# Capacity Allocation in Wireless LANs

Godfrey Tan and John Guttag
MIT Laboratory for Computer Science
Cambridge, MA 02139
{godfreyt, guttag}@lcs.mit.edu

**Abstract**

Today's access point based wireless LANs (WLANs) are inefficient and unfair. For many traffic loads they provide far less total throughput than they should, and do a poor job allocating what throughput they do deliver. Inappropriate association of nodes to access points and rates to flows plays a large role in these problems. We address a major root cause of this problem in this paper.

Current practice ignores the distinction between flows that connect two wireless nodes via an access point and flows that connect wireless nodes to the wired infrastructure. As wireless devices and applications become more pervasive, ignoring this distinction will lead to a significant degradation in perceived performance.

In this paper, we i) describe a series of examples that illustrates the impact of two-hop flows on the performance of the system, ii) provide a practical algorithm to solve the AP-assignment problem and iii) evaluate the performance of our algorithm against other approaches. Our preliminary results show that our algorithm can increase average achieved throughput by as much as $50\%$ for some traffic loads.

## 1 Introduction

The proliferation of devices with built-in wireless interfaces and the availability of low-cost wireless network access points (APs) have made wireless local area networks (WLANs) the ideal platform for homes, offices, airports, and malls. 802.11b [1] has been the most prevalent WLAN technology because of its cost effectiveness and relatively high data rate (11Mbps, in principle).

Devices that are equipped with 802.11b interfaces can form a WLAN in either *ad hoc* or infrastructure mode. In *ad hoc* mode, nodes that are physically near each other communicate in a peer-to-peer manner. In the infrastructure mode, which is far more popular, every node communicates with another node via an access point (AP) or base station that is connected to a distribution system. In the most common deployment, multiple access points are connected to each other via a wired infrastructure to provide coverage of a large area [5, 37, 38].

With the rapid growth of in the number of laptops and PDA's equipped with wireless interfaces, many enterprises have started to depend upon WLANs as critical component of their communication infrastructure. Until relatively recently, almost all of the traffic carried on these WLANs was between wireless nodes and the wired infrastructures. However, with the growing pervasiveness of wireless devices and applications, this now seems to be changing.

| Standard | Maximum Data Rate (Mbps) | Non-overlapped Channels |
|---|---|---|
| 802.11b/g/a | 11/54/54 | 3/3/12 |
| HiperLAN1/2 | 23.5/54 | 9/8 |

Table 1: Comparison of WLAN technologies.

Some smaller enterprises have started depending upon servers whose only network connection is through their WLAN. The improved performance of emerging wireless standards [24] such as 802.11a, 802.11g and HiperLAN2 (see Table 1) will likely accelerate this trend. Furthermore, the growth of applications such as instant messaging, voice over IP, and video conferencing is likely to increase the direct traffic between nodes on the WLAN.

As the demands on WLANs grow, it will become necessary to increase their capacity. Since multiple access points can operate on non-overlapped channels without interfering with each other, the capacity of WLANs can be increased by increasing the AP density. This will be facilitated by the increase in the number of non-overlapped channels provided by the emerging WLAN standards (see Table 1).

All of this suggests that we are rapidly entering an era of enterprise networks in which

1. The communication patterns in WLANs will consist of a mix of *one-hop* communication (between wireless nodes and the wired infrastructure) and *two-hop* communication (between two wireless nodes),
2. Applications will have diverse QoS requirements, and
3. The number of access points within a radio range (each operating on a non-overlapped channel) of a node will be large.

As the number of access points to which a node can be connected grows, the choice of access points becomes ever more important. Unfortunately, current approaches of selecting an AP based on signal strength alone, without considering the load, can significantly reduce the achieved system throughput.

Furthermore, as more and more communications among nodes within an enterprise network are done via the WLAN, the impact of two-hop flows on the overall system performance must be considered. The nature of two-hop flows challenges the traditional view of considering wireless access points as mere attachment points that provide mobile nodes with connectivity to the wired infrastructure. When two nodes within the WLAN communicate, the rate allocated to both down-link and up-link must be equal to avoid wasting capacity. As we show later, failure to coordinate the AP association and rate allocation to the end nodes of two-hop flows can have a dramatically deleterious impact on system efficiency, especially in the presence of applications with diverse QoS requirements.

In this paper, we:

1. Provide a model of WLAN's in which we define the problem (called the *AP-assignment* problem) of associating mobile nodes with access points and allocating bandwidth to flows so as to optimize aggregate throughput and achieve fair bandwidth allocation to flows,
2. Describe a series of examples to illustrate the impact of two-hop flows on the performance of the system,

3. Present a practical AP-assignment algorithm called *DLFD* (for Directional Least First Decreasing) that takes into account the various constraints imposed by two-hop flows,
4. Evaluate the achieved throughput and fairness of *DLFD* through simulation and compare them with other approaches under real-world scenarios, and
5. Discuss open problems and implementation issues.

In the next section, we give background information of recent work in the area of improving WLAN performance. We define the AP-assignment problem and discuss challenges surrounding it in Section 3. We present our algorithm in Section 4 and evaluate its performance through simulation in Section 5. We conclude in Section 6 by providing a summary and discussing future work.

## 2 Background

In this section, we discuss some of the literature related to the performance of WLANs and explain how our work attempts to build on it.

### 2.1 Load Balancing of Access Points

Balachandran *et al.* proposes that access points perform admission control and that users choose the access point with the least amount of load upon entering the network [4]. In their scheme, each mobile node explicitly expresses its lower and upper bounds of desired bandwidth. The admission control decision is made by a centralized admission control server that maintains the load information of all the APs and chooses the AP with the most available capacity and suggests that the node associates with it. Their work focuses on the system architecture for load balancing and admission control. In contrast, our work addresses general issues of optimizing the overall system throughput while maintaining high degree of fairness especially in the presence of two-hop flows.

Judd and Steenkiste [16] report an interesting pathological case: at the lecture halls of Carnegie Mellon University where the two access points covering the same area are constantly having differing loads, one overloaded and the other under-utilized. The reason has to do with the simple access point selection algorithm, deployed in most commercial 802.11b interfaces. That algorithm associates a node with an access point with the best signal to noise ratio (SNR). In the scenario presented, the receivers at most locations detect a slightly higher SNR from one access point than the other, thus resulting in most nodes associating with a single AP. Clearly, their study suggests that better mechanisms are necessary in the area of access point selection to enhance the overall performance of the system.

