure.

Recently, Gambiroza *et al.* [25] attempt to address this second problem in the context of wireless backhaul networks, where the network topology is relatively static. While they formulate the problem for arbitrary link contention graphs, in their work the authors only provide an optimal solution for bandwidth assignment in the special case where all the links interfere with each other, i.e., the link contention graph is a clique. However, in a typical WLAN deployment (e.g. office buildings, etc.), not all APs and clients are in direct interference range of each other and hence the link contention graph is not necessarily a clique. Optimal solution in the general case is more difficult.

In contrast, we address both of the two relevant problems in the general setting, and we also examine how our solutions should adapt to dynamic changes of network topology. Such changes can be fairly common in WLAN scenarios. In

particular, we make the following key contributions.

#### *Our contributions*

- We consider both max-min throughput fairness and max-min time fairness in multi-hop WLANs. For each fairness policy, we define and analyze an *optimal* algorithm for fair bandwidth allocation within a multi-hop WLAN.
- We design an efficient algorithm to smoothly improve the structure of WLANs by constantly searching for a better structure than the current one. Simulation results demonstrate that our solutions can effectively improve throughput (by up to 114% or more) as well as network coverage while preserving fairness.
- Our solutions seamlessly integrate legacy client devices (that do not implement our algorithms) with smart client devices (that implement our algorithms) and hence allows a smooth transition from legacy technology to smart technology through incremental deployment. Legacy client devices are directly associated with APs as usual and need not participate our algorithms to form a multi-hop WLAN. Such incremental deployment is not only more feasible than global upgrade, but also better motivated by our solutions, which reward individual clients investing to upgrade their device with much more perceivable performance improvement. These properties make our solutions practically interesting ones.
- With slight modification, our results can be generalized to other multi-hop wireless networks, such as the emerging wireless backhaul networks and wireless mesh networks.

#### *Roadmap*

The rest of the paper is organized as follows. We present relevant models, definitions, notations, and formulations in Section II. Optimal bandwidth allocation algorithms for individual fairness policies are presented in Section III and IV, respectively. We then address a number of practical problems with extensions to these algorithms in Section V. Our algorithm for improving network structure is presented in Section VI. After presenting simulation results in Section VII and reviewing related work in Section VIII, we conclude the paper in Section IX.

## II. FORMULATION

In our network model, a WLAN is represented by a graph  $G = (V, E)$ , where  $V$  is the set of nodes representing APs and clients, and  $E$  is the set of edges representing communication links between nodes. For each node  $i$ , let  $b_i$  denote the bandwidth assigned to node  $i$ . We define the *bandwidth vector*,  $\vec{B} = (b_1, b_2, \dots, b_n)$ , as the clients' bandwidths sorted in non-decreasing order. For simplicity, we assume that clients are named according to this increasing order of bandwidth.

The link layer model we use in this paper is similar to that of Bejerano *et al.* [17]. Link layer details (such as bit rates, packet loss rates, channel access contention, interference, etc) are implicitly incorporated in a link quality parameter that is called *effective bit rate (EBR)*, which is measured by clients/APs

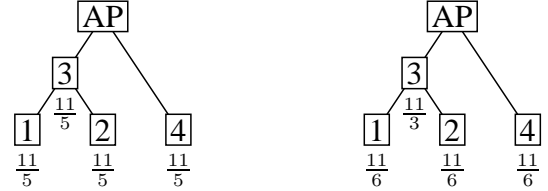
and utilized by our algorithms. For example, assume the link between client 4 and the AP in Figure 1(a) has a bit rate of 11Mbps and a packet loss rate of 20%. Assume that due to channel contention with concurrent traffic over the link between client 2 and client 3, even if client 4 and the AP are both dedicated to the conversation between them, the link between them can only be active for 75% of the time. Without further considering other factors, the EBR of the link between client 1 and the AP is thus  $11 \times (1 - 20\%) \times 75\% = 6.6\text{Mbps}$ . Some other factors may also affect the EBR of a link. For example, congestion window size has impact on the short term EBR of a link. In our model, link EBR is measured by clients and APs to reflect a reasonably long term average EBR.

Intuitively, the EBR of a link represents the amount of traffic that can be successfully delivered along the link in unit time, if both ends of the link are dedicated to the conversation. Note that in the example in Figure 1(a), EBR of the link between client 4 and the AP does not incorporate channel contention with the link between client 3 and the AP. Because in the case of that contention the AP is not dedicated to the conversation with client 4. Client 4 receives less bandwidth not because of decreased link EBR, but because of decreased amount of time of the AP that is allocated to the conversation between them. For each pair of node  $i$  and node  $j$ , we use  $r_{i,j}$  to denote the EBR of the link between them.

In this paper, we focus on the class of multi-hop WLAN structures where clients are organized into multi-hop trees each rooted at an AP. Such tree structures have been widely adopted by researchers for its simplicity and ease of management. Within a tree, each node  $i$  associates with a single parent node, denoted by  $P_i$ . Node  $i$  also has a set of child nodes, denoted by  $Q_i$ . For ease of presentation, we define  $Q_i^+ = Q_i \cup \{i\}$  and denote the subtree rooted at node  $i$  by  $T_i$ .  $|T_i|$  denotes the number of *client nodes* in  $T_i$ . The aggregate bandwidth assigned to nodes in  $T_i$  is denoted by  $B_i$ . The root node of a tree is an AP.

For ease of explanation, in the sequel we will focus on the uplink direction. The downlink direction similarly follows our discussion of the uplink direction. In the uplink direction of a tree rooted at an AP, every client needs to spend (wireless communication) time on receiving traffic from its children and on forwarding traffic for its children. The AP is a sink that does not generate its own traffic. Since it is the sink, the AP does not need to spend time on forwarding traffic that is received from its children. For simplicity, we define for each node  $i$  an indicator variable  $X_i$  such that  $X_i = 0$  if node  $i$  is an AP and  $X_i = 1$  if node  $i$  is a client. For each child node  $j \in Q_i$  of node  $i$ , the fraction of node  $i$ 's time needed to receive traffic from node  $j$  at rate  $b_j$  is  $b_j/r_{i,j}$ , and the fraction of node  $i$ 's time needed to forward traffic from node  $i$  to node  $P_i$  (if any) at rate  $b_j$  is  $X_i b_j/r_{i,P_i}$ . The fraction of node  $i$ 's time needed to transmit its own traffic (if any) to node  $P_i$  at rate  $b_i$  is  $X_i b_i/r_{i,P_i}$ . Clearly, a bandwidth allocation (or bandwidth vector)  $\mathcal{B}$  is feasible at node  $i$  if and only if the workload on node  $i$  requires no more time than node  $i$  actually has. Namely,

$b_i$	bandwidth assigned to node $i$
$P_i$	parent node of node $i$ in the multi-hop tree topology
$Q_i$	the set of child nodes of node $i$ in the multi-hop tree topology
$Q_i^+$	$Q_i^+ = Q_i \cup \{i\}$
$T_i$	the subtree rooted at node $i$ in the multi-hop tree topology
$ T_i $	the number of clients in the subtree rooted at node $i$
$B_i$	aggregate bandwidth assigned to nodes in $T_i$
$r_{i,j}$	EBR of the link between node $i$ and node $j$
$t_j^i$	the amount of node $i$ 's time needed by $B_j$ averaged over $T_j$
$X_i$	$X_i = 1$ if node $i$ is a client; $X_i = 0$ if node $i$ is an AP



(a) max-min throughput fairness (b) max-min time fairness

Fig. 1. Example of max-min throughput fairness in a multi-hop WLAN. Numbers represent the bandwidth assigned to individual nodes.

$\frac{X_i \cdot b_i}{r_{i,P_i}} + \sum_{j \in Q_i} \left( \frac{b_j}{r_{i,j}} + \frac{X_i \cdot b_j}{r_{i,P_i}} \right) \leq 1$ . A bandwidth allocation  $\mathcal{B}$  is feasible if and only if it is feasible at every node.

For ease of reading, a list of notations is given in Table I.

#### A. Max-min throughput fairness

We now examine the fairness policies we study in this paper. As an extension of throughput-based fairness in single-hop WLANs that has been introduced in Section I, we consider the more general *max-min throughput fairness* [19] in multi-hop WLANs. Informally, a feasible bandwidth allocation is max-min throughput fair if and only if it is not possible to give any user more bandwidth without decreasing the bandwidth of some user with equal or already less bandwidth. Formally, max-min throughput fairness can be defined as follows.

