Fairness and Load Balancing in Wireless LANs Using Association Control

Yigal Bejerano, *Member, IEEE*, Seung-Jae Han, *Member, IEEE*, and Li (Erran) Li, *Member, IEEE*

***Abstract*—The traffic load of wireless LANs is often unevenly distributed among the access points (APs), which results in unfair bandwidth allocation among users. We argue that the load imbalance and consequent unfair bandwidth allocation can be greatly reduced by intelligent association control. In this paper, we present an efficient solution to determine the user-AP associations for max-min fair bandwidth allocation. We show the strong correlation between fairness and load balancing, which enables us to use load balancing techniques for obtaining optimal max-min fair bandwidth allocation. As this problem is NP-hard, we devise algorithms that achieve constant-factor approximation. In our algorithms, we first compute a fractional association solution, in which users can be associated with multiple APs simultaneously. This solution guarantees the fairest bandwidth allocation in terms of max-min fairness. Then, by utilizing a rounding method, we obtain the integral solution from the fractional solution. We also consider time fairness and present a polynomial-time algorithm for optimal integral solution. We further extend our schemes for the on-line case where users may join and leave dynamically. Our simulations demonstrate that the proposed algorithms achieve close to optimal load balancing (i.e., max-min fairness) and they outperform commonly used heuristics.**

***Index Terms*—Approximation algorithms, IEEE 802.11 WLANs, load balancing, max-min fairness.**

I. Introduction

RECENT studies [2]–[4] on operational Wireless LANS (WLANs) have shown that the traffic load is often distributed unevenly among the access points (APs). In WLANs, by default, each user scans all available channels to detect its nearby APs and associate itself with the AP that has the strongest received signal strength indicator (RSSI), while ignoring its load condition. As users are, typically, not uniformly distributed, some APs tend to suffer from heavy load while adjacent APs may carry only light load or be idle. Such load imbalance among APs is undesirable as it hampers the network from providing fair services to its users. As suggested in existing studies [5]–[7] the load imbalance problem can be alleviated by balancing the load among the APs via intelligently selecting the user-AP association, termed *association control*. Association control can be used to achieve different objectives. For instance, it can be used to maximize the overall system throughput by shifting users to idle or lightly loaded APs and allowing each AP to serve only the users with maximal data rate. Clearly, this objective is not a desired system behavior from the fairness viewpoint. A more desirable goal is to provide network-wide fair bandwidth allocation, while maximizing the minimal fair share of each user. This type of fairness is known as *max-min fairness*. Informally, a bandwidth allocation is max-min fair if there is no way to give more bandwidth to any user without decreasing the allocation of a user with less or equal bandwidth. In this paper, we present efficient user-AP association control algorithms that ensure max-min fair bandwidth allocation and we show that this goal can be obtained by balancing the load on the APs.

A. Related Work

Load balancing in WLANs has been intensely studied. Various WLAN vendors have incorporated proprietary features in the device drivers firmwares [8], [9]. In these proprietary solutions, the APs broadcast their load conditions to the users via the Beacon messages and each user chooses the least loaded AP. In [5]–[7], different association criteria are proposed. These metrics typically take into account factors such as the number of users currently associated with an AP, the mean RSSI, the RSSI of the new user and the bandwidth a new user can get if it is associated with an AP. For example, Balachandran *et al.* [6] propose to associate new users with the AP that can provide a minimal bandwidth required by the user. If there are more than one such AP, the one with the strongest signal is selected. Most of these heuristics only determine the association of newly arrived users. Tsai and Lien [7] propose to reassociate users when some conditions are violated.

Load balancing in cellular networks is usually achieved via dynamic channel allocation (DCA) [10]. Ths method is not applicable to WLANs where each AP normally uses one channel and channel allocation is fixed. Another approach is to use cell overlapping to reduce the call blocking probability and maximize the network utilization. In [12] and [13], a newly arrived user is associated with the cell with the greatest number of available channels. In [14], the fairness issue is addressed by restricting the number of available channels for new calls that are made in overlapping areas. Tinnirello and Bianchi [15] take into account the channel conditions of the users. Recently, load balancing integrated with coordinated scheduling technique has been studied in [11] for CDMA networks. However, these techniques are not suitable to our goal, since they consider different objective functions, e.g., blocking probability.

Most of the work on max-min fairness addresses the problem of allocating bandwidth to a set of predetermined routes in

Manuscript received May 11, 2005; revised February 27, 2006; approved by IEEE/ACM Transactions on Networking Editor E. Knightly. This work was supported in part by KOSEF Grant R01200600001086402006. Part of this paper was published in the Proceedings of ACM MobiCom 2004.

Y. Bejerano and L. Li are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: bej@research.bill-labs.com).

S.-J. Han is with Yonsei University, Seoul, Korea.

Digital Object Identifier 10.1109/TNET.2007.893680

a wired network [16]–[18]. The problem of selecting routes for providing max-min fair bandwidth allocation is studied in [19], [20]. Megiddo [19] addresses the problem in the setting of single-source fractional flow and presents a polynomial time algorithm that finds an optimal max-min fair solution. Extending this work, Kleinberg *et al.* [20] consider the problem where a connection is routed along a single path. In particular, their approach can be applied to the load balancing problem of parallel machine scheduling [21] where each job imposes the same load per unit time on the subset of machines in which it can be run, i.e., a *load conserving* system. They argue that a coordinate-wise constant-factor approximation cannot be found for this problem, and present a prefix-sum 2-approximation algorithm to the fairest fractional solution. In other words, for every integer $k > 0$, the sum of the first k coordinates of the calculated allocation vector sorted in increasing order is at most twice the sum of the first k coordinates of the fairest fractional assignment. They use Megiddo's algorithm [19] to compute a fractional solution and use the rounding scheme of [21] for obtaining an integral solution. This problem is a special case of ours in which each user uses the same bit rate to all the APs it can associate with. Their result cannot be directly applied to our problem since each user gets different rate from different APs, i.e., our jobs are *not load conserving*. In the context of online load balancing of unrelated parallel machines, Aspnes *et al.* [22] and Goel *et al.* [23] present an algorithm with a logarithmic competitive ratio when compared with the offline optimal allocation. We will apply these results to deal with the online case of our problem.

B. Our Contributions

In this paper, we present a method for determining use-AP association that ensures the network-wide max-min fair bandwidth allocation. This goal is achieved by balancing the load of the APs. Previous studies on load balancing in wireless networks have not explicitly considered fairness in conjunction with load balancing. As shown in our simulations, if load-balancing is not done carefully, users may experience even poorer connections compared with the default strongest signal approach. To the best of our knowledge, we are the first that presents an association control algorithm that provides guarantees on the quality of the bandwidth allocation against the optimal solution.

In the existing literature there is no common notion of the load, while naive definitions such as the number of users that are associated with an AP do not properly reflect the AP load [2]–[4]. For rigorous formulation of the association control problem, we introduce a rigorous definition of the load. Under this load definition, we prove the strong correlation between load balancing and max-min fair bandwidth allocation. Since the max-min fair bandwidth allocation problem is NP-hard, we develop approximation algorithms. Ideally, we would like to guarantee to each user a bandwidth of at least $1/\rho$ of the bandwidth that it receives in the optimal (integral) solution, for a constant $\rho \geq 1$. However, due to the unbounded integrality gap, it is impossible to provide this type of approximation [20]. Instead, our guarantees are relative to an optimal fractional solution, where users can be associated with multiple APs simultaneously. The basic steps of our algorithms are as follows. First, we calculate a fractional solution for the max-min fair bandwidth allocation problem. It is the fairest among all possible allocations. Then, we extend the rounding method of [28] to obtain an integral solution where each user can only associate with one AP. In particular, we provide a 2-approximation algorithm for unweighted users and a 3-approximation algorithm for weighted users. In [1], we extend these algorithms also for instances with *bounded-demand users*, where users have upper bound on their traffic demands. In addition to bandwidth fairness, we also consider time fairness. We further extend our schemes for the online case where users may join and leave dynamically. Our simulations demonstrate that the proposed algorithms achieve near-optimal load balancing and outperform popular heuristic approaches. Although, this work currently targets at WLANs, it may be applicable to other wireless networks as well.

II. System Description

A. The Network Model

We consider an IEEE 802.11 WLAN that comprises multiple APs. We use A to denote the set of APs and let m denotes their number, i.e., $m = |A|$. All the APs are attached to a fixed infrastructure, which provides a transmission bit rate of R_a to each AP $a \in A$. Each AP has a limited transmission range and it can serve only users that reside in its range. At any given time a user can be associated with one AP. We define the network coverage area to be the union of the area covered by each AP in A. We use U to denote the set of users within the network coverage and let $n = |U|$ denotes the total number of users in U. The users are assumed to have a quasi-static mobility pattern. In other words, the users are free to move from place to place, but they tend to stay in the same physical locations for long time periods. This assumption is backed up by recent analysis of mobile user behavior [2], [3].

In this study, we assume that adjacent APs use *noninterfering* channels. Thus, transmissions at one cell do not interfere transmissions at other cells. Such interference mitigation can be obtained by allocating orthogonal transmission codes or proper frequency planning. The channel condition between an AP and a user is dynamic. Data flows have bursty characteristics and they generate dynamic load on the APs. Therefore, it is impossible to provide short-term fairness through association control without generating high communication overhead and potentially disrupting ongoing sessions. Instead, we consider long-term[1] fairness, making decisions based on the long-term channel condition, which depends mainly on path loss and slow fading. For each user $u \in U$ and each AP $a \in A$, we use $r_{a,u}$ to denote the average *effective bite rate*.[2]

Throughout this paper, we consider *greedy users* that always have traffic to send or receive, and consume all the allocated bandwidth. Furthermore, we assume that each user $u \in U$ has a weight w_u that specifies its priority. This weight is used to determine the bandwidth allocation, b_u, which it entitles to have

[1]Long-term time scale is measured in terms of tens of seconds.

[2]The effective bit rate also takes into account the overhead of retransmissions due to reception errors.

TABLE I
NOTATIONS

Symbol	Semantics
A	The set of all access points (APs).
U	The set of all users.
R_a	The infrastructure link bite rate of AP a.
$r_{a,u}$	The wireless link bite rate between AP a and user u.
w_u	The weight (priority) of user u.
b_u	The bandwidth allocation of user u.
$\tilde{b}_u$	The normalized bandwidth allocation of user u.
$\vec{B}_u$	A normalized bandwidth allocation vector.
$\mathcal{B}$	A bandwidth allocation matrix.
$x_{a,u}$	The fractional association of user u with AP a.
$\mathcal{X}$	An user-AP association matrix.
y_a	The load on AP a.
$\tilde{Y}$	An upper bound on the AP's loads.
$\vec{Y}$	The APs' load vector.
L_k	The APs of load group k.
$\tilde{L}$	The bottleneck load group.
F_k	The users of fairness group k.
$\tilde{F}$	The bottleneck fairness group.
$\tilde{\mathcal{X}}$	the user-AP association matrix of the bottleneck load group and its corresponding fairness group.
T	The load balancing threshold, *e.g.*, the minimal load that a user may generate on an AP.
ρ^*	The max-min load balanced approximation ratio with threshold T.
$J_{a,u}$	The joint load of user u on AP a on both the infrastructure and wireless links.

with respect to the other users. For instance, a user $u \in U$ entitles to have a bandwidth of $b_u = (w_u/w_v) \cdot b_v$ of any other user $v \in U$ in the same cell. The solution for the *bounded-demand users* can be found in [1].