Recently, various vendors begin selling access points with load balancing functionalities [11, 32]. Unfortunately, little or no literature exists on how these functionalities are implemented. Balachandran *et al.* [4] observes that an access point with such functionalities periodically broadcasts beacons containing current load so that mobile nodes can make a better decision about which access point to attach to. According to the Data Sheet, the load balancing functionalities of Cisco AP-350 series is based on the number of users, error rates, and signal strengths [11]. However, based on several WLAN usage studies, the number of users may not be a good index for measuring load. We examine this issue in more detail in Sections 3 and 5.

## 2.2 Characterization of WLAN Usage

Several recent studies characterize the usage of wireless networks in various environments ranging from a single auditorium [5, 23] or building [37] to corporate [6] and university campuses [22].

We summarize the following common observations among all the studies:

1. The data transfer rates of users vary widely [6],
2. The total and maximum number of users associated with each access point varies widely [6, 22, 37], and
3. Data transfer rate of an access point has little correlation with the number of associated users [5, 6, 22].

We agree with Balachandran *et al.* [5] in concluding that it may not be wise to use the number of users as the basis for measuring load at access points as some vendors now do. It is also clear that the wireless system must accommodate varying needs of the users.

## 2.3 Performance, Fairness and QoS Issues

A large amount of research has been carried out in the area of fair packet scheduling (both weighted and unweighted) for wireless networks [21, 25, 29, 30, 33, 34, 39]. Most studies have focused on enhancing the wireless multiple access mechanisms such as 802.11's DCF under perfect channel conditions. Recent studies have taken into considerations various channel models [25, 30, 34]. Most studies are limited to single-cell (or AP) scenarios or networks in ad hoc mode, and mainly focus on the single issue of fair packet scheduling. Despite these limitations, much of this work on fair packet scheduling provides useful guidance in the more complex situation that we are tackling. We also note that several ideas from previous work are being incorporated in the ongoing 802.11e standard [14] to provide QoS support for 802.11 WLANs.

# 3 Problem Formulation

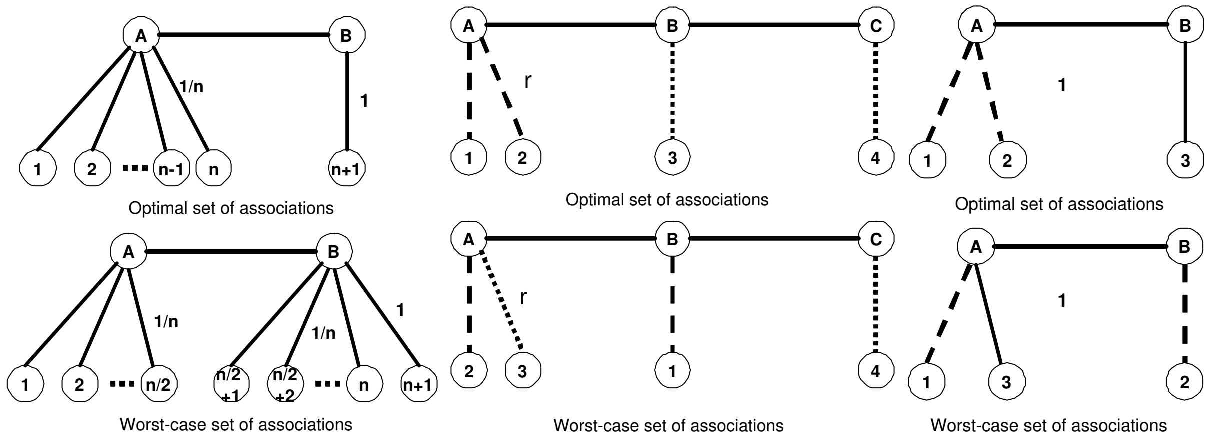
In this section, we first precisely formulate the AP-assignment problem, the problem of associating mobile nodes with access points and allocating bandwidth to flows. We then look at some simple examples illustrative of the best and worst case possibilities. We use the intuitions developed in this section to build practical algorithms described in Section 4. The section concludes with short discussion of some implementation issues.

We use the term *association* to refer to a binding between a mobile node and an access point. A set of associations is *optimal* if it allows for an optimal assignment set, i.e., a set of allocations of bandwidth to flows, that achieves optimal throughput. Three issues make the AP-assignment problem considerably more difficult than a traditional load balancing problem: i) the time-varying bandwidth demands of nodes, ii) the symmetric constraint of two-hop flows, and iii) the need to achieve fairness.

## 3.1 Model

We have developed a detailed network model of a WLAN and formally formulated the AP-assignment problem. Because of space constraints, however, here we define the problem in a less rigorous manner.

Let a WLAN be a set of one or more APs that are connected to each other and to the Internet via a wired infrastructure, and a set mobile nodes that communicate via the APs. For simplicity of presentation, we assume that APs are assigned




(a) One-hop flows with varying demands

(b) Two-hop flows with $r = 1$

(c) A two-hop and a one-hop flows

Figure 1: Example scenarios showing the optimal set of associations and the worst-case set of associations using LBU under various constraints. The numbers on the arrows indicate the nodes' bandwidth demands as fractions of the unit capacity. Each two-hop flow is identified by a unique dashed pattern and solid lines indicate one hop flows.

non-overlapped channels. In practice, the channels of the APs are usually manually configured during the initial setup of WLANs.

Let $M$ and $P$ be the (non-empty) sets of wireless nodes and access points within the WLAN respectively. We define $F$ to be the set of flows between node pairs. One end node of a flow must be a wireless node. The flow is considered two-hop flow if both are wireless nodes. *The AP-assignment problem is to associate each node $m \in M$ with an AP $p \in P$ such that the total throughput achieved by the flows $F$ is maximized without violating the fairness constraint.*

Our algorithms assume that the demand of each flow will be available to the system. This is discussed further in Section 3.3. It is well known that optimizing total throughput may lead to an unfair allocation in the sense of denying access to some users. The literature [7, 12, 36] contains numerous scheduling algorithms for wired networks as well as *ad hoc* wireless networks that achieve high degrees of fairness for some notion of fairness, e.g., max-min fairness [7] or proportional fairness [19, 20]. The goal is to provide fair bandwidth allocation to flows given a particular notion of absolute fairness.

Note that the throughput of a two-hop flow is the minimum of the bandwidth allocated to either end node. This suggests that for WLANs the fairness of the system should be measured as the achieved throughput of each node or flow instead of the allocated bandwidth. Let $\gamma^*$ be the absolute fair vector that describes the set of desired throughputs to be achieved by a set of flows $F$. The system is considered to be absolutely fair if the achieved throughput vector of all flows is proportional to $\gamma^*$. We discuss this further in Section 4.3.