*Definition 1 (Max-Min Throughput Fairness):* A feasible bandwidth allocation  $\mathcal{B}$  is max-min throughput fair if and only if its corresponding bandwidth vector  $\vec{B} = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_n)$  has the same or higher lexicographical value than the bandwidth vector of any other feasible bandwidth allocation.

In single-hop WLANs, clients associated with the same AP should receive equal bandwidth under max-min throughput fairness, which is consistent with throughput-based fairness. In multi-AP and multi-hop networks, max-min throughput fairness is better than throughput-based fairness in that it allows some clients to receive more bandwidth than other clients, if the latter are not able to consume more bandwidth. Thus, unnecessary waste of idle bandwidth can be avoided. This can be demonstrated by the example in Figure 1(a). In the example network, the measured EBR of the link between the AP and client 4 is 5.5Mbps, and the measured EBR of the other links is 11Mbps. Under max-min throughput fairness, each client receives  $\frac{11}{5}$ Mbps bandwidth and the aggregate throughput is  $\frac{44}{5}$ Mbps.

## B. Max-min time fairness

For each node  $i$  and each node  $j \in \mathcal{Q}_i$ , we define node  $j$ 's *time share at node  $i$*  to be the amount of node  $i$ 's time needed by traffic originating from nodes in  $\mathcal{T}_j$  averaged over  $\mathcal{T}_j$ , which is given by

$$t_j^i = \frac{\frac{B_j}{r_{j,i}} + \frac{X_i \cdot B_j}{r_{i,P_i}}}{|\mathcal{T}_j|}.$$

For node  $i$  itself, its time share is its own time needed by its own traffic, which is  $t_i^i = \frac{X_i \cdot b_i}{r_{i,P_i}}$ . Given a bandwidth allocation  $\mathcal{B}$  whose bandwidth vector is  $\vec{B} = (b_1, b_2, \dots, b_n)$ , we define the *time share vector at node  $i$* ,  $\vec{T}_i = (t_{j_1}^i, t_{j_2}^i, \dots, t_{j_k}^i)$ , as the time shares of the  $k = |\mathcal{Q}_i^+|$  nodes in  $\mathcal{Q}_i^+$  sorted in non-decreasing order.

Ideally, time fairness in a multi-hop tree should ensure that for any node  $i$ , nodes in  $\mathcal{Q}_i^+$  receive equal time share at node  $i$ . However, if some node  $j \in \mathcal{Q}_i^+$  is not able to consume its time share at node  $i$ , its surplus time share at node  $i$  should be evenly distributed to other nodes in  $\mathcal{Q}_i^+$ . This ideal principle leads to our proposed notion of max-min time fairness formally defined as follows.

*Definition 2 (Max-Min Time Fairness):* A feasible bandwidth allocation  $\mathcal{B}$  is max-min time fair if and only if at each node  $i$ , its corresponding time share vector  $\vec{T}_i = (t_{j_1}^i, t_{j_2}^i, \dots, t_{j_k}^i)$  has the same or higher lexicographical value than that of any other feasible bandwidth allocation.

The max-min time fair bandwidth allocation within the WLAN in Figure 1(a) is shown in Figure 1(b). Under max-min time fairness, client 1, 2, 4 each receives  $\frac{11}{6}$  Mbps bandwidth, and client 3 receives  $\frac{11}{3}$  Mbps bandwidth. The aggregate throughput is  $\frac{55}{6}$  Mbps, which is more than the  $\frac{44}{5}$  Mbps aggregate throughput of max-min throughput fairness. Compared with max-min time fairness, max-min throughput fairness allocates more bandwidth to low EBR clients at the cost of high EBR clients and aggregate throughput, while max-min time fairness leads to a higher aggregate throughput by protecting high EBR clients at the cost of low EBR clients. This is consistent with the case of single-hop WLANs.

In single-hop WLANs, clients associated with the same AP should receive equal time of the AP under max-min time fairness, which is consistent with time-based fairness. In multi-hop WLANs, our proposed max-min time fairness turns out to be quite successful in two ways.

- As we will later see, compared with single-hop WLANs using time-based fairness, multi-hop WLANs using max-min time fairness universally improve client throughput.
- Compared with max-min throughput fairness, it generally leads to a higher aggregate throughput by protecting forwarding clients near the AP. This is appealing in many cases and more importantly, gives better motivation for clients to serve as a forwarding node near the AP, which means they will forward more traffic than their descendants in the tree.

## C. Objective and design

The objective of this paper is to design solutions for improving throughput without sacrificing fairness in multi-hop WLANs. To achieve this objective, two problems need to be addressed. First, we need to design a tree construction algorithm to organize clients into a multi-hop structure such that fair bandwidth allocation within this structure leads to improved throughput. Subsequently, we need to design algorithms taking the determined structure and link EBRs as input to perform fair bandwidth allocation within the structure. As the tree construction algorithm relies on the fair bandwidth allocation algorithms to evaluate the quality of a structure, we first present our fair bandwidth allocation algorithms in Section III and Section IV, respectively.

### III. MAX-MIN THROUGHPUT FAIR ALLOCATION

In this section, we present the idea, design, and analysis of *Max-Min Fair Allocation (MMFA)*, an optimal algorithm for max-min throughput fair bandwidth allocation in multi-hop WLANs. For ease of understanding, we start with the simple case where the WLAN is organized into a tree rooted at the only AP within the WLAN. We will investigate the case of multi-AP WLANs as well as other extensions in Section V.

#### A. General idea

Given the tree structure of a multi-hop WLAN, MMFA takes a bottom-up approach. At each node  $i$  in the tree, MMFA first recursively conducts max-min throughput fair bandwidth allocation within the individual subtrees rooted at node  $i$ 's child nodes (if any), and then conducts max-min throughput fair bandwidth allocation within  $\mathcal{T}_i$  by performing *Pump-and-Drain* at node  $i$ . The Pump-and-Drain operation is as follows.

- *Pump*: If node  $i$  is a client, MMFA assigns a certain amount of bandwidth to node  $i$  such that node  $i$  receives the highest bandwidth among nodes in  $\mathcal{T}_i$  and node  $i$ 's time is completely used. There is no need to perform Pump at APs since APs should receive 0 bandwidth.
- *Drain*: The bandwidth allocation resulting from Pump may not be feasible, because node  $i$  may be overloaded after being assigned the highest bandwidth among nodes in  $\mathcal{T}_i$ . In that case, we need to decrease the bandwidth assigned to nodes in  $\mathcal{T}_i$  to ensure that the resulting bandwidth allocation is feasible and max-min throughput fair within  $\mathcal{T}_i$ .

Based on this general idea, we next present the detailed design and correctness proof of MMFA.

#### B. Detailed design

To perform Pump-and-Drain, each node  $i$  maintains and reports to its parent the following information, which can be locally determined by aggregating the information reported by its children (if any).

- The bandwidth assigned to node  $i$ , namely  $b_i$ .
- The total bandwidth assigned to nodes in  $\mathcal{T}_i$ , which is given by  $B_i = b_i + \sum_{j \in \mathcal{Q}_i} B_j$ .

- The distinct amounts of bandwidth assigned to clients in  $\mathcal{T}_i$ , which are stored in array  $\mathcal{L}_i$  in non-decreasing order. Moreover, it is assumed that  $\mathcal{L}_i$ . For simplicity, we assume that  $\mathcal{L}_i$  is automatically compacted so that  $|\mathcal{L}_i|$  is always equal to the current number of distinct amounts.
- In array  $\mathcal{N}_i$ , the  $k$ th item  $\mathcal{N}_i[k]$  is the set of clients in  $\mathcal{T}_i$  whose assigned bandwidth is  $\mathcal{L}_i[k]$ .

MMFA is defined as a recursive procedure. In particular, the execution of MMFA at node  $i$  consists of two steps.

- MMFA recursively calls MMFA for each child node  $j \in \mathcal{Q}_i$  to conduct max-min throughput fair bandwidth allocation within  $\mathcal{T}_j$ .
- After recursive MMFA executions at nodes in  $\mathcal{Q}_i$  have returned, the MMFA execution at node  $i$  concludes by performing Pump-and-Drain at node  $i$  to achieve a max-min throughput fair bandwidth allocation within  $\mathcal{T}_i$ . After that, the MMFA execution at node  $i$  returns with its locally maintained information and report the information to the parent of node  $i$  (if any).