B. System Description

First, the system requires relevant information on each user $u \in U$, such as its weight w_u and the effective bit rate $r_{a,u}$ for each AP $a \in A$. Second, it needs an algorithm to determine the user-AP association. Third, it needs a mechanism to enforce these association decisions. The effective bit rate $r_{a,u}$ between every user u and every AP a, may not be available because an AP maintains the bit rate information only for the users who are currently associated with it. In fact, the effective bit rates for other APs can only be measured from the user side, by monitoring the signal strength of beacons from nearby APs. The collected information is reported to a *network operation center* (NOC) which runs our algorithm to come up with the user-AP association decisions. Since the users are free to move, the NOC periodically recalculates the optimal user association by using the offline algorithms, described in Section IV. Meanwhile, the NOC may use an online algorithm that maintains the APs' load balancing, which is decribed in Section V. After determining a user association, the NOC notifies the user client software of his decision. The client changes the user association accordingly.

In this paper, we do not address the issue of providing fair service within each AP. We assume that such a feature is available, for instance, by using the IEEE 802.11e extension [24] or some fair scheduling mechanism, such as [25]–[27].

C. Periodic Offline Optimization

We motivate the need for optimization by showing the weakness of the existing heuristic load balancing mechanisms. Example 1 illustrates a case when a naive load balancing mechanism performs poorly. It shows that the least-loaded-first (LLF) method, a widely used load-balancing heuristic, can perform worse than the strongest-signal-first (SSF) method, the default association method of WLANs. In LLF, a user chooses the least-loaded AP, where an AP *load* is inversely proportional to the current bandwidth that its associated users receive. Our simulations in Section VI demonstrate that such bad association decisions by the online heuristics are not rare but rather typical. Similar examples can be found when other association criteria [5]–[7] are used.

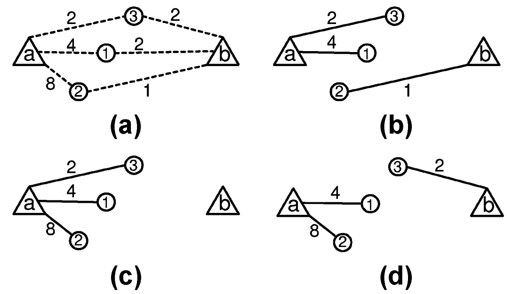


Fig. 1. The weaknesses of online association control mechanism. (a) The wireless system description; (b) the LLF association; (c) the SSF association (case I); (d) the optimal association (SSF case II).

Example 1: Consider a wireless system with two APs, $\mathbf{a}$ and $\mathbf{b}$, and three users {1,2,3}, indexed according to their arrival time, as depicted in Fig. 1(a). In this figure the numbers on the dashed lines represent the bit rate that each user experience from the corresponding AP. We assume that the APs provide fair service to their associated users. In this example we compare LLF, the SSF and optimal association strategies.

The LLF strategy: When user 1 arrives to an empty system, it joins to AP $\mathbf{a}$ that provide the higher bite rate (the stronger signal) among the two APs. Upon the arrival of user 2, AP $\mathbf{a}$ is more loaded than AP $\mathbf{b}$. Therefore, user 2 chooses AP $\mathbf{b}$ although AP $\mathbf{b}$ provides lower bit rate than AP $\mathbf{a}$. As a result, AP $\mathbf{b}$ becomes the most loaded AP. When user 3 arrives, it associates itself with AP $\mathbf{a}$. The final association is given in Fig. 1(b). Consequently, user 1 and 3 receive a bandwidth of (4/3) (from $b/4 + b/2 = 1$, we have $b = 4/3$), while user 2 gets a bandwidth of 1. Clearly, this association is far from the optimal one.

The SSF strategy: In this strategy, user 1 and 2 are associated with AP $\mathbf{a}$, and user 3 randomly selects one of the two APs.

Case I—user 3 chooses AP $\mathbf{a}$: All the users are associated with AP $\mathbf{a}$, as shown in Fig. 1(c), while AP $\mathbf{b}$ is idle. The bandwidth allocated to each user is 8/7. Obviously, this is (almost) the worst possible association.

Case II—user 3 chooses AP $\mathbf{b}$: This results in *the optimal association*, see Fig. 1(d). User 1 and 2 receive bandwidth of 8/3 while user 3 receives a bandwidth of 2. Thus, each user gets twice the bandwidth allocated to it in LLF. □

D. Wireless and Wired Bottlenecks

Though the wireless link is generally considered as the bottle neck. this assumption is not always vaild. For instance, consider a WLAN where the APs are connected to the infrastructure over

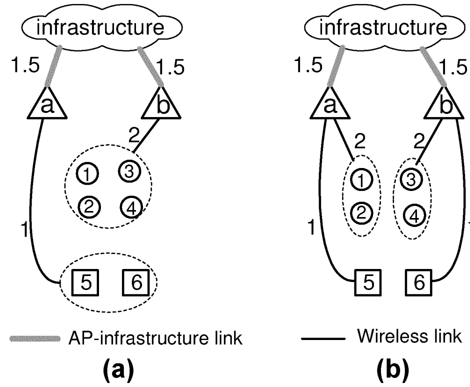


Fig. 2. Examples of bottlenecks both over the wireless and the wired links. (a) An unfair association. (b) The optimal association.

$T1$ lines, whose capacity is around 1.5 Mb/s, as illustrated in Example 2. Example 2 demonstrates the need to consider both the wireless and the wired links for load balancing.

Example 2: Consider a wireless system with 2 APs, $\mathbf{a}$ and $\mathbf{b}$, and 6 users, enumerated from 1 to 6, as depicted in Fig. 2. Users 1,2,3 and 4 experience a bit rate of 2 Mb/s from both APs, while users 5 and 6 have a bit rate of 1 Mb/s from both APs. The APs are connected to a fixed network with $T1$ lines with capacity of 1.5 Mb/s. In the following we consider two possible associations and we analyze the average bandwidth that they provide to the users.

Case I: A fair user association only from the wireless perspective—Consider the association depicted in Fig. 2(a). Here, the system can allocate a bandwidth of 0.5 Mb/s to each user over the wireless links. However, while AP $\mathbf{a}$ can allocate a bandwidth of 0.5 Mb/s to users 5 and 6 on its $T1$ line, AP $\mathbf{b}$ can only provide 3/8 Mb/s to its associated users over its $T1$ line. In this case, the wireless link of AP $\mathbf{a}$ is the bottleneck that affects the bandwidth allocation. Meanwhile, the wired link is the bottleneck of AP $\mathbf{b}$.

Case II: A fair user association—Consider the association shown in Fig. 2(b). This association provides a bandwidth of 0.5 Mb/s to each user over the wired and wireless channels. Observe that in this case different users may gain different service time on the wireless links and wired backhauls. For instance, user 5 captures 1/3 of the service time of the $T1$ link of AP $\mathbf{a}$, while, it is served 1/2 of the time by its wireless channel. This ensures that user 5, indeed, receives a bandwidth of 0.5 Mb/s. □

III. Fairness and Load Balancing

In this section, we provide formal definitions of fair bandwidth allocation and load balancing. Additionally, we describe some useful properties that we need for constructing our algorithmic tools. In the following, we consider two association models. The first is a *single-association* model, so-called an *integral-association*, where each user is associated with a single AP at any given time. This is the association mode that is used in IEEE 802.11 networks. The second is a *multiple-association* model, also termed a *fractional-association*, that allows each user to be associated with several APs and to get communication services from them simultaneously. Accordingly, a user may receive several different traffic flows from different APs, and its bandwidth allocation is the aggregated bandwidth of all of them. This model is used to develop our algorithmic tools for the integral-association case. For both association models, we denote by U_a all the users that are associated with AP $a \in A$ and A_u denotes the set of APs that user $u \in U$ is associated with.

A. Max-Min Fairness

Consider a wireless network as described in Section II-A. A *bandwidth allocation* is a matrix, $\mathcal{B} = \{b_{a,u} | u \in U, a \in A\}$, that specifies the average bandwidth, $b_{a,u}$, allocated to each user $u \in U$ by every AP $a \in A$. We denote by $b_u = \sum_{a \in A} b_{a,u}$ the *aggregated bandwidth* allocated to user u and let $\bar{b}_u = b_u / w_u$ be its *normalized bandwidth* (NB) allocation. On average, AP a is required to serve user u a period of $b_{a,u}/r_{a,u}$ over the wireless channel and a period of $b_{a,u}/R_a$ over the infrastructure link, at every time unit. Consequently, we say that a bandwidth allocation $\mathcal{B}$ is *feasible* if every AP $a \in A$ can provide the required bandwidth to all its associated users both in the wireless and the wired domains, that is, $\sum_{u \in U} b_{a,u}/r_{a,u} \leq 1$ and $\sum_{u \in U} b_{a,u}/R_a \leq 1$. In the case of an integral-association, we also require that each user is associated with a single AP.

Intuitively, a system provides a fair service if all users have the same allocated bandwidth.[3] Unfortunately, such a degree of fairness may cause significant reduction of the network throughput, since all users get the same bandwidth allocation as the bottleneck users, as we illustrate in Example 3 below. The common approach to address this issue of fair allocation that also maximizes the network throughput is to provide *max-min fairness* [18]. Informally, a bandwidth allocation of a weighted system is called *max-min fair* if there is no way to increase the bandwidth of a user without decreasing the bandwidth of another user with the same or less normalized bandwidth. Consider a bandwidth allocation $\mathcal{B}$ and let $\bar{b}_u$ be the normalized bandwidth allocated to user $u \in U$. We define the *normalized bandwidth vector* (NBV), $\vec{B} = \{\bar{b}_1, \cdots, \bar{b}_n\}$ as the users' normalized bandwidth allocations sorted in increasing order and users are renamed according to this order.

Definition 1 (Max-Min Fairness): A feasible bandwidth allocation $\mathcal{B}$ is called *max-min fair* if its corresponding NBV $\vec{B} = \{\bar{b}_1, \cdots, \bar{b}_n\}$ has the same or higher lexicographical value than the NBV $\vec{B}' = \{\bar{b}'_1, \cdots, \bar{b}'_n\}$ of any other feasible bandwidth allocation $\mathcal{B}'$. In other words, if $\vec{B} \neq \vec{B}'$ then there is an index j such that $\bar{b}_j > \bar{b}'_j$ and for every index $i < j$, it follows that $\bar{b}_i = \bar{b}'_i$.

Consider the case that each AP provides a weighted fair bandwidth allocation to its associated users. Then, a user association is termed *max-min fair* if its corresponding bandwidth allocation is *max-min fair*.

Theorem 1: The problem of finding a max-min fair integral association is NP-hard.

Proof: This Theorem can be proved by using a simple reduction from the partition problem [29] to the max-min fair integral association problem. Due to space limitation details of the proof have been omitted. □

Example 3: Consider a wireless system with 3 APs, $A = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, and 5 users, $U = \{1, 2, 3, 4, 5\}$, as depicted in

[3]The same normalized bandwidth in the case of weighted system.