At first glance, the AP-assignment problem resembles a classical load balancing or multiprocessor scheduling problem [2, 13, 17]: jobs with various sizes arrive on-line and each job must be immediately assigned to one of $m$ machines. The load on a machine is the sum of the sizes of the jobs and the goal is to minimize the maximum load on any machine.

However, the AP-assignment problem is differs from these problems (as well as bin packing [10, 18, 35] and bin stretching [3]) in significant ways: i) the total demand of the system may be larger than the total capacity, ii) two-hop flows impose the symmetric constraint, and iii) there are fairness constraints.

For the rest of this paper, we assume that each AP has the capacity of 1 unit and that each node can communicate with any AP. We make this assumption for simplicity of exposition, but we hope that it will be clear that the algorithms presented in Section 4 work in the absence of these assumptions. We will also assume that mobile nodes do not communicate directly with each, i.e., all communication occurs through access points. Clearly, if two wireless nodes can hear each other, they can form an *ad hoc* network and communicate directly instead of going through an AP, thereby doubling the maximum achievable throughput. Such optimization should be done when appropriate and we are pursing it as our future work. However, for the rest of this paper, we are focusing on cases in which each communicating wireless node pair is not within the radio range of one another and thus need to exchange data via APs.

## 3.2 Illustrative Examples

Many AP vendors ship systems that use a load balancing method that attempts to equally distribute nodes among access points [11, 32]. We call such algorithms *LBU* (load balancing by users). They have the great advantage of simplicity, but as we will see balancing nodes rather than load often yields results that are far from optimal. In the rest of this section we use LBU as the initial standard of comparison.

### *3.2.1 Impact of Demand Diversity*

For simplicity, we only consider one-hop flows. If each node has the same demand for bandwidth, LBU will achieve the maximum achievable throughput.

Of course, when nodes have differing bandwidth demands, the number of users is not a good proxy for load, and the results achieved by LBU can be far from optimal. The results achieved by LBU typically depend upon the order of arrival of users. However, there are many cases where no order of arrival yields an optimal or even good solution.

For example, Figure 1(a) contrasts the worst-case set of associations when LBU is used and the optimal association set for a simple traffic mix involving two classes of nodes that demand different fractions of bandwidth. Table 2 lists the throughputs achieved by each AP for the optimal and worst-case set of associations. The optimal association set achieves $100\%$ of the total demand whereas the worst-case LBU set of associations only meets $75\%$ of the two-unit demand. Of course, LBU will not always find the worst-case associations. However, for the example in Figure 1(a) no LBU solution exists that achieves the optimal throughput.

### *3.2.2 Impact of Two-hop Flows*

When the system contains two-hop flows, it is counter-productive to treat the AP-assignment problem as a load balancing problem. Two-hop flows introduce an additional constraint in maximizing the throughput: both end nodes must be allocated equal amount of bandwidth. A mixture of two-hop and one-hop flows make the problem harder yet.

Figure 1(b) shows the optimal association set and the worst-case association set produced by LBU for a traffic mix involving two two-hop flows each sending at the saturation rate. The optimal solution achieves the total throughput of 3 units

| Figure | $A$ | $B$ | $C$ | Total | Figure | $A$ | $B$ | $C$ | Total |
|---|---|---|---|---|---|---|---|---|---|
| 1(a) | $1:\frac{1}{2}$ | $1:1$ | - | $2:1\frac{1}{2}$ | 2(a) | $1:\frac{5}{6}$ | $1:1$ | - | $2:1\frac{5}{6}$ |
| 1(b) | $1:\frac{5}{6}$ | $1:\frac{1}{2}$ | $1:\frac{1}{2}$ | $3:2$ | 2(b) | $1:1$ | $1:\frac{2}{3}$ | $1:1$ | $3:2\frac{2}{3}$ |
| 1(c) | $1:1$ | $1:\frac{1}{2}$ | - | $2:1\frac{1}{2}$ | 2(c) | $1:1$ | $1:\frac{1}{2}$ | $1:1$ | $3:2\frac{1}{2}$ |
| - | - | - | - | - | 2(d) | $1:1$ | $1:1$ | $1:\frac{3}{4}$ | $3:2\frac{3}{4}$ |

Table 2: The total achieved throughput at each AP for each scenario shown in Figures 1 and 2. The throughput achieved in top and bottom figures are separated by ":". Assume that $r = \frac{1}{2}$ for all cases except Figure 1(b) in which $r = 1$, and that each node gets an equal share of the associated AP's capacity.

whereas the worst-case LBU solution only achieves 2 units. In the worst-case scenario, an end node from a two hop flow contends with an end node from a different two hop flow for the capacity of the AP $A$. This prevents the other end nodes of these flows from fully utilizing the bandwidth available to them. In contrast, the optimal solution keeps the two flows from competing against each other.

#### *3.2.3 Impact of Providing Fairness*

When the total demand at an AP is greater than its capacity, some nodes will have unsatisfied demands. How bandwidth is allocated among the nodes depends on the MAC layer fairness. For the following example, we assume that each node connected to an AP achieves an equal amount of channel bandwidth (this is the behavior of 802.11's DCF).

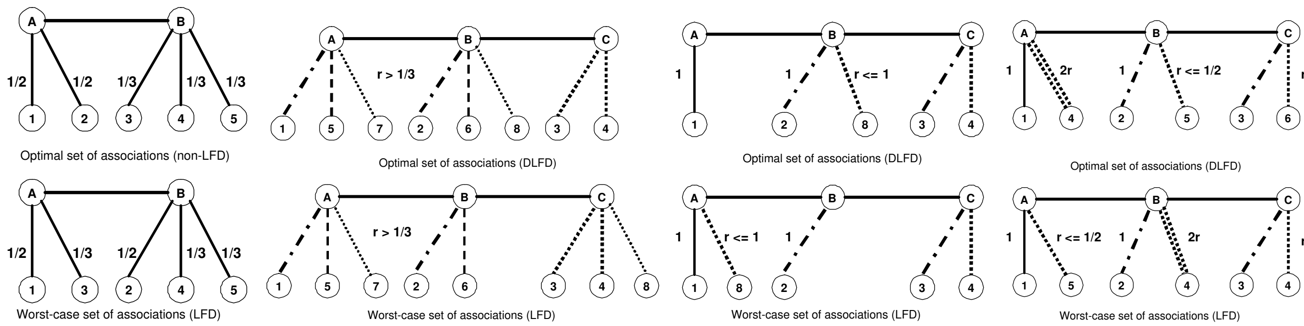
Figure 1(c) contrasts the best-case and worst-case sets of associations for a three-node WLAN containing a two-hop flow and a one-hop flow. In the worst-case association set, node 3 contends for the channel with node 1 and thus, gets $50\%$ of the channel bandwidth. The total achieved throughput is $1.5$ units, since $50\%$ of the capacity of AP $B$ goes unused. The best-case set of associations, however, achieves the maximum 2 units of throughput. Notice that if fairness is abandoned, the worst-case association set could yield 2 units of throughput by allocating no bandwidth to node 3.