MMFA runs in a distributive way. For each tree rooted at an AP in the WLAN, MMFA is called for the AP, which then recursively calls MMFA for its descendants in the tree. The sequence of recursive MMFA executions expand in a top-down fashion and return in a bottom-up fashion (reporting local information to their calling MMFA execution at their parent). Finally, the AP determines the max-min throughput fair bandwidth allocation within the whole tree and spreads the allocation to clients in a top-down fashion. Link EBRs are periodically measured by clients and APs, and are reported in a bottom-up fashion along the tree so that the root node of each subtree has complete information to correctly perform bandwidth allocation within the subtree.

Pump-and-Drain is described in details as follows.

**Pump:** Once we have determined the value of  $b_i$  (which is currently initialized to 0), it will be straightforward to determine the value of  $B_i$ ,  $\mathcal{L}_i$ , and  $\mathcal{N}_i$  according to their foregoing description and the data structures reported by nodes in  $\mathcal{Q}_i$ . For simplicity, we assume that these data structures are implicitly updated each time a bandwidth allocation adjustment is made. If node  $i$  is an AP,  $b_i = 0$  and there is no need to perform Pump. If node  $i$  is a client, the resource constraint at node  $i$  dictates that  $W_i = \frac{b_i}{r_{i,P_i}} + \sum_{j \in \mathcal{Q}_i} \left( \frac{B_j}{r_{j,i}} + \frac{B_j}{r_{i,P_i}} \right) \leq 1$ , where  $W_i$  represents the fraction of node  $i$ 's time needed to support the  $B_i$  bandwidth assigned to nodes in  $\mathcal{T}_i$ . We refer to  $W_i$  as the *workload* on node  $i$ . If  $W_i \geq 1$ , node  $i$  is considered *saturated*. To make  $W_i = 1$ , the amount of bandwidth that should be assigned to node  $i$  is

$$\Delta = \left( 1 - \sum_{j \in \mathcal{Q}_i} \left( \frac{B_j}{r_{j,i}} + \frac{B_j}{r_{i,P_i}} \right) \right) \cdot r_{i,P_i}. \quad (1)$$

- If  $\Delta$  turns out to be the highest bandwidth assigned to nodes in  $\mathcal{T}_i$ , we assign  $\Delta$  bandwidth to node  $i$ . Since  $W_i = 1$ , there is no need to perform Drain. Pump-and-Drain is thus done.

- If  $\Delta$  is not the highest bandwidth, we assign the current highest bandwidth,  $\mathcal{L}_i[|\mathcal{L}_i|]$ , to node  $i$ . Consequently,  $W_i > 1$  and Drain needs to be performed to decrease the bandwidth assigned to nodes in  $\mathcal{T}_i$  so that  $W_i = 1$ .

**Drain:** Drain is performed in an iterative fashion. During each iteration, only those nodes with the highest bandwidth in  $\mathcal{T}_i$  (i.e., nodes in  $\mathcal{N}_i[|\mathcal{L}_i|]$ ) are decreased, each by an appropriate amount  $\delta$ . Let  $n_j$  denote the number of nodes in  $\mathcal{T}_j$  that are also in  $\mathcal{N}_i[|\mathcal{L}_i|]$ , namely  $n_j = |\mathcal{T}_j \cap \mathcal{N}_i[|\mathcal{L}_i|]|$ . To make  $W_i = 1$ , the amount of bandwidth to be decreased at each node in  $\mathcal{N}_i[|\mathcal{L}_i|]$ ,  $\delta$ , is given by

$$W_i - \left[ \frac{X_i \cdot \delta}{r_{i,P_i}} + \sum_{j \in \mathcal{Q}_i} \left( \frac{n_j \cdot \delta}{r_{j,i}} + \frac{X_i \cdot n_j \cdot \delta}{r_{i,P_i}} \right) \right] = 1$$

$$\implies \delta = \frac{W_i - 1}{\frac{X_i}{r_{i,P_i}} + \sum_{j \in \mathcal{Q}_i} \left( \frac{n_j}{r_{j,i}} + \frac{X_i \cdot n_j}{r_{i,P_i}} \right)}.$$

- If  $\delta > \mathcal{L}_i[|\mathcal{L}_i|] - \mathcal{L}_i[|\mathcal{L}_i| - 1]$ , we decrease the assigned bandwidth of each node in  $\mathcal{N}_i[|\mathcal{L}_i|]$  by  $\mathcal{L}_i[|\mathcal{L}_i|] - \mathcal{L}_i[|\mathcal{L}_i| - 1]$  and repeat this iterative adjustment again.
- Otherwise, we decrease the assigned bandwidth of each node in  $\mathcal{N}_i[|\mathcal{L}_i|]$  by  $\delta$  and it is now the case that  $W_i = 1$ . Drain is thus done.

### C. Correctness proof

*Theorem 1:* MMFA achieves max-min throughput fairness.

*Proof:* Without loss of generality, we ignore the trivial case where the tree is a singleton of an AP. In a bottom-up order, we prove by induction on the depth of nodes in the tree that, after MMFA is executed at any node  $i$ , the bandwidth allocation within  $\mathcal{T}_i$  (denoted by  $\mathcal{B}_i$ ) is feasible and max-min throughput fair and that  $W_i = 1$ .

*Base case:* In the base case, node  $i$  is a leaf client node. Pump will assign  $r_{i,P_i}$  bandwidth to node  $i$ , which is clearly feasible and max-min throughput fair for the singleton  $\mathcal{T}_i$ , and  $W_i = 1$ .

*Inductive case:* If  $\Delta \geq \mathcal{L}_i[|\mathcal{L}_i|]$ , it is clear that  $\mathcal{B}_i$  is feasible within  $\mathcal{T}_i$  and that  $W_i = 1$ .  $\mathcal{B}_i$  is max-min throughput fair, because there is no way to increase the bandwidth of any node in  $\mathcal{T}_i$  without decreasing the bandwidth of another node in  $\mathcal{T}_i$  that has equal or already less bandwidth.

- On one hand, there is no way to increase the bandwidth of node  $i$  without decreasing the bandwidth of another node in  $\mathcal{T}_i$  (which must have the same or less bandwidth), since node  $i$  is saturated.
- On the other hand, for any node  $k$  in  $\mathcal{T}_j$  rooted at some node  $j \in \mathcal{Q}_i$ , there is no way to increase node  $k$ 's bandwidth without decreasing the bandwidth of another node in  $\mathcal{T}_j$  with the same or less bandwidth. Because by our inductive assumption, bandwidth allocation within  $\mathcal{T}_j$  (denoted by  $\mathcal{B}_j$ ) has been max-min throughput fair.

We next examine the case where  $\Delta < \mathcal{L}_i[|\mathcal{L}_i|]$ .

- On one hand, it is clear from the description of Drain that  $\mathcal{B}_i$  is feasible after Drain, and  $W_i = 1$ . Since node

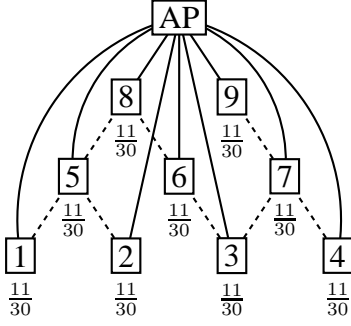


Fig. 3. Max-min throughput fair bandwidth allocation in a single-hop WLAN. Numbers represent bandwidth assigned to individual nodes.

$i$  has been saturated, it is not possible to increase the bandwidth of any node with the highest bandwidth in  $\mathcal{T}_i$  without decreasing the bandwidth of another node in  $\mathcal{T}_i$ , which must have the same or less bandwidth.