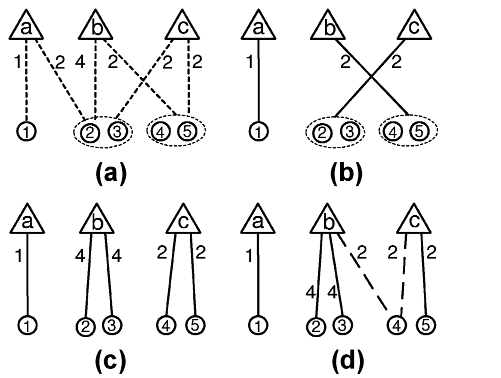


Fig. 3. Examples of a wireless system with 3 APs and 5 users. (a) The wireless system description; (b) a fair association where b = 1 for each user. (c) The max-min fairness single association; (d) the max-min fairness multiple association.

Fig. 3(a). In this figure, doted lines represent possible association and the number near each line represents the bit rate $r_{a,u}$ of the corresponding wireless link. All the users have weight 1 and we assume that all the APs are connected to a high bandwidth infrastructure. Fig. 3(b) presents a feasible fair association in which every user receives a bandwidth $b = 1$, where the solid lines represents the users' associations. Note that this is the maximal bandwidth that can be allocated to user 1. Thus, one can argue that this is the optimal bandwidth allocation. However, in Figs. 3(c) and (d), we describe two feasible associations, in which each user get at least 1 unit of bandwidth. Here, the solid lines indicates an integral association and the dashed line represents fractional association. Fig. 3(c) presents the integral max-min fair allocation with NBV $\vec{B} = \{1,1,1,2,2\}$. While, Fig. 3(d) introduces the fractional max-min fair allocation with NBV $\vec{B} = \{1, 4/3, 4/3, 4/3, 4/3\}$. □

Clearly, the NBV of a fractional max-min fairness allocation always has the same or higher lexicographical value than the NBV of the integral max-min fairness allocation. We will use this property to construct our solution for the integral-association case. Furthermore, consider a max-min bandwidth allocation $\mathcal{B}$ of either a fractional or an integral association. The users can be divided into *fairness groups*, such that each fairness group, $F_k \subseteq U$, consists of all users that experience the same normalized bandwidth allocation, denoted by $\bar{b}_k$.

Theorem 2: Let $\mathcal{B}$ be a max-min fair bandwidth allocation and let $\{F_k\}$ be its corresponding fairness groups. Then all the users served by a given AP belongs to the same fairness group. Formally, for each fairness group F_k, $\bigcup_{u\in F_k} \bigcup_{a\in A_u} U_a = F_k$.

Proof: Initially we prove that $\bigcup_{u\in F_k} \bigcup_{a\in A_u} U_a \supseteq F_k$. This is trivial since every user $u \in F_k$ is included in the set U_a for each AP a it is associated with. Now, we turn to prove that $\bigcup_{u\in F_k} \bigcup_{a\in A_u} U_a \subseteq F_k$. In the case of an integral association, this is satisfied since each user is associated with a single AP and this AP guarantees the same normalized bandwidth allocation to all its associated users. For fractional-association, lets suppose that this property is not valid. Thus, there is an AP a that serves users of two different fairness groups F_j and F_i. Suppose that $\bar{b}_j < \bar{b}_i$. Thus, AP a may increase the bandwidth of its associated users in F_j on behalf of its associated users in F_i. This results in a NBV with a higher lexicographical value. However, this contradicts the assumption that the given allocation is max-min fair. □

B. Min-Max Load Balancing

It is widely accepted that the primary approach for obtaining a fair service is balancing the load on the APs. However, for WLANs the notion of load is not well defined. Several recent studies [2]–[4] have shown that neither the number of users associated with an AP nor its throughput reflect the AP's "load." This motivates the need for an appropriate definition. *Intuitively, the load of an AP needs to reflect its inability to satisfy the requirements of its associated users and as such it should be inversely proportional to the average bandwidth that they experience.* Our load definition captures this intuition and it is also aligned with the standard load definition that are used in the computer science literature, e.g., scheduling of unrelated parallel machines [30]. Consequently, we are able to *extend* existing load balancing techniques to balance the AP loads and obtain a fair service.

We define the notion of fractional association. A *fractional association* is a matrix $\mathcal{X} = \{x_{a,u} | a \in A \wedge u \in U\}$, such that for each user $u \in U$, Equation $\sum_{a\in A} x_{a,u} = 1$ holds. Each parameter $x_{a,u} \in [0,1]$ specifies the *fractional association of user* u *with AP* a. Generally speaking, $x_{a,u}$ reflects the fraction of user u's total flow that it expects to get from AP a. A fractional association $\mathcal{X}$ is termed *feasible* if the users are associated only with APs that can serve them, i.e., for each pair $a \in A$ and $u \in U$, it follows that $x_{a,u} > 0$ only if $r_{a,u} > 0$. Moreover, a feasible association matrix that consists of just 0 and 1 is termed an *integral association*.

Consider a feasible association $\mathcal{X}$, either integral or fractional. We define the *load induced by user* u *on AP* a to be the time that is required of AP a to provide user u a traffic volume of size $x_{a,u} \cdot w_u$. Thus, user u produces a load of $x_{a,u} \cdot w_u / r_{u,a}$ on the wireless channel of AP a and a load of $x_{a,u} \cdot w_u / R_a$ on its backhaul link. Consequently, we define the *load*, y_a, on AP a to be the period of time that takes AP a to provide a traffic volume of size $x_{a,u} \cdot w_u$ to all its associated users $u \in U_a$. Formally,

Definition 2 (AP Load): The *load* on an AP $a \in A$, denoted by y_a, is the maximum of its aggregated loads on both its wireless and infrastructure links produced by all the users. Thus

$$y_a = \max\left\{ \sum_{u\in U} \frac{x_{a,u} \cdot w_u}{r_{u,a}}, \sum_{u\in U} \frac{x_{a,u} \cdot w_u}{R_a} \right\}.$$

Therefore, the load of an AP is given in terms of the time it takes to complete the transmission of certain traffic volume from each associated user. This is not surprising, since the load should be inversely proportional to the bandwidth that the AP provides to its users. Furthermore, the bandwidth that AP a provides to user u is

$$b_{a,u} = x_{a,u} \cdot w_u / y_a \tag{1}$$

We define the *load vector* $\vec{Y} = \{y_1, \cdots, y_m\}$ of an association matrix $\mathcal{X}$ to be the n-tuple consisting of the load of each AP sorted in decreasing order.

Definition 3 (Min-Max Load Balanced Association): A feasible association $\mathcal{X}$ is termed *min-max load balanced* if its corresponding load vector $\vec{Y} = \{y_1, \cdots, y_m\}$ has the same or lower lexicographical value than any other load vector $\vec{Y}' = \{y'_1, \cdots, y'_m\}$ of any other feasible assignment $\mathcal{X}'$. In other words, if $\vec{Y} \neq \vec{Y}'$, then there is an index j such that $y_j < y'_j$ and for every index $i < j$, it follows that $y_i = y'_i$.

Example 4: Consider the wireless system described in Example 3. Fig. 3(c) presents the min-max load balanced association for the single-association case and its load vector is $\vec{Y} = \{1, 1, 1/2\}$. While, Fig. 3(d) introduces the min-max load balanced association for the multiple-association case and its load vector is $\vec{Y} = \{1, 3/4, 3/4\}$. Recall that in this case the association of user 4 is $x_{\mathbf{b},4} = x_{\mathbf{c},4} = 1/2$, thus the load that it induces on each one of these APs is $(1/2) \times (1/2) = 1/4$. □

Consider the min-max balanced association $\mathcal{X}$ and its corresponding load vector $\vec{Y}$. Recall that users can be partitioned into fairness groups. Similarly, APs can be partitioned into *load groups*. Each load group, $L_k \subseteq A$ contains all the APs with the same load, denoted by y_k. Furthermore, lets assume that the indices of the load groups are assigned in decreasing order according to their corresponding loads.

Theorem 3: Consider a min-max load balanced association $\mathcal{X}$ and let $\{L_k\}$ be its APs partitioned into load groups, then each user is associated with APs with the same load, i.e., for each load group L_k we have $\bigcup_{a \in L_k} \bigcup_{u \in U_a} A_u = L_k$.

Proof: Recall that this is trivial in the case of a single association since every user is associated with a single AP. in the case of multiple association it is clear that $\bigcup_{a \in L_k} \bigcup_{u \in U_a} A_u \supseteq L_k$, since each AP is included in the sets A_u of each user u that it serves. We now turn to prove that $\bigcup_{a \in L_k} \bigcup_{u \in U_a} A_u \subseteq L_k$. Let us suppose in contrast that this property is not valid. Thus, there is an user u that is served by to APs a and b such that $y_a > y_b$. Recall that both $x_{a,u}$ and $x_{b,u}$ are strictly more than 0 and less than 1. Thus, we can reduce the load of AP a by shift some load from AP a to AP b. This is obtained by decreasing the fractional association $x_{a,u}$ and increasing a little bit the fraction association $x_{b,u}$. This load shift produces a new association that its corresponding load vector has lower lexicographical value than the load vector of the current association $\mathcal{X}$. However, this contradicts the assumption that $\mathcal{X}$ is a min-max load balanced association. □

Theorem 4: Consider a min-max load balanced association $\mathcal{X}$ and consider any user $u \in U$ and any one of its associated APs $a \in A_u$. Then, the bandwidth allocation for user u determined by $\mathcal{X}$ is $b_u = w_u/y_a$.

Proof: Since $\mathcal{X}$ is a min-max load balanced association, it follows that $\sum_{q \in A_u} x_{q,u} = 1$ and all the APs $q \in A_u$ has the same load y_a as the selected AP a. By (1), we have

$$b_u = \sum_{q \in A_u} b_{q,u} = \sum_{q \in A_u} x_{q,u} \cdot w_u / y_q = w_u / y_a.$$

□

From Theorems 3 and 4, we have Corollary 1.

Corollary 1: Consider a min-max load balanced association $\mathcal{X}$. $\mathcal{X}$ partitions the APs into load groups $\{L_k\}$, where the load on each AP in a group L_k is y_k. It also divides the users into fairness groups $\{F_{k'}\}$ such that all the users in the same group experience the same normalized bandwidth $\bar{b}_{k'}$. Furthermore, the APs of a given load group L_k serve only users from a corresponding fairness group $F_{k'}$ and the normalized bandwidth that each user in $F_{k'}$ experiences is $1/y_k$.

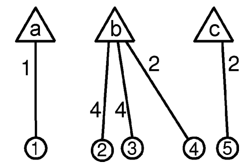


Fig. 4. Examples of a single association that is min-max load balanced but is not max-main fair.

In the following we refer to the load group of the most loaded APs and the corresponding fairness group as the bottleneck groups. We now turn to prove the strong relationship between fairness and load balancing in the case of fractional-association. A sketch of Theorem 5's proof can be found in Appendix VII.

Theorem 5 (The Main Theorem): In the fractional-association case, a min-max load balanced association $\mathcal{X}$ defines a max-min fair bandwidth allocation and vise versa.

Unfortunately, Theorem 5 is not satisfied in the case of a single association, as we illustrate in Example 5. However, by using approximation algorithm we can provide an approximated solution to these NP-hard problems by rounding the calculated fractional solutions, as described in Section IV.