## 3.3 Implementation Issues

In Section 4, we present two centralized algorithms that compute the association set based on available global knowledge. We believe that a centralized algorithm is practical since we can obtain most of the information pertaining to the WLAN, as discussed below.

Clearly, the algorithm the AP-assignment problem needs to be revised as nodes come and go and traffic patterns change. Whenever a new node arrives or significant changes in traffic conditions are detected, the algorithm should run again to find a new assignment set. Our algorithms are semi-online in that they use knowledge available within bounded space and time to improve performance. We are currently investigating how often these algorithms should be run. It will depend in part of the overheads associated with the re-association process. Based on a recent study, we note that the re-association or hand-off process between APs could take as little as a few milli-seconds provided that a node already knows it neighboring APs [28].

In addition to knowing $M$, $P$ and $F$, there are several additional inputs required to find an optimal solution to the AP-assignment problem: i) the capacity of APs, ii) the range matrix describing which APs are within the radio range of each




(a) LFD and DLFD produce the worst-case solution.

(b) Nodes with same demand

(c) A mixture of one-hop and two-hop flows

(d) A mixture containing a node with two flows

Figure 2: Example scenarios illustrating the strength and weaknesses of LFD and DLFD. The numbers on the arrows indicate the nodes' bandwidth demands as fractions of the unit capacity. Each two-hop flow is identified by a unique dashed pattern.

mobile node, iii) whether a flow is one-hop or two-hop, and iv) the demand associated with each flow.

Since APs are connected together via a distribution system such as the wired Ethernet LAN, the system can accurately obtain $P$ and $M$. In most WLANs, it is practical to figure out whether a particular flow is one-hop or two-hop based on the source and destination IP addresses. Furthermore, the capacity of each AP and the set of flows $F$ can be constructed by access points. By monitoring the connections either at APs or mobile nodes, the system can also construct the range matrix. For example, in the current 802.11b standard, each mobile node scans various channels to examine the availability of APs each of which periodically broadcasts beacons. This information can be communicated from mobile nodes to APs.

Estimates of the demand of each flow can be explicitly obtained through a demand specification mechanism such as the RSVP [8, 40] mechanism developed for the applications on the Internet. However, the demand may also be estimated by monitoring the recent behavior of the traffic flows. For instance, it is well understood how long-lived TCP flows increase their data rate over time [27]. Estimating the demands of short-lived flows is more problematical. Fortunately, however, the quality of a solution to the AP-assignment problem depends primarily on the behavior of long-lived flows.

Finally, once the algorithm computes the set of association between nodes and APs, and rate allocation to flows, we can provision the bandwidth of individual flows and nodes by controlling their sending rates and dropping the packets at the APs as appropriate. We suggest a simple mechanism using a leaky bucket filter to achieve that goal in Section 4.3.

## 4 Algorithms

In this section, we describe two algorithms for computing the set of associations between nodes and APs, and a mechanism to allocate rates to flows. Again, our goal is to run these algorithms periodically to compute an efficient association set for all nodes. The rate allocation mechanism described at the end of this section allocates rates to nodes and flows according to the specified fairness criteria.

The first algorithm, $LFD$ (for Least Fit Decreasing), associates each node with the access point with the smallest load. LFD does not take into account the symmetry constraint of two-hop flows and thus, its performance significantly suffers in certain traffic mixes involving two-hop flows.

The second algorithm, $DLFD$ (for Directional Least Fit Decreasing), is a variant of LFD but takes into account the various constraints of two-hop flows as explained in Section 3. As we demonstrate in Section 5, DLFD outperforms LFD in many traffic mixes. The rest of this section describes LFD and DLFD in detail.

## 4.1 LFD

LFD sorts the nodes in non-increasing order of demand, and associates each node with the access point with the smallest load. Ties are broken randomly. This greedy algorithm is similar in spirit to the well-known Best Fit Decreasing (BFD) algorithm for the off-line bin packing problem, which is known for its simplicity and good results [10].

In practice, there will be several sets of nodes each having the same demand. For instance, the two nodes each transferring a file to another node on the WLAN will have the same demand assuming that the file transfer protocol will transmit at the highest rate available. Thus, ties on node's demands will be common. Similarly, in many situations, there may be multiple access points with the same amount of load. When more than one choice exists, LFD will pick one randomly. Thus, even for a particular traffic mix, LFD may not always find the best solution although it could.

We define the $level$, $L(p)$, of the AP $p$ as the total demand of the nodes that are associated with the AP. $L(p)$ is initially $0$ and $L(p) >= 1$ means that the AP has at least as much demand as the capacity, which is always considered as 1 unit. We say that an algorithm respects the *LFD property* in associating a node $m1 \in M$ with an AP $p \in P$, if $m1$ has no less demand than the rest of the nodes that remain unassociated and that no other AP has a greater level than $p$.

When the system only contains one-hop flows, LFD performs very well in many cases. Clearly, LFD may fail to find the optimal solution even in those cases. Figure 2(a) shows an example scenario in which LFD produces the worst-case association set. The optimal solution requires that the two nodes with the most demand not be split between the APs. However, LFD will never associate those two nodes with a single AP. Assuming that each node should get equal share of the AP's channel bandwidth, the optimal achieved throughput is $9.1\%$ more than that achieved by the worst-case solution produced by LFD (see Table 2). The problem is worse when the system contains two-hop flows. We elaborate on this issue in the next section.

## 4.2 DLFD

DLFD is a variant of LFD that takes into account the various constraints imposed by two-hop flows. Our goal is to show that how a particular algorithm such as LFD can be modified to improve the performance when the system contains two-hop flows. DLFD respects the LFD property and therefore, the solution space of DLFD is a subset of LFD's.

DLFD is based on the following observations:

1. It's better to associate the end nodes of a two-hop flow with APs simultaneously instead of associating one at a time,
2. Asymmetric communications among APs can lead to inefficiencies,
3. In many cases, a few discriminating factors exist to choose the best AP, instead of a random choice, to improve the achieved throughput while maintaining a high degree of fairness.