- On the other hand, consider any node  $k$  in  $\mathcal{T}_i$  whose bandwidth is not the highest. It is clear that  $k \neq i$  and thus node  $k$  must reside in  $\mathcal{T}_j$  rooted at some node  $j \in \mathcal{Q}_i$ . Meanwhile,  $b_k$  is not decreased by Drain, since only nodes in  $\mathcal{N}_i[\mathcal{L}_i]$  are ever decreased. To prove by contradiction, assume that we can increase  $b_k$  to  $b'_k$  without decreasing the bandwidth of another node in  $\mathcal{T}_i$  with the  $b_k$  or less bandwidth. Let  $\mathcal{B}'_j$  denote the resulting bandwidth allocation within  $\mathcal{T}_j$ , which is clearly feasible. Compared with the original  $\mathcal{B}_j$  prior to Pump-and-Drain, the nodes in  $\mathcal{T}_j$  with  $b_k$  or less bandwidth have never been decreased, since they are not in  $\mathcal{N}_i[\mathcal{L}_i]$ . However, bandwidth of node  $k$  can be increased. This contradicts the inductive assumption that  $\mathcal{B}_j$  has been max-min throughput fair. ■

We illustrate the dynamics of MMFA and the effectiveness of multi-hop WLANs using the example in Figure 3. In the example network, solid lines represent direct associations between clients and the AP. Dashed lines represent unused links between nodes. Client-client links have an EBR of 11Mbps. Links connecting client 8,9 to the AP also have an EBR of 11Mbps. Links connecting client 5,6,7 to the AP have an EBR of 5.5Mbps. Links connecting client 1,2,3,4 to the AP have an EBR of 2Mbps. Under max-min throughput fairness, each client will receive  $\frac{11}{30}$ Mbps bandwidth.

In Figure 2, a multi-hop tree organization of the same network as well as a level-by-level illustration of MMFA applied on the tree are presented. By utilizing high quality client-client links, MMFA significantly improves throughput for every client in the network. In particular, some nodes receive 1Mbps bandwidth and the others receive  $\frac{5}{3}$ Mbps bandwidth, which are almost  $3 \sim 5$  times as much as the  $\frac{11}{30}$ Mbps assigned to each client in the single-hop organization.

#### IV. MAX-MIN TIME FAIR ALLOCATION

Following the general idea of Pump-and-Drain in a bottom-up fashion, we here present the design and analysis of *Time-*

*Based Fair Allocation (TBFA)*, an optimal algorithm for max-min time fair bandwidth allocation in multi-hop WLANs. At this point, we also consider the simple case of single-AP WLANs where clients are organized into a tree rooted at the AP. Multi-AP WLANs as well as other extensions will be investigated in Section V.

##### A. General idea

Given the multi-hop tree of a WLAN, TBFA also takes a bottom-up approach. At each node  $i$  in the tree, TBFA first recursively conducts max-min time fair bandwidth allocation within the subtrees rooted at node  $i$ 's child nodes (if any), and then conducts max-min time fair bandwidth allocation within  $\mathcal{T}_i$  by performing a similar but different Pump-and-Drain operation at node  $i$ . The Pump-and-Drain operation of TBFA is as follows.

- *Pump*: We divide node  $i$ 's time within  $\mathcal{Q}_i^+$  to ensure that: (1) If node  $i$  is an AP, its receives 0 time; otherwise, it receives the highest time share among nodes in  $\mathcal{Q}_i^+$ . (2) Each node  $j \in \mathcal{Q}_i$  either receives the highest time share at node  $i$ , or receives the amount of node  $i$ 's time that is required to support the aggregate throughput  $B_j$  of nodes in  $\mathcal{T}_j$ . Let  $B_j^p$  denote the aggregate throughput of nodes in  $\mathcal{T}_j$  that can be supported with node  $i$ 's that is allocated to nodes in  $\mathcal{T}_j$  by Pump.
- *Drain*: For each node  $j \in \mathcal{Q}_i$ , if  $B_j > B_j^p$ , we need to decrease the bandwidth of nodes in  $\mathcal{T}_j$  appropriately to ensure that  $B_j = B_j^p$  and that the resulting bandwidth allocation within  $\mathcal{T}_j$  (denoted by  $\mathcal{B}_j$  is feasible and max-min time fair. Consequently, the resulting bandwidth allocation within  $\mathcal{T}_i$  (denoted by  $\mathcal{B}_i$ ) is max-min time fair as well.

##### B. Detailed design

TBFA is a recursive procedure. In particular, the execution of TBFA at each node  $i$  consists of two steps.

- TBFA recursively calls TBFA for each child node  $j \in \mathcal{Q}_i$  to conduct max-min time fair bandwidth allocation within  $\mathcal{T}_j$ .
- After these recursive TBFA executions at nodes in  $\mathcal{Q}_i$  have returned, the TBFA execution at node  $i$  concludes by performing Pump-and-Drain at node  $i$  to achieve a max-min time fair bandwidth allocation within  $\mathcal{T}_i$ . After that, the TBFA execution at node  $i$  returns with its locally maintained information and report the information to the parent of node  $i$  (if there is one). To implement Pump-and-Drain, node  $i$  needs to maintain and report to its parent  $b_i$ ,  $B_i$ , and  $|\mathcal{T}_i|$ , which can be locally determined by aggregating the information reported by its children.

TBFA runs in a distributive way. For each tree rooted at an AP in the WLAN, TBFA is called for the AP, which then recursively calls TBFA for its descendants in the tree. The sequence of recursive TBFA executions expand in the top-down order and return in the bottom-up order (reporting local information to their calling TBFA execution at their parent). Finally, the AP determines the max-min time fair bandwidth

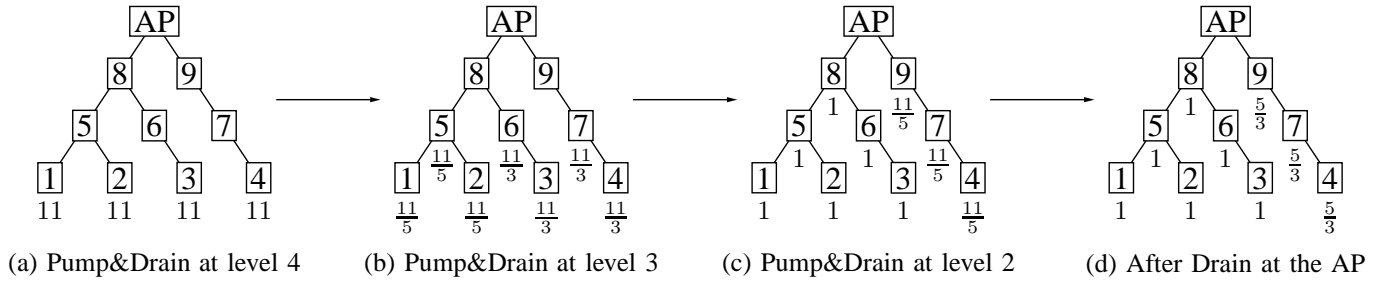


Fig. 2. Level-by-level illustration of MMFA in a multi-hop WLAN. Numbers represent bandwidth assigned to individual nodes.

allocation within the whole tree and spreads the allocation to clients in a top-down fashion. Link EBRs are periodically measured by clients and APs. Unlike MMFA, there is no need to spread link EBRs.

Now we describe the Pump-and-Drain of TBFA in details.

**Pump:** Pump is done in an iterative fashion. Initially, each node in  $\mathcal{Q}_i^+$  is assigned 0 time share at node  $i$ . We refer to nodes in  $\mathcal{Q}_i^+$  that have been assigned time of node  $i$  as *solved* nodes, and refer to the other nodes in  $\mathcal{Q}_i^+$  as *unsolved* nodes. Let  $\mathcal{U}_i$  denote the set of unsolved nodes in  $\mathcal{Q}_i$ . If a child node  $j$  in  $\mathcal{Q}_i$  is assigned time of node  $i$ , we consider all nodes in  $\mathcal{T}_j$  to have been assigned time of node  $i$  as well.

During each iteration, based on the fraction of time available at node  $i$ , denoted by  $C_i$  ( $C_i \leq 1$ ), Pump calculates the average amount of node  $i$ 's time that should be assigned to each unsolved node, which is given by

$$\frac{C_i}{X_i + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|}.$$

Then, the fraction of node  $i$ 's time that should be assigned to an unsolved child node  $j \in \mathcal{Q}_i$  is clearly

$$\frac{C_i \cdot |\mathcal{T}_j|}{X_i + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|}.$$

The aggregate throughput  $B_j^p$  of nodes in  $\mathcal{T}_j$  that is allowed by node  $i$ 's time allocated to nodes in  $\mathcal{T}_j$  is thus given by

$$\frac{B_j^p}{r_{j,i}} + \frac{X_i \cdot B_j^p}{r_{i,P_i}} = \frac{C_i \cdot |\mathcal{T}_j|}{X_i + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|} \implies B_j^p = \frac{\frac{C_i \cdot |\mathcal{T}_j|}{X_i + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|}}{\frac{1}{r_{j,i}} + \frac{X_i}{r_{i,P_i}}}.$$

If node  $i$  is a client, its allowed bandwidth is given by

$$\frac{b_i}{r_{i,P_i}} = \frac{C_i}{1 + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|} \implies b_i = \frac{C_i \cdot r_{i,P_i}}{1 + \sum_{k \in \mathcal{U}_i} |\mathcal{T}_k|}.$$

If node  $i$  is an AP,  $b_i$  is always 0.