Example 5: Consider the wireless system described in Example 3. As mentioned above, Fig. 3(c) presents the min-max load balanced association $\mathcal{X}$. Its load vector is $\vec{Y} = \{1, 1, 1/2\}$ and its corresponding NBV is $\vec{B} = \{1, 1, 1, 2, 2\}$. However, the association $\mathcal{X}'$ presented in Fig. 4 has the same load vector while its NBV vector is $\vec{B}' = \{1, 1, 1, 1, 2\}$. Observe that in both associations $\mathcal{X}$ and $\mathcal{X}'$, one of the two APs $\mathbf{b}, \mathbf{c}$ has a load 1 and the other has 1/2. However, in association $\mathcal{X}$ only two users are associated with each one of these two APs, while in association $\mathcal{X}'$ three users are associated with AP $\mathbf{b}$ whose load is 1 and only one user is associated with AP $\mathbf{c}$ whose load is 1/2. This disparity leads to the suboptimality of association $\mathcal{X}'$. □

IV. Association Control Algorithms

In this section we present our algorithms that give approximate solutions to the integral max-min fair bandwidth allocation for greedy users. This is a challenging problem, as even identifying the users in the bottleneck fairness group and finding their normalized bandwidth is NP-hard. From Definition 2 and (1) follows that the minimal normalized bandwidth allocation is maximized when the maximal load on the APs is minimized, i.e., when the load on the APs is balanced. Our load balancing problem is actually an extension of the *scheduling unrelated parallel machines* problem [21], [28]. For this problem, Lenstra, Shmoys and Tardos, in [21], proved that for any positive $\epsilon < 1/2$ there is no polynomial-time $(1 + \epsilon)$-approximation algorithm exists, unless $P = NP$. Moreover, in [21] and [28], they gave a polynomial-time 2-approximation algorithms, which is currently the best known approximation ratio achieved in polynomial time. However, unlike the solutions given in [21], [28] that balance the load on the most loaded machines, our solution seeks for a complete min-max load balanced association.

We consider three different settings. We provide a 2-approximation algorithm for unweighted users, a 3-approximation algorithm for weighted users and an optimal solution for fair time allocation.

A. ρ^*-Approximation With Threshold

Intuitively, we would like to guarantee to each user a bandwidth of at least $1/\rho$ of the bandwidth that it receives in the optimal integral solution, for a constant $\rho \geq 1$. However, due to the unbounded integrality gap, it is impossible to provide this type of aQpproximation [20], as we demonstrate below. Let y_a^{int} and y_a^{frac} be the load on a given AP $a \in A$ in the optimal integral and fractional solutions, respectively. We show that there is neither upper nor lower constant bounds for the ratio $y_a^{\text{int}}/y_a^{\text{frac}}$.

Example 6: Consider a wireless network with two APs $\{a, b\}$ and two users {1,2}, where $r_{a,1} = r_{b,1} = c$ and $r_{a,2} = r_{b,2} = c/(2 \cdot c - 1)$ for a given constant $c > 1$. In the optimal fractional solution, the load on each AP is $y_a^{\text{frac}} = y_b^{\text{frac}} = 1/2 \cdot (1/c + (2c-1)/c) = 1$. However, in any integral solution, one AP, let say a, experiences a load of $y_a^{\text{int}} = 1/c$ while the other has a load of $y_b^{\text{int}} = (2c-1)/c$. Consequently, the ratio $y_a^{\text{int}}/y_a^{\text{frac}} = 1/c$ and it cannot be lower bounded by any constant. □

Example 6 demonstrates the difficulty to provide guarantees that are comparable with the integral solution. Accordingly, our guarantees are relative to an optimal fractional solution. Recall that the NBV of the latter has the same or higher lexicographical value than the NBV of the optimal integral solution. Thus, the fractional solution is at least as fair as an integral one. In fact, the optimal fractional solution is the fairest among all feasible allocations.

Example 7 (from [30]): Consider a wireless network with m APs, denoted by A, and a single user u, and let $r_{a,u} = 1$ for each $a \in A$. Clearly, in the fractional solution the load of u is equally divided among all the APs and thus for each $a \in A$, it follows that $y_a^{\text{frac}} = 1/m$. However, in the integral solution user u is associated with a single AP, lets say a, and the load of this AP is $y_a^{\text{int}} = 1$. Thus, the ratio between y_a^{int} and y_a^{frac} is m and it cannot be upper bounded by any constant. □

This obstacle occurs since the fractional load is smaller than the load induced by a single user on any AP. Since, our practical goal is to reduce the load of highly loaded APs, there is no need to balance the load of APs with load below a certain threshold T. To this end, we select T to be the maximal load that a user may generate on an AP as formulated in

$$T = \max_{\{u,a \mid u \in U \wedge a \in A \wedge r_{a,u} > 0\}} \max\left\{\frac{w_u}{r_{a,u}}, \frac{w_u}{R_a}\right\}. \quad (2)$$

Recall that T is indeed a very small value and in practical 802.11 networks $T \leq 1$ s/Mb. In light of these difficulties, we now formulate load and bandwidth guarantees that we provide in our solutions.

Definition 4: Let $\mathcal{X}^*$ be a fractional min-max load balances association and let y_a^* be the load of each AP $a \in A$. Then, a ρ^* *min-max load balanced approximation with threshold* T is an integral association $\mathcal{X}$ such that the load y_a of each AP $a \in A$ satisfies $y_a \leq \rho \cdot \max\{y_a^*, T\}$.

Alg Integral_Load_Balancing(A, U)
  $\mathcal{X}^{frac} \leftarrow Fractional\_Load\_Balancing(A, U)$
  $\mathcal{X}^{int} \leftarrow Rounding(\mathcal{X}^{frac})$
  return $\mathcal{X}^{int}$
end

Fig. 5. A formal description of the integral load balancing algorithm.

Definition 5: Let $\mathcal{X}^*$ be a fractional max-min fair association, and let $\bar{b}_u^*$ be its normalized bandwidth allocation to user $u \in U$. Then, a ρ^* *max-min fairness approximation with threshold* T is an integral association $\mathcal{X}$ such that the normalized bandwidth $\bar{b}_u$ of each user $u \in U$ satisfies $\bar{b}_u \geq (1/\rho) \cdot \min\{\bar{b}_u^*, 1/T\}$.

B. Scheme Overview

We now present our *integral load balancing algorithm*. The algorithm comprises two steps. Initially, it calculates the optimal fractional association i.e., the min-max load balanced fractional association. From Theorem 5, it follows that this association is also a max-min fair fractional allocation. Then, the algorithm utilizes the rounding method of Shmoys and Tardos [28] to obtain an approximate max-min fair integral association. A formal description of the algorithm is provided in Fig. 5.

1) The Fractional Load Balancing Algorithm: Our algorithm results from the observations made in Section III. More specific, let $\mathcal{X}$ be a max-min load balanced fractional association. According to Corollary 1, $\mathcal{X}$ partitions the APs and the users into load groups $\{L_k\}$ and corresponding fairness groups $\{F_k\}$, such that the APs in a load group L_k are associated only with the users in a fairness group F_k and vise versa. Moreover, all APs in a given load group L_k have the same load y_k and the corresponding users in the fairness group F_k experience a normalized bandwidth allocation of $1/y_k$.

Based on these observations, we present an iterative algorithm, referred to as the *fractional load balancing algorithm*. The algorithm calculates the load groups and their corresponding load values. For each load group, it also infers the users that are associated with the APs of this load group. To ease our presentation, lets assume that the load groups are enumerated in decreasing order according to their loads y_k. Thus, the APs in the group L_1 are the ones with the maximal load according to the association $\mathcal{X}$. We refer to the group L_1 as the *bottleneck load group* and the set F_1 of their associated users as the *bottleneck fairness group*. Moreover, load y_1 on the APs in L_1 is termed as the *bottleneck load* and it is denoted by $\tilde{Y}$.

Initially, the iterative algorithm assumes a system that contains all the APs and the users. At each iteration, the algorithm invokes the *bottleneck-group detection routine* to calculate the bottleneck load group and the corresponding fairness group. Then, it updates the fractional solution accordingly. Before proceeding to the next iteration, the algorithm removes the bottleneck load and fairness groups from the system. Note that in the new iteration the load group with the succeeding index becomes the bottleneck group. A formal description of the algorithm is given in Fig. 6.

Now, we turn to present the bottleneck-group detection routine. In this routine, we denote by $\tilde{L}$ and $\tilde{F}$ the load and

```
Alg Fractional_Load_Balancing(A, U)
    Initialize 𝒳
    k ← 1
    while (U ≠ ∅) do
        {L_k, F_k 𝒳_k} ← bottleneck_detection(A, U)
        Update 𝒳 with the association 𝒳_k.
        A ← A − L_k
        U ← U − F_k
        k ← k + 1
    end of while
    return 𝒳
end
```

Fig. 6. A formal description of the fractional load balancing algorithm.

fairness bottleneck group respectively. This routine consists of three steps. In the *first step*, we calculate the optimal bottleneck load value $\tilde{Y}$, that upper bounds the load y_a of every AP $a \in A$ in any min-max load balancing association. To infer its value, we utilize a linear program, denoted as $\mathbf{LP1}$, that calculates a feasible association $\mathcal{X}$, which also minimizes the maximal load on all the APs over both their wireless and wired channels

$$\begin{array}{ll}\mathbf{LP1}: & \min \tilde{Y} \\ \text{subject to:} & \\ \forall a \in A: & \sum_{u \in U} (w_u \cdot x_{a,u})/r_{a,u} \leq \tilde{Y} \\ \forall a \in A: & \sum_{u \in U} (w_u \cdot x_{a,u})/R_a \leq \tilde{Y} \\ \forall u \in U: & \sum_{a \in A} x_{a,u} = 1 \\ \forall u \in U, \forall a \in A: & x_{a,u} \in [0,1].\end{array}$$

Note that $\mathbf{LP1}$ minimizes the maximal load on all the APs. Consequently, the calculated association $\mathcal{X}$ ensures that the load on each AP in the bottleneck load group $\tilde{L}$ is exactly $\tilde{Y}$ and it also specifies the association of the APs in $\tilde{L}$ with the corresponding users in $\tilde{F}$. However, $\mathcal{X}$ does not optimize the load on the other APs, which may be as high as $\tilde{Y}$. We observe that, in the worst case, $\mathbf{LP1}$ may calculate a bad association such that the load on *all* the APs is $\tilde{Y}$ although the optimal association contains several load groups with lower loads, as illustrated in Example 8.

Example 8: Consider the wireless system described in Example 3 and the association presented in Fig. 3(b). This association induces a load of $\tilde{Y} = 1$ on all the APs. However, from Example 4 we know that a min-max fair allocation generates a load of 3/4 on AP $\mathbf{b}$ and $\mathbf{c}$ and accordingly the allocated bandwidth to each of the associated user 2,3,4,5 is 4/3. □

Such association is very deceptive, since it gives the impression that all the APs are included in the bottleneck load group. Therefore, we have developed a method to separate the APs in the bottleneck load group $\tilde{L}$ from the rest of the APs. In the *second step*, we use an auxiliary linear program, $\mathbf{LP2}$, which enables us to identify whether some APs are not in $\tilde{L}$ or whether $\tilde{L}$ comprises all the APs. $\mathbf{LP2}$ is based on Property 1, proved in Appendix VII.

Property 1: The bottleneck load group $\tilde{L}$ contains all the APs if there is *no* feasible association such that

1) Every AP has a load at most $\tilde{Y}$ and
2) Some APs have load strictly less than $\tilde{Y}$.