Procedure RUNDLFD {

1. Sort $M$ in non-increasing order of demand. Break ties according to the node's one-hop demand and its connections with other nodes with higher demand.
2. For each $m1 \in M$ that is not associated,
   (a) Sort P in non-decreasing order of $level$. Break ties by the total two-hop demand.
   (b) Let $m2$ be the next node in $M$ after $m1$.
   (c) If $m1$ and $m2$ communicates and $m2$ has not been associated, associate each node with an AP from the pair returned by PICKAPPAIR($m1$, $m2$, $P$).
   (d) Otherwise, associate $m1$ with the AP returned by PICKAP($m1$, $P$).

}

PROCEDURE PICKAPPAIR($m1$, $m2$, $P$) {

1. Let $PP'$ be a subset of $P \times P$ such that choosing $(p1, p2) \in PP'$ to associate with $m1$ and $m2$ will not violate the *LFD property*.
2. Pick the most suitable AP pair $(p1, p2) \in PP'$ as follows assuming that $m1$ and $m2$ will be associated with $p1$ and $p2$ respectively, and return the pair.
3. If all APs have demands greater than their capacities, pick the AP pair that will result in the least amount of asymmetric communications. Break ties by selecting the pair with the most number of one-hop nodes.
4. Otherwise, pick the AP pair with the maximum potential throughput if $m1$ and $m2$ are associated with $p1$ and $p2$ respectively.

}

PROCEDURE PICKAP($m1$, $P$) {

1. Let $P'$ be a subset of $P$ with the smallest levels without violating the *LFD property*.
2. Pick the best AP $p \in P'$ as follows and return it.
3. If all APs have demands greater than their capacities, pick the AP with the least amount of two-hop bandwidth.
4. Otherwise, pick the AP with the highest potential throughput. Break ties as follows. Let $m2 \in M$ be the node with which $m1$ has the most two-hop communications and let $p2$ be $m2$'s associated AP. If $p2$ exists, construct a list of AP pairs $PP' = P' \times \{p2\}$. From $PP'$, pick the AP pair $(p1', p2)$ that will result in the least amount of asymmetric communication when $m1$ is associated with $p'$.

}

Figure 3: Pseudocode of DLFD

We explain each of these observations by showing the example scenarios where LFD will fail to find the best solution on average. Consider the example network in Figure 1(b) that contains two two-hop flows each sending at the saturation rate. Without loss of generality, assume that each of the nodes 1, 3 and 4 is associated with each of the three of APs as in the top figure. Since the load at each AP is the same (i.e. 1 unit), LFD arbitrarily picks an AP with which the remaining node 2 is associated. If node 2 is associated with an AP of either node 3 or 4, the total achieved throughput will be 2 units compared to the maximum achievable throughput of 3 units (see top figure in in Figure 1(b)). On average, LFD will find the worst-case solution. One way to avoid this is *to associate the end nodes of a two-hop flow with APs simultaneously instead of associating one node with an AP at a time*. For instance, the first node pair will be associated with two APs but the second node pair will associated with the last $AP$ which is the only one with remaining capacity.

When all APs have demands greater than their capacities, some nodes will receive less bandwidth than their demands. This could cause asymmetry in bandwidth available to the end nodes of two-hop flows, reducing the total throughput. Consider the WLAN in Figure 2(b) with four two-hop flows with the same demand $r$. Assume $r = \frac{1}{2}$. LFD often will provide the worst-case solution as shown in the bottom figure. However, there are ways to prevent this. Assume that LFD associates the first six nodes (from 1 to 6) with the APs as shown in the top figure. Now, there are three possible AP pairs with which nodes 7 and 8 can associate without violating the LFD property: $(A, B)$, $(A, C)$ and $(B, C)$. LFD will pick one randomly. However, if it picks a pair that contains AP $C$, the total achieved throughput will be sub-optimal (see bottom figure) since the bandwidth available to one end node of each flow (e.g. node 1) is less than the other (e.g. node 2). However, choosing the AP pair $(A, B)$ will yield the optimal throughput (see top figure). Thus, a good algorithm must *minimize the bandwidth reduction as a result of asymmetric bandwidth allocation to the end nodes of two-hop flows*.

DLFD attempts to achieve superior throughput than LFD based on these two main observations. Figure 3 depicts the pseudocode of DLFD. Like LFD, DLFD sorts the nodes in non-increasing order of demand and associates each node with an AP in that order. When ties exist, nodes with the highest one-hop demand are given priority. This is to avoid situations like the one shown in Figure 1(c). In this case, if nodes 1 and 2 are associated with the APs before node 3 (with a one-hop flow) is associated, the resulting association will yield sub-optimal throughput. Put it another way, nodes with only one-hop flows can easily cause the asymmetric bandwidth allocation to the already associated end nodes of two-hop flows. On the other hand, if node 3 is associated first, nodes 1 and 2 can be associated with AP $A$ (with the remaining capacity) simply because that's the only way the total throughput will increase.

Unlike LFD, DLFD attempts to associate a pair of nodes with a pair of APs instead of a single node with a single AP. Procedure PICKAPPAIR selects the most suitable pair of APs for the two nodes $m1$ and $m2$ that communicate with each other. In many cases, we may be able to choose among the possible AP pairs without violating the *LFD property*. Again, the reason we are imposing this constraint is to show that how the total achieved throughput can be increased in the presence of two-hop flows, by incrementally modifying an algorithm such as LFD.

When more than one choice exists, DLFD picks the best AP pair among the set of potential pairs. How the algorithm chooses the best pair among the candidates depends on whether there is any capacity left in any AP. If all APs have demands greater than their capacities, the goal is to minimize throughput reduction as a result of asymmetry in two-hop communications as explained earlier. For each AP pair, DLFD computes the potential throughput reduction as a result of associating $m1$ and $m2$ with the AP pair, based on the amount of load between the two APs. DLFD then picks the pair that will potentially lead to the least bandwidth reduction. This allows DLFD to find the best solutions in situations shown in Figures 2(b) and 2(d).

If at least one of the APs has any capacity left, DLFD attempts to maximize the resulting throughput if the nodes are associated with the APs. This is simply done by comparing the achieved throughput based on the remaining capacity of each AP and the symmetry constraint of two-hop flows. This optimization will allow DLFD to always pick the optimal set of associations for the example scenarios in Figures 1(b) and 1(c).