If for some unsolved child node  $j \in \mathcal{U}_i$ ,  $B_j < B_j^p$ , the assumption that bandwidth allocation within  $\mathcal{T}_j$  is max-min time fair implies that node  $j$  is not able to consume all of its allocated time of node  $i$ . Let  $\Delta C$  denote the amount of node  $i$ 's time required to support  $B_j$  at node  $i$ , which is given by

$$\Delta C = \frac{B_j}{r_{j,i}} + \frac{X_i \cdot B_j}{r_{i,P_i}}.$$

We *solve* such node  $j$  by allocating  $\Delta C$  time of node  $i$  to node  $j$  and removing node  $j$  from  $\mathcal{U}_i$ . After all such unsolved

child nodes are solved, the current iteration terminates and we repeat the iterative procedure again.

If for every node  $j \in \mathcal{U}_i$  it is the case that  $B_j \geq B_j^p$ , we node  $j$  its calculated time share at node  $i$ , and Pump is done. Node  $i$  is always solved during the last iteration.

**Drain:** Drain is also defined as a recursive procedure. In order to carry out the Drain operation, each node  $i$  locally maintains the following information.

- $\mathcal{L}_i$  is a complete list of distinct amounts of time share at node  $i$  assigned to nodes in  $\mathcal{Q}_i^+$ . For ease of presentation, we assume that  $\mathcal{L}_i$  is an array organized in non-decreasing order. Namely,  $\mathcal{L}_i[1] < \mathcal{L}_i[2] < \dots < \mathcal{L}_i[|\mathcal{L}_i|]$ . Moreover,  $\mathcal{L}_i$  is automatically compacted so that the length of  $\mathcal{L}_i$  is always equal to the number of distinct amounts.
- The  $k$ th element of array  $\mathcal{N}_i$ ,  $\mathcal{N}_i[k]$ , is the set of nodes in  $\mathcal{Q}_i^+$  whose time share at node  $i$  is  $\mathcal{L}_i[k]$ .

Assume that we need to perform Drain at node  $j$  because  $B_j > B_j^p$ . We decrease the time share at node  $j$  of nodes in  $\mathcal{Q}_j^+$  in an iterative procedure that is reverse to the iterative procedure of Pump. During each iteration, only those nodes with the highest time share at node  $j$  (i.e., nodes in  $\mathcal{N}_j[|\mathcal{L}_j|]$ ) are decreased, each by an appropriate amount  $\delta > 0$  such that  $B_j = B_j^p$ .  $\delta$  is given by

$$\delta \cdot X_j \cdot r_{j,i} + \sum_{k \in \mathcal{Q}_j \cap \mathcal{N}_j[|\mathcal{L}_j|]} \frac{\delta \cdot |\mathcal{T}_k|}{\frac{1}{r_{k,j}} + \frac{X_j}{r_{j,i}}} = B_j - B_j^p,$$

which gives us

$$\delta = \frac{B_j - B_j^p}{X_j \cdot r_{j,i} + \sum_{k \in \mathcal{Q}_j \cap \mathcal{N}_j[|\mathcal{L}_j|]} \frac{|\mathcal{T}_k|}{\frac{1}{r_{k,j}} + \frac{X_j}{r_{j,i}}}}.$$

If  $\delta > \mathcal{L}_j[|\mathcal{L}_j|] - \mathcal{L}_j[|\mathcal{L}_j| - 1]$ , we decrease the time share at node  $j$  of nodes in  $\mathcal{N}_j[|\mathcal{L}_j|]$  by  $\mathcal{L}_j[|\mathcal{L}_j|] - \mathcal{L}_j[|\mathcal{L}_j| - 1]$  and repeat this iterative adjustment again. Otherwise, we decrease their time share at node  $j$  by  $\delta$  and it is now the case that  $B_j = B_j^p$ . Then for each child node  $k \in \mathcal{Q}_j$  whose time share at node  $j$  has been decreased during the iterative adjustment, we recursively perform Drain at that node, too. After these recursive Drain executions have returned, the Drain operation at node  $j$  returns as well.

### C. Correctness proof

*Theorem 2:* TBFA achieves max-min time fairness.

*Proof:* In a bottom-up order, we prove by induction on the depth of nodes in the tree that after TBFA has finished executing at any node  $i$ : (1) Node  $i$  is saturated. (2) The computed bandwidth allocation within  $\mathcal{T}_i$  is feasible and max-min time fair. Without loss of generality, we here ignore the trivial case where the tree is an AP singleton.

*Base case:* If TBFA is executed at a leaf node  $i$ , node  $i$ 's time is totally allocated to its own traffic. It is straightforward that node  $i$  is saturated and bandwidth allocation at singleton  $i$  is feasible and max-min time fair.

*Inductive case:* For a non-leaf node  $i$ , it is clear from the description of Pump and Drain that resource constraints at individual nodes are always obeyed. Therefore, TBFA achieves a feasible bandwidth allocation. Moreover, node  $i$  is saturated after Pump-and-Drain at node  $i$ .

From the description of Pump, it is clear that for each node  $i$ , any node  $j \in \mathcal{Q}_i^+$  either receives  $t_j^i$  time of node  $i$  or receives the highest time share at node  $i$  among nodes in  $\mathcal{Q}_i^+$ . Drain operation reduces the time share at node  $i$  of nodes in  $\mathcal{Q}_i^+$  in decreasing order of their time share at node  $i$ , and nodes with the highest time share at node  $i$  always remain among the nodes with the highest time share at node  $i$ . Therefore, it remains to be the case that any node  $j \in \mathcal{Q}_i^+$  either receives  $t_j^i$  time of node  $i$  or receives the highest time share at node  $i$  among nodes in  $\mathcal{Q}_i^+$ .

Since node  $i$  is saturated, there is no way to increase the time share at node  $i$  of any node with the highest time share among nodes in  $\mathcal{Q}_i^+$ , without decreasing the time share at node  $i$  of some node in  $\mathcal{Q}_i^+$  that has equal or already less time share at node  $i$ . Any node  $j \in \mathcal{Q}_i^+$  that receives  $t_j^i$  time of node  $i$  can not receive more time share at node  $i$ , since node  $j$  is saturated (by inductive assumption). The conclusion is that bandwidth allocation within  $\mathcal{T}_i$  is max-min time fair. ■

A level-by-level bottom-up illustration of TBFA applied on the example network in Figure 3 is given in Figure 4. The multi-hop tree structure is the same as Figure 2. Compared with the client bandwidth vector computed by MMFA in Figure 2, TBFA generally leads to a higher aggregate throughput by favoring forwarding clients near the AP. This property is appealing in many cases and more importantly, gives better motivation for clients to serve as a forwarding node near the AP, which means they will forward more traffic than their descendants in the tree.

## V. EXTENSIONS

So far we have been dealing with a simplified WLAN model. In real applications, there are a number of practically important factors that need to be considered. Here we extend MMFA and TBFA to address the following issues.

### *Legacy clients*

It is important for our proposed bandwidth allocation algorithms to be incrementally deployable. Namely, they should seamlessly adapt to the case where a considerable portion of clients use legacy interfaces that are not able to participate and

respond to our algorithms. We point out that our algorithms do not assume or rely on the cooperation of such legacy interfaces. For example, a legacy 802.11 client interface is still directly associated with the AP that presents it with the strongest received signal strength indicator, and it is not required to forward traffic for any other node. Since such direct associations form a part of the tree topology and bandwidth allocation can be done by APs and forwarding nodes, our bandwidth allocation algorithms apply as usual.

### *Limited capacity backhaul*

By now, we have been assuming that the AP is connected to the Internet infrastructure through a backhaul link with sufficiently high bandwidth, so that the aggregate throughput of the WLAN is always fully supported. This is the case in many office buildings with 100Mbps or Gigabit LAN infrastructure. However, there are also many cases where the WLAN is connected to the Internet using a limited capacity backhaul such as a 768Kbps DSL link. We point out that a slight extension of our bandwidth allocation algorithms suffices to handle such cases. In particular, we can create a virtual switch node representing the backhaul link and connect it to the AP using a channel with infinite EBR. The virtual switch is also connected to the infrastructure using an uplink with the same EBR as the backhaul link. Like the AP, the virtual switch always receives 0 bandwidth, too. Our bandwidth allocation algorithms apply on this new virtual WLAN as usual.