$\mathbf{LP2}$ looks for an association $\mathcal{X}$ that minimizes the overall load on all the APs subject to the constraint that the load on each AP is no higher than $\tilde{Y}$

$$\begin{array}{ll}\mathbf{LP2}: & \min \sum_{a \in A} y_a \\ \text{subject to:} & \\ \forall a \in A: & y_a \leq \tilde{Y} \\ \forall a \in A: & \sum_{u \in U} (w_u \cdot x_{a,u})/r_{a,u} \leq y_a \\ \forall a \in A: & \sum_{u \in U} (w_u \cdot x_{a,u})/R_a \leq y_a \\ \forall u \in U: & \sum_{a \in A} x_{a,u} = 1 \\ \forall u \in U, \forall a \in A: & x_{a,u} \in [0,1]\end{array}$$

Clearly, if the bottleneck load groups do not comprise all the APs then $\mathbf{LP2}$ should find an association where some APs have load strictly less than $\tilde{Y}$ and these APs are not included in $\tilde{L}$. However, $\mathbf{LP2}$ does not specify the APs that are included in $\tilde{L}$, as APs with loads equal to $\tilde{Y}$ are not necessarily included in $\tilde{L}$, as we illustrate in Example 9 below. Consequently, in the *third step*, we introduce a method to separate $\tilde{L}$ from the other APs based on the results given in Definition 3; The load of each AP $a \notin \tilde{L}$, $y_a = \tilde{Y}$, can be reduced by shifting the association of some of its associated users to less loaded APs.

Consider the association $\mathcal{X}$ determined by $\mathbf{LP2}$. Initially, we build a directed graph $G = (V, E)$ that each node $a \in V$ represents an AP in A, and there is an edge $(a, b) \in E$ if AP a can shift some load to AP b. In other words, there exists a user $u \in U$ such that $x_{a,u} > 0$ and $r_{b,u} > 0$. Note that the graph $G = (V, E)$ represents paths in which loads may be shifted. The method colors each node either white or black, where *white* represents APs not in $\tilde{L}$ and *black* indicates APs that *may* be included in the bottleneck group. Thus, the initial color of each node with load $\tilde{Y}$ is black, while the other nodes are colored white. Now, as long as there is an edge $(a, b) \in E$ such that node a is black and node b is white, we color node a white. At the end of this iterative process, the bottleneck load group $\tilde{L}$ comprises all the APs that are colored black and their associated users $\tilde{F}$ are determined by the association $\mathcal{X}$ calculated by $\mathbf{LP1}$ (or $\mathbf{LP2}$). Finally, the bottleneck-group detection routine returns the sets $\tilde{L}$, $\tilde{F}$ and their corresponding user-AP association $\tilde{\mathcal{X}}$. A formal description of this routine is given in Fig. 7 and an example of its execution is provided in Example 9.

Example 9: Consider the wireless system described in Example 3. In this case, a possible association $\mathcal{X}$ calculated by $\mathbf{LP2}$ is the one depicted in Fig. 8(a). Fig. 8(b) represents the calculated graph $G = (V, E)$ and the nodes' initial colors. Recall that $y_{\mathbf{a}} = y_{\mathbf{c}} = 1$ and $y_{\mathbf{b}} = 1/2$. Moreover, some load of user 2 or 3 can be shift from AP $\mathbf{b}$ to APs $\mathbf{c}$ or $\mathbf{a}$, which is indicated by the edges $(\mathbf{b}, \mathbf{c})$ and $(\mathbf{b}, \mathbf{a})$, and some load of user 4 or 5 can be shift from AP $\mathbf{c}$ to AP $\mathbf{b}$, which is indicated by

Routine bottleneck_detection(A, U)
  Use $\mathbf{LP1}$ to calculate $\tilde{Y}$.
  Use $\mathbf{LP2}$ to calculate an association $\mathcal{X}$.
  Construct a graph $G = (V, E)$.
  Color each AP a black if $y_a = \tilde{Y}$.
  Color each AP a white if $y_a < \tilde{Y}$.
  while exist $(a, b) \in E$ and a is black and b is white **do**
    Color AP a white.
  end while
  $\tilde{L} \leftarrow \{a | a \textit{ is colored black}\}$
  $\tilde{F} \leftarrow \{u | \exists x_{a,u} > 0 \wedge a \in \tilde{L}\}$
  $\tilde{\mathcal{X}} \leftarrow$ the association of $\tilde{F}$ and $\tilde{L}$.
  Return $\{\tilde{L}, \tilde{F}, \tilde{\mathcal{X}}\}$
end

Fig. 7. A formal description of the bottleneck-group detection routine.

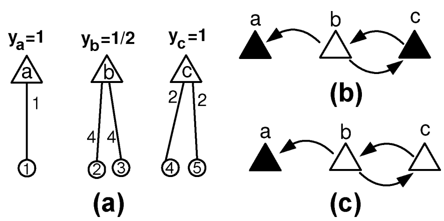


Fig. 8. Examples of an execution of the bottleneck-groups detection routine. (a) A possible association calculated by LP2. (b) The graph G(V,E) and the nodes' initial colors. (c) The nodes' final colors.

the edge $(\mathbf{c}, \mathbf{b})$. In the following, our routine colors AP $\mathbf{c}$ with white and ends the coloring iterations. Consequently, the computed groups are $\tilde{L} = \{\mathbf{a}\}$ and $\tilde{F} = \{1\}$, which are indeed the bottleneck groups. □

Theorem 6: The load balancing algorithm calculates a min-max load balanced association in the case that users are allowed to have fractional associations with APs.

Theorem 6 is proven in Appendix VII.

2) The Rounding Method: For the sake of completeness, we provide a short description of the rounding method of Shmoys and Tardos [28]. This description is tailored for unweighted greedy users but with minor modifications it can address weighted users, as we explain in the following subsection. Consider a fractional association $\mathcal{X}$ and for each AP $a \in A$ let $S_a = \lceil \sum_{u \in U} x_{a,u} \rceil$. Initially, the rounding method constructs a bipartite graph $G'(\mathcal{X}) = (U, V, E)$. Each node u in the set U of the bipartite graph represents a user u in U. The set V contains S_a nodes for each AP $a \in A$ denoted by $\{v_{a,1}, v_{a,2}, \cdots, v_{a,S_a}\}$. The graph edges are determined by the following process. For each AP $a \in A$, the users U_a are sorted according to a given *sorting criterion*. In the case of unweighted greedy users, the users in U_a are sorted in nondecreasing wireless bit rate $r_{a,u}$ and they are renamed according to this order, $\{u_1, u_2, \cdots, u_{|U_a|}\}$. Moreover, let $C(a, u_j) = \sum_{i=1}^{j} x_{a,u_i}$. For each AP a, we divide the users in U_a into S_a groups, denoted by $Q_{a,s}$ where $1 \leq s \leq S_a$, according to their $C(a, u_j)$ values. Each group $Q_{a,s}$ contains all the users u_j such that $s - 1 < C(a, u_j) \leq s$ or $s - 1 \leq C(a, u_{j-1}) < s$. A user that is included in two groups is referred as *border node*. The edges E of the graph represent user-AP association. Thus, for each AP a and every integer $s \in S_a$ node $v_{a,s}$ is connected to each user u_j in $Q_{a,s}$. Such bipartite graph is given in Example 10. After constructing the graph G', the rounding method looks for a maximal matching [31] from each user to one of the nodes $v_{a,s} \in V$. Since the association $\mathcal{X}$ specifies a fractional matching such maximal matching exists (more details are provided in [28]) and it determines the integral association of the users.

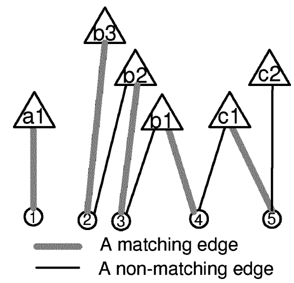


Fig. 9. Examples of the graph G' and a matching.

Example 10: Consider the wireless system described in Example 3 and the fractional max-min fair association depicted in Fig. 3(d). In this association $x_{\mathbf{b},4} = x_{\mathbf{c},4} = 1/2$ and its NBV is $\vec{B} = \{1, 4/3, 4/3, 4/3, 4/3\}$. Fig. 9 presents the graph G' calculated by the rounding method and a corresponding matching. Consequently, the obtained load vector $\vec{Y} = \{1, 1, 1/2\}$ and the corresponding NBV is $\vec{B} = \{1, 1, 1, 1, 2\}$. The latter is not the optimal max-min fair association. However, the bandwidth of each user u is at least half of its bandwidth in the fraction association. □

C. Analysis of the Unweighted Case

We now prove the approximation ratio of our algorithm for the case of unweighted greedy users. We start with a useful property of the rounding method. We assign to each edge e of G' a weight, $x'(e)$, termed the *association weight*, that represents the fractional association of the corresponding user and AP. More specifically, consider an edge $e = (v_{a,s}, u) \in E$ indicating that user u is associated with AP a. If user u is a nonborder node then it is included only in the set $Q_{a,s}$ and we assign $x'(v_{a,s}, u) = x_{a,u}$. Otherwise, user u is included in the sets $Q_{a,s-1}$ and $Q_{a,s}$ and we partition the association $x_{a,u}$ with the two edges $(v_{a,s-1}, u)$ and $(v_{a,s}, u)$, such that $x'(v_{a,s}, u) = C(a, u) - s + 1$ and $x'(v_{a,s-1}, u) = x_{a,u} - x'(v_{a,s}, u)$. This assignment ensures the following property.

Property 2: Consider an AP $a \in A$ and a set $Q_{a,s}$, where s is an integer between 1 and S_a. Then, for any $s < S_a$, it follows that $\sum_{u \in Q_{a,s}} x'(v_{a,s}, u) = 1$ and $\sum_{u \in Q_{a,S_q}} x'(v_{a,S_a}, u) \leq 1$.

Consider a node $v_{a,s} \in V$. We define its fractional wireless load as $y^{\mathrm{frac},w}(v_{a,s}) = \sum_{u \in Q_{a,s}} x'(v_{a,s}, u)/r_{a,u}$. Moreover, suppose that node $v_{a,s}$ is associated to user $u \in Q_{a,s}$ in the calculated matching. We define its integral wireless load as $y^{\mathrm{int},w}(v_{a,s}) = 1/r_{a,u}$. Similarly, we define the fractional and integral infrastructure load of node $v_{a,s}$ as $y^{\mathrm{frac},i}(v_{a,s}) = \sum_{u \in Q_{a,s}} x'(v_{a,s}, u)/R_a$ and $y^{\mathrm{int},i}(v_{a,s}) = 1/R_{a,u}$. Consequently:

Lemma 1: Consider a node $v_{a,s} \in V$ such that $s > 1$. Then, $y^{\mathrm{int},w}(v_{a,s}) \leq y^{\mathrm{frac},w}(v_{a,s-1})$ and $y^{\mathrm{int},i}(v_{a,s}) \leq y^{\mathrm{frac},i}(v_{a,s-1})$.