Procedure PICKAP selects the most suitable AP to associate with node $m1$. Again there may be multiple APs that respect the *LFD property* if they are to be associated with $m1$. When all APs have demands greater than their capacities, DLFD picks the one with the least amount of two-hop demand for the same reasons explained earlier. If at least one AP has positive remaining capacity, the AP that will potentially yield the highest throughput when associated with $m1$ is chosen. If $m1$ has two-hop flows, the potential throughput after association is computed in a similar way as in PICKAPPAIR.

### 4.3 Rate Allocation

Each algorithm described in this section computes the set of associations between nodes. When the total demand at any AP exceeds its capacity, the limited bandwidth must be allocated based on the absolute fair vector $\gamma^*$ that specifies the set of desired throughputs to be achieved by the flows. However, some flows may be considered more important than others. For our purpose, we simply assume that the fair share of each flow is the ratio of the flow's desired demand to the total demand of the system, and allocate each flow its fairshare of bandwidth. The fair share of the node is simply the sum of the fair share of its flows.

We note that the correctness of our algorithms does not depend on $\gamma^*$. If the system specifies a different $\gamma^*$ than the one we are suggesting, both LFD and DLFD should be adjusted as follows. In the first step of each algorithm, nodes should be sorted according to their fair shares (as indicated by $\gamma^*$) instead of their absolute demands.

802.11's DCF does not provide bandwidth guarantee to nodes. One possible way to achieve rate allcoation of individual nodes is to use the 802.11e standard [14, 26] that defines QoS support for 802.11 WLANs. However, 802.11e is still being drafted. Alternatively, senders' traffic can be shaped using well-known mechanisms such as the leaky bucket filter [7]. For our simulation, we use the latter for its simplicity. Due to space constraint and its relatively lesser degree of importance, we have omitted the details.

## 5 Evaluation

In this section, we evaluate through simulation the throughput and fairness achieved by the three algorithms LFD, DLFD and LBU for various traffic mixes. (Recall that LBU is a simple algorithm that divides nodes equally among the APs.) The purpose of this section is not to give a thorough evaluation of the algorithms. Rather, our intent is to give a flavor of how algorithms fare in a set of traffic mixes that we believe to be representative of the way WLAN's are and will be used in practice.

We consider two major scenarios: i) a peer-to-peer scenario and ii) a wireless-server scenario. For each scenario, we consider a handful of traffic mixes that clearly demonstrate both the weaknesses and the strengths of the algorithms. In the wireless-server scenario, a few nodes are the source of much of the traffic. In the peer-to-peer scenario traffic between wireless nodes is more uniformly distributed. For both scenarios, the traffic mixes also contain one-hop flows.

We extended the network simulator $ns{-}2$ [31] to mimic the infrastructure mode of 802.11b and implemented the three off-line algorithms. Like others, we found the maximum per-channel throughput of 802.11b to be roughly $6.45$Mbps. This throughput is much lower than the raw data rate of 11Mbps mainly because the packet headers and preambles are sent using a much lower rate of 1Mbps. Unless otherwise noted, each simulation run contains 3 APs, and thus the maximum achievable throughput is 3 units or $19.35$Mbps.

For each traffic mix, we performed $100$ simulation runs. First, we run the algorithm to be evaluated to compute the associa-

| Type | Rate (unit) | Rate (Mbps) | Ho-ps |
|---|---|---|---|
| 1 | 1 | 6.45 | 2 |
| 2 | 0.2328 | 1.5 | 2 |
| 3 | 0.5587 | 0.36 | 2 |
| 4 | 1 | 6.45 | 1 |
| 5 | 0.2328 | 1.5 | 1 |
| 6 | 0.5587 | 0.36 | 1 |

Table 3: Flow Types

| Traffic Mix | Flow Distr | No-des | Demand |
|---|---|---|---|
| P1 | 2:0:0:0:0:0 | 4 | 4 |
| P2 | 0:7:0:0:0:0 | 14 | 3.262 |
| P3 | 1:2:3:0:0:0 | 12 | 3.268 |
| P4 | 1:2:3:1:2:3 | 18 | 4.79 |
| P5 | 1:0:3:2:0:3 | 13 | 4.504 |

Table 4: Peer-to-peer traffic mixes

| Traffic Mix | Flow Distr | No-des | Demand |
|---|---|---|---|
| S1 | 3(1):0:0:0:0:0 | 4 | 6 |
| S2 | 6(1):0:0:0:0:0 | 7 | 12 |
| S3 | 2(1):3(1):0:0:0:0 | 7 | 5.398 |
| S4 | 1:2(1):0:1:0:0 | 5 | 3.932 |

Table 5: Wireless-server traffic mixes

tions between nodes and APs. The rate of each node is then allocated using the leaky bucket filter as explained in Section 4.3. If a node has multiple flows, each flow gets its fair share of the bandwidth allocated to the node according to the absolute fair vector $\gamma^*$. For simplicity, we assume that each node is within radio range of each AP.

Each traffic mix contains a mixture of the six types of flows as shown in Table 3. Type 1 represents bandwidth intensive applications such as file transfer protocols. This type of flow can consume the entire 1-unit of a single 802.11b single-channel bandwidth. Type 2 models the DVD-quality video applications rendering at full-screen. Type 3 models the relatively lower quality video applications rendering at a smaller screen size of 352x240. The remaining three types are the one-hop counterparts of the first three types. For simplicity, we use Constant Bit Rate (CBR) applications for each type of flow. Each packet carries the maximum 1500-byte payload with 1472 bytes of application data. The rest of this section evaluates the total achieved throughput and the fairness of each algorithm for various traffic mixes.