### *Multi-AP WLANs*

In many locations such as a large office building, a number of APs may be deployed to provide improved coverage and throughput. We point out that it is also straightforward to extend our bandwidth allocation algorithms to handle such cases. If the shared backhaul link has sufficient bandwidth, there is no need to do anything. Because there is no contention between APs for limited backhaul capacity. We just run our algorithms within the trees rooted at individual APs separately. In the presence of a shared backhaul link with limited capacity, we can extend our algorithms in a similar way. In particular, we create a virtual switch node representing the backhaul link and connect it to the APs using channels with infinite EBR. The virtual switch is also connected to the infrastructure using an uplink with the same EBR as the backhaul link. The virtual switch always receives 0 bandwidth. Our bandwidth allocation algorithms apply on this new virtual WLAN as usual.

## VI. TREE CONSTRUCTION ALGORITHM

Our fair bandwidth allocation algorithms in combination with appropriate multi-hop structures can significantly improve client throughput without sacrificing fairness. For the example WLAN in Figure 3, the multi-hop structure in Figure 5 demonstrates the effectiveness of a good tree structure. Using the multi-hop structure in Figure 2(d), the max-min throughput fair bandwidth allocation assigns 1Mbps bandwidth to some clients and  $\frac{5}{3}$ Mbps bandwidth to other clients. However, using the multi-hop structure in Figure 5 leads to a more throughput



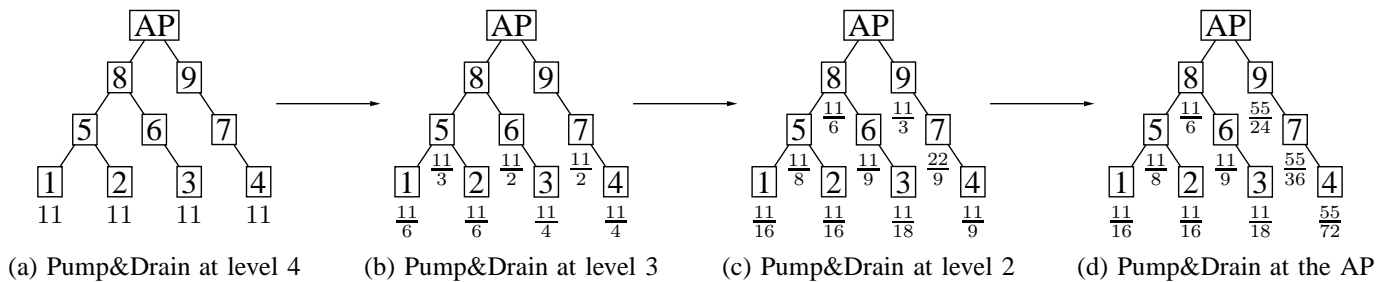


Fig. 4. Level-by-level illustration of TBFA in a multi-hop WLAN. Numbers represent bandwidth assigned to individual nodes.

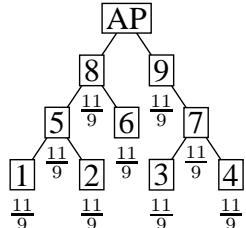


Fig. 5. Effectiveness of a good multi-hop structure. Numbers represent assigned bandwidth of individual nodes.

fair bandwidth allocation where each client receives  $\frac{11}{9}$  Mbps bandwidth. In this section, we study the problem of finding good structures and present our *Tree Construction Algorithm (TCA)* for that purpose.

Before we can proceed to “find a good structure”, it still remains to define a notion of “good structure”. Consider two structures,  $\alpha$  and  $\beta$ , whose resulted client bandwidth vectors under some certain fairness policy are  $\mathcal{B}_\alpha$  and  $\mathcal{B}_\beta$ , respectively. We define  $\mathcal{T}_\alpha$  to be “better” than  $\mathcal{T}_\beta$  if and only if  $\mathcal{B}_\alpha$  has a higher lexicographical value than  $\mathcal{B}_\beta$ . We adopt this definition of “good structure” because our algorithms significantly improve throughput in multi-hop WLANs. Therefore, in defining “good structure” we should be more focused on fairness to balance between throughput and fairness. However, we point out that the reader is free to use any other notion of “good structure” in our TCA to find good structures of interest.

Finding the optimal network topology is a non-trivial task. In the case of throughput-based max-min fairness, this problem is actually provably intractable. Bejerano *et al.* [17] have shown that it is NP-hard to find the optimal association between APs and clients such that each client is associated with one single AP. The NP-hardness of finding the optimal multi-hop structure directly follows, since the problem is a more generalized problem.

Since finding the optimal structure is a non-trivial and even intractable task, we here present our TCA, a heuristic to incrementally improve network structure in a smooth way. TCA runs in a (possibly periodic) iterative fashion. During each iteration, for each smart client node  $i$  (that implements our scheme) in the WLAN, it probes to *migrate* node  $i$  with its subtree from its current location to become a child of another node  $j$ , which may be one of the APs or a smart client that is not in  $\mathcal{T}_i$ . The resulting client bandwidth vector is calculated

using the corresponding bandwidth allocation algorithm. All such candidate  $(i, j)$  pairs are tested. If the bandwidth vector of the best  $(i, j)$  pair is better than the current bandwidth vector, TCA migrates client node  $i$  with its subtree to be a child of node  $j$ . This resulting new structure is the locally optimal structure that we can find at this point, and is accepted as the new structure. After that, TCA calls MMFA or TBFA for the new structure to perform fair bandwidth allocation, according to the adopted fairness policy. The migration decision and the corresponding bandwidth allocation within the WLAN are notified to the clients involved. The involved clients perform the migration and rate control accordingly. This iterative procedure halts at the point where a locally better structure can not be found.

For example, assume the current structure of the WLAN in Figure 3 is the one in Figure 2(d). MMFA assigns 1 Mbps bandwidth to some clients and  $\frac{5}{3}$  Mbps bandwidth to other clients. During the following iteration, TCA decides that migrating client 3 to become a leaf child of client 7 leads to the better structure in Figure 5, where MMFA assigns  $\frac{11}{9}$  Mbps bandwidth to every client, which is considered better according to max-min throughput fairness.

Node arrival and departure can be smoothly handled in a similar way. Each time a new client joins the WLAN, TCA tests all (smart) accepting nodes and picks the one that will lead to the locally best new structure after accepting the new client as its child. The new client is then attached as its child, and bandwidth allocation within the new structure is conducted using the appropriate fair bandwidth allocation algorithm. If fast association is preferred, TCA may pick one of the APs based on some quick evaluation, and try to improve the resulting structure in successive iterations as usual. Such quick evaluation may be highest EBR, strongest received signal strength indicator (RSSI), or least-loaded-first (LLF), etc. When a node leaves the WLAN, its children (if any) can be directly attached to some AP (based on similar quick evaluations) in order to minimize the communication disruption perceived by clients which are descendants of the leaving node. After adjusting bandwidth allocation, TCA will try to improve the resulting structure as usual.

Legacy clients still associate with APs as they usually do, and TCA does not interfere with them at all. They are neither required to migrate nor required to accept migrating clients as their children. As legacy clients do not have any child node,

they do not need to participate bandwidth allocation algorithm, either. Because their own bandwidth can be controlled by the AP that they are associated with.

In summary, our solution combining our TCA and bandwidth allocation algorithms provides a practically interesting smooth transition from legacy single-hop WLANs to smart multi-hop WLANs by two means.

- Our solution seamless integrates legacy clients with smart clients and hence allows a smooth transition from legacy technology to smart technology through incremental deployment. Such incremental deployment is not only more feasible than global upgrade, but also better motivated by our proposed algorithms, which reward individual clients investing to upgrade their device with much more perceivable throughput improvement.
- TCA constantly improves network throughput by improving network structure. Arrival and departure of nodes can be handled as structure changes in the same smooth way.

## VII. EVALUATION

In our simulated scenarios, we compare the performance of our proposed solutions with the strongest received signal strength indicator (RSSI) method as well as the *integral load balancing algorithm (ILBA)* proposed by Bejerano *et al.* [17]. For max-min throughput fairness, we compare our “TCA+MMFA” with both RSSI and ILBA. For max-min time fairness, we compare our “TCA+TBFA” with RSSI. Since ILBA is designed for optimizing max-min throughput fairness, it is not compared under max-min time fairness.