Proof: This lemma results directly from the selected sorting criterion and we first prove it for wireless channel. For each user $u \in Q_{a,s}$, $s > 1$ satisfied that $r_{a,u} \geq r_{a,u'}$ for every user $u' \in Q_{a,s-1}$. This is also true for the user $u^* \in Q_{a,s}$ that is matched with node $v_{a,s}$. Thus

$$\begin{aligned} y^{\text{frac},w}(v_{a,s-1}) &= \sum_{u' \in Q_{a,s-1}} \frac{x'(v_{a,s}, u')}{r_{a,u'}} \\ &\geq \sum_{u' \in Q_{a,s-1}} \frac{x'(v_{a,s}, u')}{r_{a,u^*}} \\ &= \frac{1}{r_{a,u^*}} = y^{\text{int},w}(v_{a,s}). \end{aligned}$$

We now consider the backhaul link. Recall that all the users pose the same load, $1/R_a$, on the backhaul link. Therefore, independent of the user order, for each node $v_{a,s} \in V$ such that $s < S_a$, it follows that $y^{\text{frac},i}(v_{a,s}) = 1/R_a$ and for any node $v_{a,S_a} \in V$, it follows that $y^{\text{frac},i}(v_{a,S_a}) \leq 1/R_a$. Consequently, $y^{\text{int},i}(v_{a,s}) \leq y^{\text{frac},i}(v_{a,s-1})$. □

Theorem 7: The association $\mathcal{X}$ calculated by integral load balancing algorithm ensures 2* max-min fairness approximation with threshold T, defined by (2).

Proof: First, we prove for each AP $a \in A$ that $y_a^{\text{int}} \leq y_a^{\text{frac}} + T$. We prove this property for the wireless link. The proof for the backhaul link is similar. From Lemma 1 and the definition of T follows:

$$\begin{aligned} y_a^{\text{int},w} &= \sum_{s \in [1..S_a]} y^{\text{int},w}(v_{a,s}) \\ &\leq T + \sum_{s \in [1..(S_a-1)]} y^{\text{frac},w}(v_{a,s}) \leq T + y_a^{\text{frac},w}. \end{aligned}$$

Consequently, $y_a^{\text{int}} \leq T + y_a^{\text{frac}}$. In the sequel we consider two cases:

Case I: Suppose that $y_a^{\text{frac}} \geq T$. Thus $y_a^{\text{int}} \leq 2 \cdot y_a^{\text{frac}}$. From Theorems 4 and 6, it results that bandwidth allocation of each user u associated with AP a in the integral solution is $b_u^{\text{int}} = 1/y_a^{\text{int}} \geq 1/(2 \cdot y_a^{\text{frac}}) = b_u^{\text{frac}}/2$.

Case II: Suppose that $y_a^{\text{frac}} < T$. Thus $y_a^{\text{int}} \leq 2 \cdot T$. Accordingly, each user u that is associated with AP a in the integral solution experiences a bandwidth $b_u^{\text{int}} = 1/y_a^{\text{int}} > 1/(2 \cdot T)$, and this complete our proof. □

D. Weighted Users

We turn to describe our integral load balancing algorithm for weighted users. This algorithm is similar to the one described in Section IV-B with different sorting criterion. We observed that in weighted instances, the calculated fractional solution $\mathcal{X}^{\text{frac}}$ does not satisfy Lemma 1. This prevents from us to providing 2* max-main fairness approximation. However, by using a different sorting criterion, our algorithm ensures 3* approximation. For our needs, we define the *joined load* of user u on AP a as

$$J_{a,u} = \frac{x_{a,u} \cdot w_u}{r_{a,u}} + \frac{x_{a,u} \cdot w_u}{R_{a,u}}$$

The joined load may be either fractional or integral. For a given AP a, the algorithm sorts the users U_a in decreasing order of their joined loads, $J_{a,u}$. This order determines the manner in which the users U_a are divided into groups $\{Q_{a,s}\}$. The rest of the rounding method remains the same.

We turn to calculate the approximation ratio of the algorithm with same threshold T defined in (2). Consider a node $v_{a,s} \in V$ we define its fractional joined load $J^{\text{frac}}(v_{a,s}) = \sum_{u \in Q_{a,s}} x'(v_{a,s}, u) \cdot J_{a,u}$. Now, suppose that node $v_{a,s}$ is associated to user $u \in Q_{a,s}$ in the integral solution. Thus, its integral joined load is $J^{\text{int}}(v_{a,s}) = J_{a,u}$. Note that the fractional and integral joined loads of AP $a \in A$ satisfy

$$J_a^{\text{frac}} = y_a^{\text{frac},w} + y_a^{\text{frac},i} = \sum_{u \in U_a} J_{a,u}^{\text{frac}} = \sum_{s=1}^{S_a} J^{\text{frac}}(v_{a,s}).$$

Similarly,

$$J_a^{\text{int}} = y_a^{\text{int},w} + y_a^{\text{int},i} = \sum_{u \in U_a} J_{a,u}^{\text{int}} = \sum_{s=1}^{S_a} J^{\text{int}}(v_{a,s}).$$

Lemma 2: Consider a node $v_{a,s} \in V$ such that $s > 1$. Then, $J^{\text{int}}(v_{a,s}) \leq J^{\text{frac}}(v_{a,s-1})$.

Proof: This proof is similar to the proof of Lemma 1 and it is direct result from the definition of joined load. □

Lemma 3: Consider an AP $a \in A$ then $J_a^{\text{frac}} \leq 2 \cdot y_a^{\text{frac}}$

Proof: By definition, $J_a^{\text{frac}} = y_a^{\text{frac},w} + y_a^{\text{frac},i} \leq 2 \cdot \max\{y_a^{\text{frac},w}, y_a^{\text{frac},i}\} = 2 \cdot y_a^{\text{frac}}$. □

Theorem 8: The association $\mathcal{X}$ calculated by integral load balancing algorithm ensures 3* max-min fairness approximation with threshold T, defined by (2).

Proof: First, we prove that for each AP $a \in A$ follows that $y_a^{\text{int}} \leq 2 \cdot y_a^{\text{frac}} + T$. From Lemma 2 and the definition of T, it follows:

$$\begin{aligned} y_a^{\text{int}} &= \max\left\{\sum_{s=1}^{S_a} y^{\text{int},w}(v_{a,s}), \sum_{s=1}^{S_a} y^{\text{int},i}(v_{a,s})\right\} \\ &\leq \sum_{s=1}^{S_a} J^{\text{int}}(v_{a,s}) \leq T + \sum_{s=1}^{S_a-1} J^{\text{frac}}(v_{a,s}) \leq T + J_a^{\text{frac}}. \end{aligned}$$

From Lemma 3 results that $y_a^{\text{int}} \leq T + 2 \cdot y_a^{\text{frac}}$. In the sequel we consider two cases.

Case I: Suppose that $y_a^{\text{frac}} \geq T$. Thus, $y_a^{\text{int}} \leq 3 \cdot y_a^{\text{frac}}$. From Theorems 5 and 6, it results that the normalized bandwidth $\bar{b}_u^{\text{int}}$ allocated to user u associated with AP a in the integral solution is $\bar{b}_u^{\text{int}} = 1/y_a^{\text{int}} \geq 1/(3 \cdot y_a^{\text{frac}}) = \bar{b}_u^{\text{frac}}/3$.

Case II: Suppose that $y_a^{\text{frac}} < T$. Thus $y_a^{\text{int}} \leq 3 \cdot T$. Accordingly, each user u that is associated with AP a in the integral solution experiences a normalized bandwidth $\bar{b}_u^{\text{int}} = 1/y_a^{\text{int}} \geq 1/(3 \cdot T)$, and this complete our proof. □

E. Time Fairness

We now introduce our results for max-min time fairness. Time fairness attempts to provide a fair service time to the users regardless of the effective bit rates, $r_{a,u}$ and R_a, that they experience. Consequently, it enables us to trade off throughput between fairness and system throughput, while not starving any user with low bit-rate, $r_{a,u}$. Informally, a service time allocation is called *max-min time fair* if there is no way to increase the service time of a user without decreasing the service time of another user with the same or less service time. Usually, there can be multiple time fairness associations that satisfy the min-max time fairness requirement. Consequently, time fairness requirement is, typically, coupled with a secondary objective. For instance, a time fair association that also maximizes the system overall throughput or one the maximizes the minimal bandwidth allocated to each user. Due to space limitation we do not consider these complicated variations of time fairness and we leave these challenges to future work. In this study, we address the fundamental max-min time fairness problem as described above. Such fairness is relevant, for instance, when the system bottlenecks are the backhaul links and all these links have the same bit rate, R. In such instance, a max-min time fairness solution also guarantees max-min bandwidth fairness.

To achieve this goal, we use the scheme presented in Section IV-B with the following modifications. First, for each user $u \in U$ and AP $a \in A$, we set their effective bit rates $r_{a,u}$ and R_a to 1 and we utilize the unweighted variant for obtaining a fractional solution. Then, after calculating the bipartite graph $G'(\mathcal{X}) = (U, V, E)$, we assigned a cost $c(v_{a,s}, u) = s$ to each edge $(v_{a,s}, u) \in E$. Finally, the integral association is determined by the minimal cost maximal matching [31] of the graph G'.

Theorem 9: The time fairness algorithm calculates the optimal max-min time fairness association.

Proof: From Theorem 6, it follows that our scheme finds the optimal fractional solution. Thus, to complete the proof it is sufficient to prove that the algorithm finds the optimal integral association for every fairness group $F_k \subseteq U$ and its corresponding load group $L_k \subseteq A$ with load y_k of the fractional solution. Clearly, in this case the load of each AP $a \in L_k$ is $y_k = y_a = \sum_{u \in U_a} x_{a,u}$. Thus, from the definition of S_a in Section IV-B-2, it results that $S_a - 1 < y_a \leq S_a$ for every AP $a \in L_k$. Since all APs in L_k have the same S_a we denote it by S_k and the number of users that are associated with any AP $a \in L_k$ is at most S_k. We consider two cases.

Case I: $y_k = S_k$. Thus, each AP in L_k is associated with exactly S_k users and this guarantees the required time fairness.

Case II: $y_k < S_k$. Consequently, some APs are associated with fewer than S_k users. Note that we are addressing now a load conserving system, i.e., in any possible association of the user in F_k associated with the APs in L_k, the total load on all the APs is $y_k \cdot |L_k| = |F_k|$. Since, our algorithm seeks for minimal cost matching no AP is associated with fewer than $S_k - 1$ users. From this, it results that exactly $(S_k - y_k) \cdot |L_k|$ APs are associated with $S_k - 1$ users and others are associated with S_k users. This is a max-min time fair association and this completes our proof. □

V. Online Integral-Association

In this section, we present an algorithm that deals with dynamic user arrivals and departures. Clearly, a repeated execution of the offline algorithm each time a user arrives or departs may cause frequent association changes that disrupt existing sessions. To avoid this, we propose a strategy that enables us to strike a balance between the frequency of the association changes and the optimality of the network operation in terms of load balancing. For this propose we use two configuration parameters; *time threshold*, τ, and *load threshold* Δ. We rerun our offline algorithm if either of the following two conditions hold.

1) The time elapsed since our last offline optimization is more than the time threshold τ.
2) The current bottleneck load, i.e., the maximal load among all APs, is Δ more than the bottleneck load obtained by the last execution of the offline algorithm.

After rerunning the algorithm, each user who needs to change association can be done between its session arrivals to avoid disruption of its ongoing sessions. Our algorithm is illustrated in Fig. 10.

Alg Online_Load_Balancing(A, U, u)
  if (elapsed time from last offline optimization $> \tau$) **then**
    Integral_Load_Balancing(A, $U \cup u$)
    $y_{offline} \leftarrow \max_{a \in A} y_a$
  else
    $a \leftarrow$ AlgorithmByAAFPW(A, u)
    $y_{online} \leftarrow \max_{a \in A} y_a$
    if ($y_{online} - y_{offline} > \Delta$) **then**
      Integral_Load_Balancing(A, $U \cup u$)
    else
      assign u to AP a
end

Fig. 10. A formal description of the online load balancing algorithm.