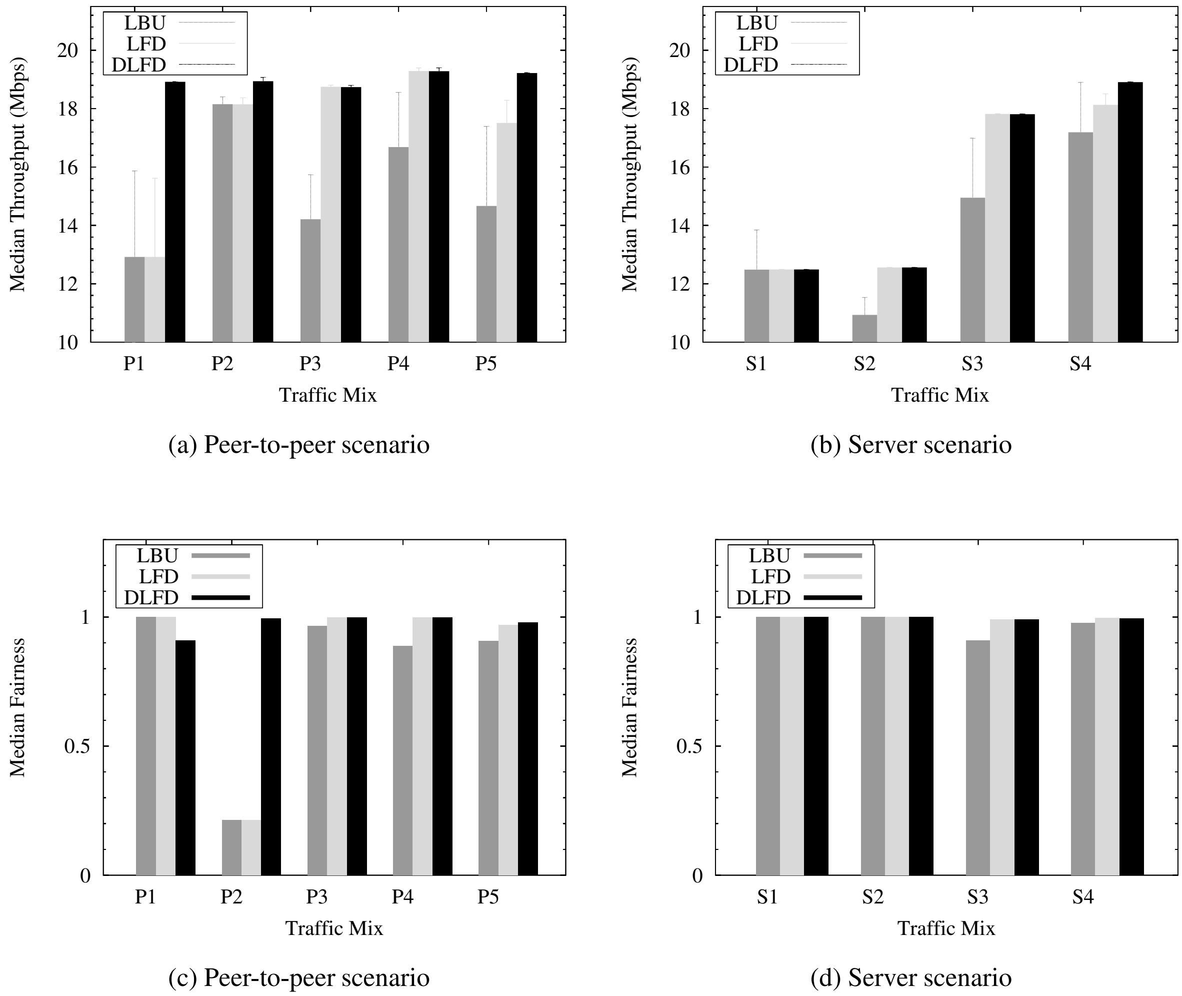
## 5.1 Computing Fairness

There are many different ways to evaluate fairness. Abstractly, fairness can be defined as a function that computes the fairness measure given the two vectors: the absolute fair vector $\gamma^*$ and the actual achieved throughput vector $\gamma'$.

A well-known fairness index based on the max-min fairness notion is the Jain's index [15, 9]. Jain's index can easily be extended to compute the fairness for a given pair of $\gamma^*$ and $\gamma'$ as follows. Let $n$ be the number of elements in $\gamma^*$. From $\gamma^*$, we can create a normalization vector $\alpha = \frac{min(\gamma^*)}{\gamma^*}$ that will be used in computing the fairness measure $\psi(\gamma')$ as follows:

$$\psi(\gamma') = \frac{(\sum_{i=1}^{n} \alpha_i * {\gamma'}_i)^2}{n * \sum_{i=1}^{n} (\alpha_i * {\gamma'}_i)^2} \quad (1)$$

This fairness has two major properties. First, the fairness is bounded between 0 and 1 (for $0\%$ and $100\%$). Second, if only $k$ of $n$ users share the resource equally with the remaining *n-k* users not receiving any resource, then the fairness in $k/n$ [15]. For example, assume that $\gamma^* = \{0.8, 0.2\}$. Then, the normalization vector $\alpha = \{0.25, 1\}$. If $\gamma' = \gamma^*$, the fairness measure will be $1$. If $\gamma' = \{0.2, 0.8\}$, the fairness will be $0.5623$. And if only one of the node has positive allocation, the fairness measure will be $0.5$. The fairness is $0$ if no nodes receive positive allocation. For more properties of this fairness index, we refer the readers to Jain's papers [9, 15]. Note that for max-min fairness, $\gamma^*$ should be set to a unit vector. As described earlier, we compute $\gamma*$ based on the nodes' demands for bandwidth.



(a) Peer-to-peer scenario

(b) Server scenario

(c) Peer-to-peer scenario

(d) Server scenario

Figure 4: Median throughput and fairness achieved by the three algorithms for various traffic mixes.

## 5.2 Peer-to-peer Scenario

Today, most WLANs carry relatively little peer-to-peer traffic. However, with the growth of pervasive computing applications, this is likely to change. In this section, we evaluate the throughput and fairness achieved by the three AP assignment algorithms for traffic mixes in which wireless peer-to-peer communications represent a significant fraction of the total demand. Table 4 details various peer-to-peer traffic mixes. The second column in the table represents the distribution of flows for each of the six types described in Table 3. The last two columns contain the number of nodes and the total demand in units respectively. For example, P1 has two flows each of which is attempting to transfer one unit of data between two wireless nodes. The total demand is thus four units, two for the uplinks and two for the downlinks.

Recall that the total capacity available is 3 units, so each mix is demanding more bandwidth than is available. Observe also that the traffic mixes cover a wide range of properties from homogeneous flows to complete heterogeneous sets of flows. For simplicity, we limit the number of flows from each type between 0 and 3 inclusively except for $P2$.

Figure 4(a) shows the median throughput achieved by each scheme for various peer-to-peer traffic mixes. DLFD achieves the highest throughput for each traffic mix. LFD performs as well as DLFD in some cases and worse than DLFD in others. As expected, LBU consistently performs worse than DLFD in all cases and LFD in most cases. The difference in throughput between DLFD and the other two is highest at about $50\%$ with traffic mix $P1$ that reflects the scenario shown in Figure 1(b). DLFD was able to find the optimal solution for each simulation run whereas the other two discovered it only occasionally. Notice that in all cases, DLFD has very small standard deviations.