For comparison, we choose link EBRs in the same way as Bejerano *et al.* [17], according to the bit rates commonly advertised by 802.11b vendors. In particular, we assume that the EBR is 11Mbps for links no longer than 50 meters, 5.5Mbps for links no longer than 80 meters, 2Mbps for links no longer than 120 meters, and 1Mbps for links no longer than 150 meters, respectively. The transmission range of an AP is 150 meters. The backhaul link connecting APs to the infrastructure has a typical LAN capacity of 100Mbps, which is more than enough to support the aggregate throughput of the WLAN. Therefore, the bottleneck is on the WLAN side.

### Performance metrics

The performance of our solutions has two key aspects.

- *Throughput*: We study the client throughput distribution provided by individual schemes. In particular, we examine the lowest client throughput, median client throughput, highest client throughput, and aggregate throughput.
- *Fairness*: We use Jain’s Fairness Index [20] of to evaluate the fairness provided by individual schemes. The Jain’s Fairness Index of a bandwidth vector  $\vec{B} = (b_1, b_2, \dots, b_n)$  is given by

$$\frac{(\sum_{i=1}^n b_i)^2}{n \times \sum_{i=1}^n b_i^2}.$$

Intuitively, a bandwidth vector’s Jain’s Fairness Index is 1 if it is perfectly fair (i.e., clients receive equal bandwidth),

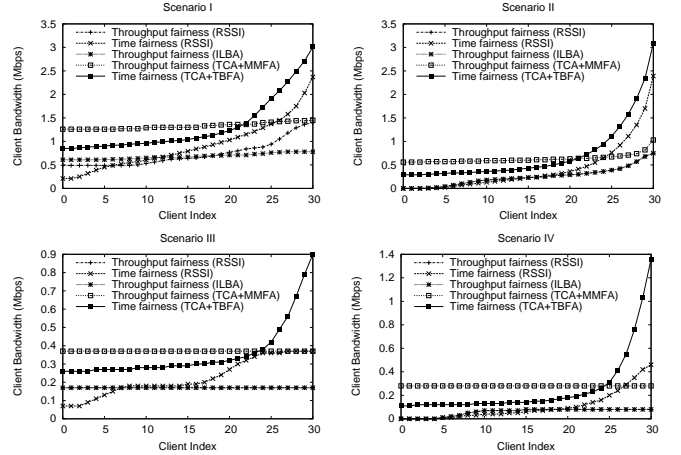


Fig. 6. Bandwidth allocation by different schemes in different scenarios.

and is  $\frac{1}{n}$  if it is completely unfair (i.e., only one client is assigned bandwidth and all other clients are not).

Additionally, as TCA tries to iteratively improve the multi-hop tree structure, we also study the converging speed and adaptation ability of our proposed solutions.

- *Convergence*: We examine the converging process of our proposed solutions, starting from the single-hop association based on RSSI until TCA converges to some locally optimal multi-hop tree structure.
- *Adaptation*: We examine the ability of our solutions to quickly adapt to network topology changes such as node join and node leave. Both regular topology changes and random topology changes are simulated.

### Throughput and fairness

We first compare the client throughput provided by the schemes in comparison. In our network setting, 30 clients are distributed in a square area uniformly at random. We believe this represents a moderately loaded network. Four representative scenarios are examined in our simulations.

- *Scenario I*:  $150m \times 150m$  area, one AP at each corner.
- *Scenario II*:  $300m \times 300m$  area, one AP at each corner.
- *Scenario III*:  $150m \times 150m$  area, one AP at the center.
- *Scenario IV*:  $300m \times 300m$  area, one AP at the center.

For each scenario, we conduct 1000 simulations and present the results in Figure 6. For each individual scheme in comparison, we present client throughput averaged over 1000 simulations in non-decreasing order and we assume that clients are indexed in this order as well. Note that in Scenario II, clients barely have access to more than one AP, so ILBA behaves essentially the same as RSSI. In Scenario III and IV, ILBA does not make difference at all since there is only one AP. Intuitively, the reader can think of the network setting as that of an indoor environment. Although in a typical indoor environment nodes may not be as far away from each other as in our network setting, various indoor signal degradation effects are similar to the effect of longer distance.

TABLE II  
JAIN’S FAIRNESS INDEX OF SCHEMES IN COMPARISON.

	Throughput fairness			Time fairness	
	RSSI	ILBA	TCA+MMFA	RSSI	TCA+TBFA
Scenario I	0.88	0.99	1.00	0.76	0.84
Scenario III	1.00	1.00	1.00	0.85	0.84
Scenario II	0.66	0.66	0.98	0.39	0.55
Scenario IV	0.82	0.82	1.00	0.44	0.46

TABLE III  
AGGREGATE THROUGHPUT (MBPS) OF SCHEMES IN COMPARISON.

	Throughput fairness			Time fairness	
	RSSI	ILBA	TCA+MMFA	RSSI	TCA+TBFA
Scenario I	22.09	20.53	39.91	27.88	40.26
Scenario II	7.72	7.72	19.08	13.20	22.23
Scenario III	5.15	5.15	11.00	6.88	10.90
Scenario IV	1.75	1.75	8.45	3.25	8.04

From the simulation results presented in Figure 6, the following conclusions clearly stand out. (1) Our proposed “TCA+MMFA” solution provides universally better client throughput than RSSI for max-min throughput fairness and ILBA for max-min throughput fairness. (2) Although our proposed “TCA+TBFA” solution is not as fair as “TCA+MMFA”, it provides universally better client throughput than RSSI for max-min time fairness. Moreover, it also provides universally better client throughput than RSSI for max-min throughput fairness and ILBA for max-min throughput fairness. This is an appealing property of “TCA+TBFA”. (3) Our multi-hop WLAN solutions not only improves client throughput, but also improves coverage. In a  $300m \times 300m$  area, clients located around the center can not directly associate with any AP. Techniques based on single-hop direct association such as RSSI and ILBA will allocate 0 bandwidth to these clients. In contrast, our multi-hop WLAN solutions can effectively provide good throughput for these clients as well.

Using the same client throughput data, we also examine the fairness properties of and aggregate throughput provided by schemes in comparison. To evaluate their fairness properties, we present the Jain’s Fairness Index [20] computed from the bandwidth vector of each scheme in Table II. The aggregate throughput of individual schemes is presented in III. From these tables, two conclusions can be drawn by comparison. (1) Our multi-hop schemes significantly improve throughput without sacrificing fairness. If clients can directly associate with APs (Scenario I and III), the aggregate throughput is improved by up to 114%. If some clients can not directly associate with any AP (Scenario II and IV), even much more throughput improvement can be achieved. (2) In cases where some clients can not directly associate with any AP, our multi-hop schemes significantly improve fairness as well, by providing good throughput for those clients.

#### Dynamic topologies, convergence, and adaptation

As TCA works in an iterative fashion, we examine the convergence properties and adaptation properties of “TCA+MMFA” and “TCA+TBFA” using Scenario I and 40

clients. Simulation results of the the two examined solutions are presented in Figure 7 and Figure 8, respectively. For each solution, we first examine its convergence process starting from the single-hop configuration where each of the 40 clients directly associates with an AP, until after TCA converges to a locally optimal multi-hop tree structure. The lowest client bandwidth, the median client bandwidth, and the highest client bandwidth at each iteration of TCA are presented. Note that TCA aims to improve max-min throughput fairness, so the lowest client bandwidth is only increasing.

After TCA has converged, the adaptation properties of each solution are investigated under two different scenarios: regular topology changes and random topology changes. The lowest client bandwidth at each iteration of TCA is presented. In the scenario of regular topology changes, topology changes occur every 20 iterations. In the scenario of random topology changes, a topology change occurs with probability 0.25 at each iteration. Each topology change involves a random node departure followed by a random node arrival. Topology changes are handled by TCA using the proposed quick evaluation based on RSSI. In the figures, each topology change is represented by a dashed vertical line. From the figures, the conclusion is that our solutions converge quickly and are quite responsive to topology changes.

## VIII. RELATED WORK

Fairness in wireline networks has been extensively studied in the literature, but resource allocation constraints significantly differ in wireless networks. Therefore, fair bandwidth allocation problems require a fresh investigation in the context of wireless network. We briefly review the most relevant ones.