Between two offline optimization occurrences, we need to associate users to APs as they arrive. We adapt the online algorithm of Aspnes *et al.* in [22], to achieve a $O(logn)$ approximation factor as compared to the offline optimal, where n is number of users in the system. We refer their algorithm as AlgorithmByAAFPW. All we need to change is to substitute the load in their algorithm by the integral load of the APs, y_a^{int}. In online user association, we need to address two conflicting factors. Intuitively, a user should be assigned to the less loaded APs that are within its transmission range. However, the data rate from the user to these APs can be very low which adds very high additional load to them. Therefore, a user should be assigned to an AP where it causes small additional load. To capture these two trade-offs, Aspnes *et al.* [22] define a function $b^{\tilde{l}_a}$ that is exponential in the load of an AP where $\tilde{l}_a = y_a^{\text{int}}/\Lambda$, $b \approx 2$ and $\Lambda \approx 1$. When a new user arrives, all possible user-AP association are evaluated. After the evaluation, the assignment that minimizes the increase of the function is selected. They show that, using certain potential functions, the highest load among all APs of the online algorithm is within $O(logn)$ factor of the highest load among all APs of the offline algorithm.

VI. Simulation Results

Via simulations, we compare the performance (in the context of max-min fairness) of our scheme with two popular heuristics, namely the Strongest-Signal-First(SSF) method and the Least-Loaded-First(LLF) method. The SSF method is the default user-AP association method in the 802.11 standard. The LLF method is a widely used load-balancing heuristic, in which a user chooses the least-loaded AP that he can reach. The simulation setting is as follows. We use a simple wireless channel model in which the user bit rate depends only on the distance to the AP. Adopting the values commonly advertised

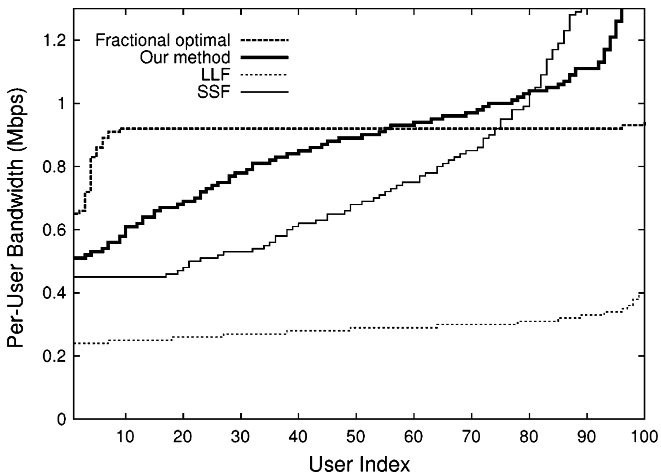


Fig. 11. Per-user bandwidth of 100 users.

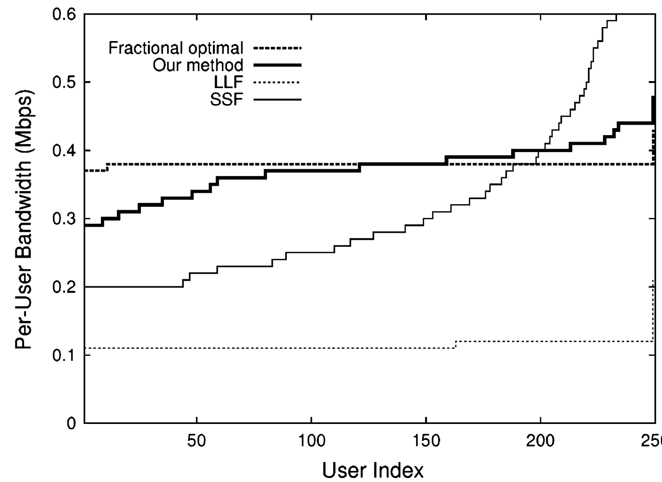


Fig. 12. Per-user bandwidth of 250 users.

by 802.11b vendors [8], [9], we assume that the bit rate of users within 50 meters from AP is 11 Mb/s, 5.5 Mb/s within 80 meters, 2 Mb/s within 120 meters, and 1 Mb/s within 150 meters, respectively. The maximum transmission range of an AP is 150 meters. The backhaul capacity is set to 10 Mb/s to emulate the Ethernet infrastructure. A total of 20 APs are located on a 5 by 4 grid, where the distance between two adjacent APs is set to 100 meters and we assume that an appropriate frequency planning was made. The number of users is either 100 to simulate a moderately loaded network or 250 to simulate a heavily loaded network.

Users are randomly positioned in a circle-shape hot-spot with the radius of 150 meters near the center of the 20 AP network. Notice that in this setup different users can reach different set of cells. Figs. 11 and 12 show the results with 100 and 250 users, respectively. The Y axis represents the per-user bandwidth and the X axis represents the user index. The users are sorted by their bandwidth in increasing order. The user locations are different at each run, and therefore the bandwidth of the user with the same x index actually indicates the average bandwidth of x-th lowest bandwidth user. Somewhat surprisingly, our method outperforms the two heuristics not only in terms of fairness but also in terms of total system throughput. For instance, in Fig. 11, the median per-user bandwidth value of our method is over 20% higher than that of SSF. The bandwidth values are obtained by averaging the results of 100 simulation runs. We also noticed that SSF outperforms LLF in terms of both max-min fairness and overall network throughput. This supports our early claim that a naive load-balancing algorithm may yield poor results. By comparing Fig. 11 and 12, we also conclude the gap between our method and the fractional optimal solution narrows as the number of users increases. It can be explained by the fact that the impact of each user in the integral association scheme decreases as the number of users increases. Thus, with an infinite number of users, the results of integral association and fractional association will converge.

We also simulated the online algorithm. To simulate the dynamic user departure/arrival (or the user mobility), at each time slot a certain portion of users are taken out and the same number of new users are injected. The result of the case that we replace

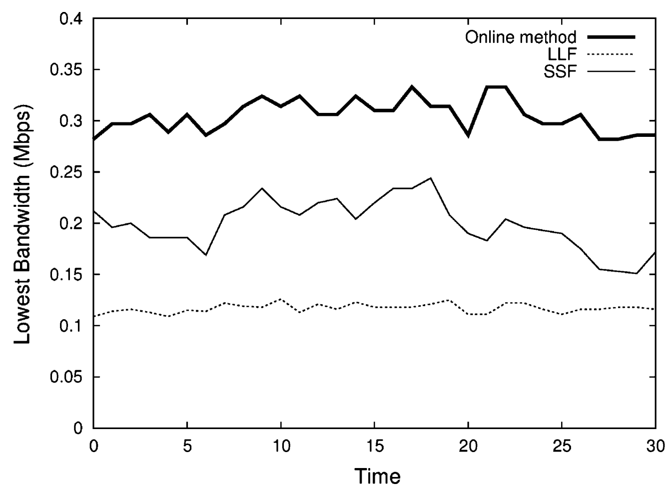


Fig. 13. Simulation result of the online case with 250 users.

20% of users at every time slot is shown in Fig. 13. The Y axis represents the lowest user bandwidth and the X axis represents the time. The offline algorithm is periodically invoked at every 15 time slots or when the bottleneck difference exceeds 25% (in presented case, the offline algorithm was invoked total 5 times). Note that the result is episodic, since it depicts the evolution of the system for one simulation run. Nevertheless, the presented result is very typical.

VII. CONCLUSION

As wireless LANs are deployed to cover larger areas and are increasingly relied on to carry important tasks, it is essential that they be managed in order to achieve desired system performance objectives. In this paper, we study the problem of providing fair service to users and balancing the load among APs. This goal is achieved by intelligently determining the user-AP association. We rigorously formulate this association control problem in the context of wireless LANs and present approximation algorithms that provide guarantees on the quality of the solution. Our simulations confirm that the proposed methods, indeed, achieve close to optimal load balancing and max-min fair bandwidth allocation, and significantly outperform popular heuristics. Moreover, we show that in some cases, by balancing the load on the APs the overall network throughput is increased. In the future, we intend to develop a practical management system based on the theoretical foundation presented in this study.

Appendix

A. *Proof Sketch of Theorem 5*

In the following we only prove that the min-max load balanced association determines a max-min fair bandwidth allocation. By similar arguments the other direction can be proven as well. Let $\mathcal{X}$ be a min-max load balanced association and let $\vec{B}$ be its normalized bandwidth vector. Lets assume, that $\mathcal{X}$ does not produce a max-min fair bandwidth allocation. Thus, there is an association $\mathcal{X}'$ that its normalized bandwidth vector $\vec{B}'$ has higher lexicographical value than $\vec{B}$. Let $\{F_k\}$, $\{F'_k\}$, $\{L_k\}$ and $\{L'_k\}$ be the fairness and the load groups of the associations $\mathcal{X}$ and $\mathcal{X}'$, respectively. We define an additional association, $\tilde{\mathcal{X}} = (\mathcal{X} + \mathcal{X}')/2$, i.e., for each AP a and user u, it follows $\tilde{x}_{a,u} = (x_{a,u} + x'_{a,u})/2$, and let $\{\tilde{F}_k\}$ and $\{\tilde{L}_k\}$ be its fairness and load groups, respectively. Let j be the lowest index such that $F_j \neq F'_j$ or $L_j \neq L'_j$. Recall, that for every index $i < j$ follows that $\tilde{F}_i = F'_i$ and $\tilde{\bar{b}}_i = \bar{b}'_i$. Since, $\mathcal{X}$ is min-max load balanced association, it follows that $y_j \leq y'_j$. Similarly, $\mathcal{X}'$ is max-min fair bandwidth association, thus, $\bar{b}_j \leq \bar{b}'_j$. As $y_j = 1/\bar{b}_j$ and $y'_j = 1/\bar{b}'_j$ we have $y_j = y'_j$ and $\bar{b}_j = \bar{b}'_j$. In the following we assume, without lost of generality, that $F_j \neq F'_j$, the case where $L_j \neq L'_j$ can be proven in similar way. We consider three cases:

Case I: $F_j \subset F'_j$: However, this contradicts the assumption $\mathcal{X}'$ is a max-min fair bandwidth association.

Case II: $F'_j \subset F_j$: Now suppose that $L_j \subset L'_j$, but in this case the set of APs L_j is sufficient to provide the bandwidth $\bar{b}'_j$ to all the users in the set F'_j. While, APs in the sets $L'_j - L_j$ can be used to increase the bandwidth allocation of other users with the same or higher bandwidth, which contradicts the assumption that $\mathcal{X}'$ is max-min fair bandwidth association. Consequently, it follows that $L_j \not\subset L'_j$, which implies that $L_j - L'_j \neq \emptyset$. Thus, the association $\tilde{\mathcal{X}}$, obviously, reduces the load from every AP $a \in L_j - L'_j$, without increasing the load of any AP with load y_j or more. This contradicts the assumption that $\mathcal{X}$ is a min-max load balanced association.

Case III: $F'_j - F_j \neq \emptyset$: In this case, the association $\tilde{\mathcal{X}}$ guarantees to each user $u \in F'_j - F_j$ a bandwidth $\tilde{\bar{b}}_u > \bar{b}_j$ without decreasing the bandwidth of any other user that has normalized bandwidth of $\bar{b}_j$ or less in $\mathcal{X}'$. This contradicts the assumption that $\mathcal{X}'$ is a max-min fair bandwidth association.