In [21], Nandagopal *et al.* propose scheduling schemes for maximizing the sum of user utility in wireless networks, and point out that max-min fairness can be achieved as a special case using a certain choice of utility function. Tassiulas and Sarkar [22] argue that the optimization scheme becomes inefficient in such special cases and that max-min fair bandwidth allocation should be addressed separately. To derive a solution for max-min fair bandwidth allocation, the authors use a network model with a number of simplifying assumption. For example, it is assumed that links that do not share nodes will never contend for channel access. Moreover, only single-hop flows are considered. [23] and [24] study arbitrary link contention graphs, but stick to the formulation where only single-hop flows are considered. Recently, Gambiroza *et al.* [25] take efforts to formulate the case of multi-hop flows and arbitrary link contention graphs. However, rigid analysis and bandwidth assignment are only presented for the special case where only one link within the whole network can be active at any point of time. In other words, the link contention graph is a clique containing all the links.

This paper differs from these previous work in the following ways. First, we focus on the case of multi-hop WLANs. Second, we focus on allocating achievable fair shares of bandwidth to individual flows instead of scheduling and queueing schemes. Third, we not only address multi-hop flows, but also

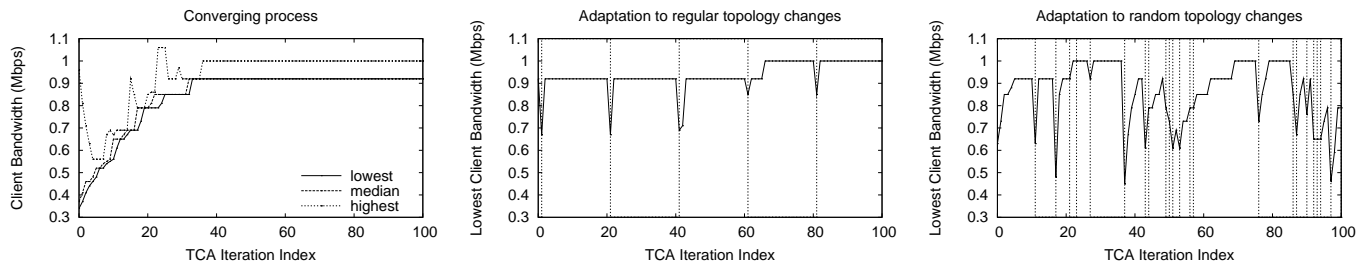


Fig. 7. Convergence/adaptation properties of TCA+MMFA under different scenarios. Vertical lines mark the time a topology change occurs.

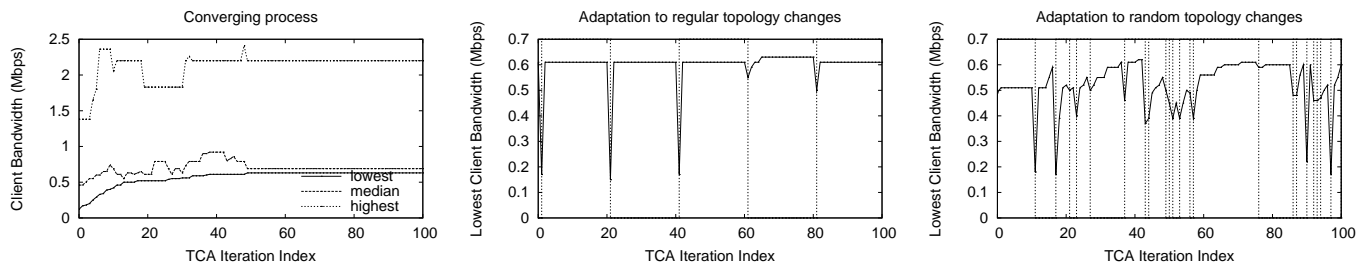


Fig. 8. Convergence/adaptation properties of TCA+TBFA under different scenarios. Vertical lines mark the time a topology change occurs.

use a more general formulation that (implicitly) accommodates arbitrary link contention graphs. Fourth, we also consider the relevant problem of choosing an appropriate multi-hop structure. Fifth, we also address the problem of choosing an appropriate multi-hop structure.

In the context of single-hop WLANs, a number of researchers have investigated fairness and throughput issues. To name a few, Heusse *et al.* [10] report the “performance anomaly” of IEEE 802.11 WLANs due to the throughput-based fairness they implement. Tan and Guttag [5] propose time-based fairness to improve aggregate throughput by reserving a fixed share of channel access time for high bit rate clients. For multi-AP WLANs, Bejerano *et al.* [17] propose to improve throughput-based max-min fairness by balancing load among APs using association control.

There has also been some previous work on multi-hop WLANs (e.g. [18]). However, formal analysis and precise solutions for throughput maximization and fair bandwidth allocation have been lacking. To the best of our knowledge, this paper is the first of this type.

## IX. CONCLUSION

In this paper, we study the problem of improving throughput while preserving fairness in multi-hop wireless LANs. To ensure fairness, we design optimal algorithms for max-min fair bandwidth allocation and time-based fair bandwidth allocation in multi-hop WLANs. To improve throughput, we design a heuristic to constantly and smoothly search for a better new network structure so that fair bandwidth allocation within the new structure results in a better client throughput profile. Although our results are presented in the context of multi-hop WLANs, with slight modifications they can be generalized to other multi-hop wireless networks such as the emerging wireless backhaul networks and wireless mesh networks. The

proposed solutions seamlessly integrate with legacy devices and hence are incrementally deployable. Simulation results demonstrate that our solutions can significantly improve both throughput (by up to 114% or more) and coverage while preserving fairness.

## REFERENCES

- [1] A. Kamerman and L. Monteban, “Wavelan ii: A high-performance wireless lan for the unlicensed band,” *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.
- [2] Data Sheet of Cisco Aironet 350 Series Access Points.
- [3] ORiNOCO AS-2000 System Release Note.
- [4] D. Kotz, C. Newport, and C. Elliott, “The mistaken axioms of wireless network research,” Computer Science, Dartmouth College, Technical Report TR2003-467, July 2003.
- [5] G. Tan and J. Guttag, “Time-based fairness improves performance in multi-rate wireless lans,” in *USENIX Annual Technical Conference*, 2004.
- [6] M. Balazinska and P. Castro, “Characterizing mobility and network usage in a corporate wireless local area network,” in *MobiSys*, 2003.
- [7] D. Kotz and K. Essien, “Analysis of a campus-wide wireless network,” in *ACM MobiCom*, September 2002.
- [8] C. E. Koksal, H. I. Kassab, and H. Balakrishnan, “An analysis of short-term fairness in wireless media access protocols,” in *ACM SIGMETRICS*, 2000.
- [9] Y. Tay and K. Chua, “A capacity analysis for the ieee 802.11 mac protocol,” *ACM/Baltzer Wireless Networks*, vol. 7, no. 2, pp. 159–171, March 2001.
- [10] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in *IEEE INFOCOM*, 2003.
- [11] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” *Internetworking: Research and Experience*, vol. 1, pp. 3–26, April 1990.
- [12] M. Shreedhar and G. Varghese, “Efficient fair queueing using deficit round robin,” in *ACM SIGCOMM*, 1995.
- [13] P. Goyal, H. M. Vin, and H. Cheng, “Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks,” *IEEE/ACM Transaction on Networking*, October 1997.
- [14] P. Ramanathan and P. Agrawal, “Adapting packet fair queueing algorithms to wireless networks,” in *ACM MobiCom*, 1998.
- [15] S. Lu, V. Bharghavan, and R. Srikant, “Fair scheduling in wireless packet networks,” *IEEE/ACM Transaction on Networking*, vol. 7, no. 4, pp. 473–489, 1999.

- [16] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless lan," in *ACM MobiCom*, 2000.
- [17] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless lans using association control," in *ACM MobiCom*, 2004.
- [18] S. Lee, S. Banerjee, and B. Bhattacharjee, "The case for a multi-hop wireless local area network," in *IEEE INFOCOM*, 2004.
- [19] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987.
- [20] R. Jain, Ed., *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1991.
- [21] T. Nandagopal, T.-E. Kim, X. Guo, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," in *ACM MobiCom*, 2000.
- [22] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *IEEE INFOCOM*, 2002.
- [23] X. L. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation," in *ACM MobiHoc*, 2001.
- [24] H. Luo, J. Cheng, and S. Lu, "Self-coordinating localized fair queueing in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, 2004.
- [25] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," in *ACM MobiCom*, 2004.