Consequently, we conclude that for every j, $L_j = L'_j$ and $F_j = F'_j$ and this completes our proof. □

B. *Proof Sketch of Theorem 6*

We start with some properties of the bottleneck-group detection routine.

Lemma 4: $\mathbf{LP1}$ infers the value of the bottleneck load $\tilde{Y}$ of any min-max load balanced association. Moreover, it calculates an association such that $\tilde{Y}$ upper bounds the load of each AP.

Proof: $\mathbf{LP1}$ seeks for an association $\mathcal{X}$ that minimizes $\tilde{Y}$. The first two conditions verify that $\tilde{Y}$ upper bounds the load of each AP, while, the other conditions ensure the $\mathcal{X}$ is a feasible association. □

Lemma 5: Let $\mathcal{X}$ be the association calculated by $\mathbf{LP2}$ for a giving bottleneck load value $\tilde{Y}$ as determined by $\mathbf{LP1}$. The bottleneck load group comprises all the APs if and only if the load on each AP is $\tilde{Y}$. Otherwise, there is at least one AP that its load is strictly less than $\tilde{Y}$.

Proof: From Lemma 4, it follows that the bottleneck load value is $\tilde{Y}$. Recall that $\mathbf{LP2}$ finds a feasible association $\mathcal{X}$ that minimizes the overall load with the constraint that the load of each AP is at most $\tilde{Y}$ (the latter is termed as the upper bound constraint). Consequently, if all the APs are included in $\tilde{L}$, then, by definition, the overall load of any such association calculated by $\mathbf{LP2}$ is $|A| \cdot \tilde{Y}$. Thus, there is no feasible association that satisfies the upper bound constraint and some APs have load strictly less then $\tilde{Y}$. On the other hand, if not all the APs are included in $\tilde{L}$, then there is an association whose overall load is strictly less than $|A| \cdot \tilde{Y}$. In such cases, $\mathbf{LP2}$ finds a feasible association such that the load of some APs is strictly less than $\tilde{Y}$. □

Lemma 6: Let $G = (V, E)$ be the graph that results from the association $\mathcal{X}$ calculated by $\mathbf{LP2}$ and consider the initial node colors. A given AP is included in $\tilde{L}$ if and only if its corresponding node in G, denoted by b, is colored black and there is no directed path in G from b to any white colored node.

Proof: Consider a black node b that is included in a directed path of black nodes $P = \{b = v_1, v_2, \cdots, v_k = a\}$ ended with a white node a. This means that the corresponding AP of node v_{k-1} can shift some load to AP represented by node a. Therefore, it can reduce its load without increasing the load of any AP with load $\tilde{Y}$. In an iterative manner, this process can be done for any node $v_i \in P$. Thus, the AP represented by node b will not be included in $\tilde{L}$.

We now prove the other direction. From Corollary 1, it follows that all the APs in $\tilde{F}$ have load $\tilde{Y}$, hence their corresponding nodes are colored black. In addition, the load of any AP $b \in \tilde{f}$ cannot be reduced by shifting some load to a nonbottleneck AP. Thus, there is no directed link in G between a node representing a bottleneck AP to a node representing a nonbottleneck AP. Consequently, nodes that represent APs in $\tilde{F}$ are not included in any directed path ending with a white node. □

Lemma 7: The bottleneck-group detection routine determines the load and the fairness bottleneck groups, $\tilde{L}$ and $\tilde{F}$, and their corresponding user-AP association in the fractional-association model.

Proof: From Lemma 4, it follows that $\mathbf{LP1}$ determines the bottleneck load value $\tilde{Y}$ and also calculates a feasible association that satisfies the upper bound constraint. From Lemmas 5 and 6, it follows that the routine separates the APs in $\tilde{L}$ from the other APs. Finally, from Corollary 1 the APs in $\mathcal{X}$ are associated only with the users in $\tilde{F}$. □

Proof of Theorem 6: From Lemma 7 and Corollary 1 results that at each iteration the load balancing algorithm detects the current load and fairness bottleneck groups, denoted as L_k and F_k, and their user-AP association. Thus, at each iteration, the algorithm reduces the size of the AP and user sets until a complete min-max load association is detected. □

REFERENCES

[1] Y. Bejerano, S.-J. Han, and L. Li, “Fairness and load balancing in wireless LANs using association control,” in *Proc. ACM MobiCom’04*, 2004, pp. 315–329.

[2] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, “Characterizing user behavior and network performance in a public wireless LAN,” in *Proc. ACM SIGMETRICS*, 2002, pp. 195–205.

[3] D. Kotz and K. Essien, “Analysis of a campus-wide wireless network,” in *Proc. ACM MobiCom’02*, 2002, pp. 107–118.

[4] M. Balazinska and P. Castro, “Characterizing mobility and network usage in a corporate wireless local-area network,” in *Proc. USENIX MobiSys*, 2003.

[5] I. Papanikos and M. Logothetis, “A study on dynamic load balance for IEEE 802.11b wireless LAN,” in *Proc. COMCON*, 2001.

[6] A. Balachandran, P. Bahl, and G. M. Voelker, “Hot-spot congestion relief and service guarantees in public-area wireless networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 1, pp. 59–59, 2002.

[7] T.-C. Tsai and C.-F. Lien, “IEEE 802.11 hot spot load balance and QoS-maintained seamless roaming,” in *Proc. National Computer Symp. (NCS)*, 2003.

[8] ORINOCO AP-600 Data Sheet, Proxim Wireless Networks, 2004.

[9] Data Sheet for Cisco Aironet 1200 Series, Cisco Systems Inc., 2004.

[10] I. Katzela and M. Nagshineh, “Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey,” *IEEE Pers. Commun.*, vol. 3, no. 3, pp. 10–31, Jun. 1996.

[11] S. Das, H. Viswanathan, and G. Rittenhouse, “Dynamic load balancing through coordinated scheduling in packet data systems,” in *Proc. IEEE INFOCOM*, 2003.

[12] B. Eklundh, “Channel utilization and blocking probability in a cellular mobile telephone system with directed retry,” *IEEE Trans. Commun.*, vol. 34, no. 4, pp. 329–337, 1986.

[13] T. P. Chu and S. R. Rappaport, “Overlapping coverage with reuse partitioning in cellular communication systems,” *IEEE Trans. Veh. Technol.*, vol. 46, no. 1, pp. 41–54, 1997.

[14] X. Lagrange and B. Jabbari, “Fairness in wireless microcellular networks,” *IEEE Trans. Veh. Technol.*, vol. 47, no. 2, pp. 472–479, 1998.

[15] I. Tinnirello and G. Bianchi, “A simulation study of load balancing algorithms in cellular packet networks,” in *Proc. ACM/IEEE MSWiM*, 2001, pp. 73–78.

[16] J. M. Jaffe, “Bottleneck flow control,” *IEEE Trans. Commun.*, vol. 29, pp. 954–962, 1981.

[17] Y. Afek, Y. Mansour, and Z. Ostfeld, “Convergence complexity of optimistic rate based flow control algorithms,” in *Proc. ACM STOC*, 1996, pp. 89–98.

[18] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[19] N. Megiddo, “Optimal flows in networks with multiple sources and sinks,” *Math. Program.*, vol. 7, pp. 97–107, 1974.

[20] J. M. Kleinberg, Y. Rabani, and E. Tardos, “Fairness in routing and load balancing,” in *Proc. IEEE FOCS*, 1999, pp. 568–578.

[21] J. K. Lenstra, D. B. Shmoys, and E. Tardos, “Approximation algorithms for scheduling unrelated parallel machines,” *Math. Program.*, vol. 46, pp. 259–271, 1990.

[22] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, “On-line load balancing with applications to machine scheduling and virtual circuit routing,” in *Proc. ACM STOC*, 1993, pp. 623–631.

[23] A. Goel, A. Meyerson, and S. Plotkin, “Approximate majorization and fair online load balancing,” in *Proc. SODA*, 2001, pp. 384–390, Society for Industrial and Applied Mathematics.

[24] Q. Ni, L. Romdhani, T. Turletti, and I. Aad, “QoS Issues and Enhancements for IEEE 802.11 Wireless LAN,” INRIA, France, Tech. Rep. RR-461, Nov. 2002 [Online]. Available: http://www.inria.fr/rrrt/rr-4612.html

[25] P. Ramanathan and P. Agrawal, “Adapting packet fair queueing algorithms to wireless networks,” in *Proc. ACM MobiCom’98*, Oct. 1998, pp. 1–9.

[26] S. Lu, T. Nandagopal, and V. Bharghavan, “A wireless fair service algorithm for packet cellular networks,” in *Proc. ACM MobiCom’98*, Oct. 1998, pp. 10–20.

[27] M. Buddhikot, G. Chandranmenon, S.-J. Han, Y.-W. Lee, S. Miller, and L. Salgarelli, “Integration of 802.11 and third-generation wireless data networks,” in *Proc. IEEE INFOCOM*, 2003.

[28] D. B. Shmoys and E. Tardos, “An approximation algorithm for the generalized assignment problem,” *Math. Program.*, vol. 62, no. 3, pp. 461–474, 1993.

[29] M. R. Garey and D S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman , 1979.

[30] V. Vazirani, *Approximation Algorithms*. New York: Springer-Verlag, 1999.

[31] L. Lovazs and M. D. Plummer, *Matching Theory*. Amsterdam, The Netherlands: North-Holland, 1986.

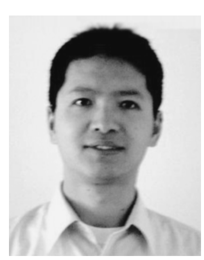
Yigal Bejerano (M’06) received the B.Sc. degree (*summa cum laude*) in computer engineering in 1991, the M.Sc. degree in computer science in 1995, and the Ph.D. degree in electrical engineering in 2000, from the Technion–Israel Institute of Technology, Haifa, Israel.

He is currently a Member of the Technical Staff at Bell Labs, Lucent Technologies, Murray Hill, NJ. His research interests are mainly management aspects of high-speed and wireless networks, including the areas of mobility management, network monitoring, topology discovery, quality of service (QoS) routing, wireless LAN and wireless mesh networks.

Dr. Bejerano is on the technical program committee (TPC) of the IEEE INFOCOM 2002–2006 conferences.

Seung-Jae Han (M’06) received the B.S. and M.S. degrees in computer science and engineering from the Seoul National University, Seoul, Korea, in 1989 and 1991, respectively. In 1998, he received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor.

From 1999 to 2005, he was a Member of Technical Staff in Bell Labs, Lucent Technologies, Murray Hill, NJ. He is currently an Associate Professor in the Department of Computer Science, Yonsei University, Seoul, Korea. His research interests are wireless networks, multimedia communication, and embedded systems.



Li (Erran) Li (M’99) received the Ph.D. degree in computer science from Cornell University, Ithaca, NY, in 2001.

He is presently a member of the Networking Research Center in Bell Labs, Lucent Technologies, Murray Hill, NJ. He has published over 30 papers. His research interests are in networking with a focus on wireless networking and mobile computing.

Dr. Li has served as a program committee member for several conferences including ACM MobiCom, ACM MobiHoc, IEEE INFOCOM, and IEEE ICNP. He was a guest editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS special issue on Non-Cooperative Behavior in Networking.