Figure 4(c) shows the median fairness measure of the throughput achieved by the nodes. Except for the traffic mix $P1$, DLFD achieves the highest fairness values as LFD does. Even for $P1$, DLFD achieves the maximum achievable fairness without leaving the system under-utilized. Recall that the absolute fair vector $\gamma^*$ is computed based on the demand of individual nodes as a fraction of the total demand. Thus, for traffic mix $P1$, each element of $\gamma^*$ contains the same value, $0.25$. Again consider Figure 1(b) which represents the $P1$ traffic mix. Notice that the only way to achieve the maximum throughput is when two end nodes of the same flow is associated with the same AP. This will lead to the throughput vector $\gamma' = \{0.5, 0.5, 1, 1\}$ and the fairness measure of $0.9$, which is the maximum achievable fairness measure without leaving the system under-utilized.

### 5.3 Wireless-server Scenario

In this subsection, we evaluate the throughput and fairness achieved by the three AP assignment algorithms with various traffic mixes depicting the wireless-server scenario. Table 5 details various wireless-server traffic mixes. The number of servers, when greater than $0$, for each type of flow is shown in brackets.

Traffic mixes $S1$ and $S2$ represent a mix in which a single wireless file server is pushing data to each mobile node at the saturation rate. Clearly, the maximum achievable throughput is no more than 2 units since the server cannot obtain more than 1 unit of bandwidth. As shown in Figure 4(b), both DLFD and LFD achieve close to the maximum for $S1$ and $S2$. Notice that LBU fails to achieve the maximum throughput with traffic mix $S2$ since a mobile node will be associated at the AP where the server is also associated, thus limiting the server's sending rate.

Traffic mix $S3$ contains two servers: a file server and a high-quality video server. Both DLFD and LFD achieve high throughput. $S4$ consists of a mixture of video server traffic, peer-to-peer and one-hop communications. DLFD achieves the throughput close to the maximum whereas LFD and LBU achieves lower throughput. Observe that for all wireless-server traffic mixes, DLFD achieves the fairness measure of close to 1.

## 6 Discussion

In this paper, we investigated the problem of associating wireless nodes with access points with the goals of optimizing the system's aggregate throughput while providing fair bandwidth allocation. This problem differs from other resource allocation problems in the need to account for two-hop flows. These flows lead to a slightly different notion of fairness and have a large impact on achieved throughput.

In addition to carefully formulating the problem and explaining the problems with existing approaches, we presented a relatively simple and practical algorithm, DLFD, to solve the AP-assignment problem, Through simulations, we demonstrated that our algorithm achieves up to a $50\%$ improvement in throughput and a high degree of fairness for many realistic traffic loads.

We have no illusions about this paper being the last word on this important set of problems. On the contrary, our intent was to frame the problem in a way that would help to structure an ongoing dialog in the research community about how to improve the performance of AP-based wireless networks. We hope that our model and the set of touchstone examples contained in this paper will facilitate comparisons among various current and future proposals.

## References


[1] IEEE 802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Aug. 1999.

[2] Y. Azar, A. Z. Broder, and A. R. Karlin. On-line load balancing. *Theoretical Computer Science*, 130(1):73–84, 1994.

[3] Y. Azar and O. Regev. On-line bin-stretching. *Theoretical Computer Science*, 268(1):17–41, 2001.

[4] A. Balachandran, G. Voelker, and P. Bahl. Hot-spot congestion relief in public-area wireless networks. In *IEEE Workshop on Mobile Computing Systems and Applications*, Callicoon, NY, June 2002.

[5] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing user behavior and network performance in a public wireless LAN. In *Proc. of ACM SIGMETRICS'02*. ACM Press, June 2002.

[6] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *The First International Conference on Mobile Systems, Applications, and Services (Mobisys)*, May 2003.

[7] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

[8] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (rsvp) - version 1 functional specifications (IETF RFC 2205), 1997.

[9] D.-M. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17(1):1–14, June 1989.

[10] E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 2. PWS Publishing Company, 1997.

[11] Data Sheet of Cisco Aironet 350 Series Access Points. `http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/carto_in.ht%m`.

[12] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. *Internetworking: Research And Experience*, 1:3–26, April 1990.

[13] R. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, (45):1563–1581, 1966.

[14] IEEE 802.11 Working Group. Draft Supplement to International Standard for Information Exchange between systems - LAN/MAN Specific Requirements, Nov. 2001.

[15] R. Jain, D.-M. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. Technical Report 301, Digital Equipment Corporation, Sept. 1984.

[16] G. Judd and P. Steenkiste. Fixing 802.11 Access Point Selection (Sigcomm Poster Session). `http://www-2.cs.cmu.edu/~glennj/scp/FixingAPSelection.html`, Aug. 2002.

[17] Karger, Phillips, and Torng. A better algorithm for an ancient scheduling problem. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1994.

[18] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum, 1972.

[19] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, Jan. 1997.

[20] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[21] C. E. Koksal, H. I. Kassab, and H. Balakrishnan. An analysis of short-term fairness in wireless media access protocols. In *ACM SIGMETRICS*, June 2000.

[22] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proc. of the Eigth Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*. ACM Press, Sept. 2002.

[23] K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker. Experiences with a mobile testbed. In *Proc. of the Second International Conference on Worldwide Computing and its Applications (WWCA'98)*, Mar. 1998.

[24] J. Larocca and R. Larocca. *802.11 Demystified*. McGraw-Hill, 2002.

[25] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.

[26] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *European Wireless*, volume 1, pages 32–39, Feb. 2002.

[27] M. Mathis, J. Semke, and J. Mahdavi. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), 1997.

[28] A. Mishra, M. Shin, and W. Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. Technical Report 75, University of Maryland, 2002.

[29] T. Nandagopal, T. eun Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Mobile Computing and Networking*, pages 87–98, 2000.

[30] T. S. E. Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. In *INFOCOM (3)*, pages 1103–1111, 1998.

[31] ns-2 Network Simulator. `http://www.isi.edu/vint/nsnam/`, 2000.

[32] ORiNOCO AS-2000 System Release Note. `http://www.michiganwireless.org/tools/Lucent/ORiNOCO/AS-2000_Rel2_1/AS2%000_R2_10_01_Readme.txt`.

[33] D. Qiao and K. G. Shin. Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF. In *IWQoS*, May 2002.

[34] P. Ramanathan and P. Agrawal. Adapting packet fair queueing algorithms to wireless networks. In *Mobile Computing and Networking*, pages 1–9, 1998.

[35] S. Seiden. On the online bin packing problem. In *The 28th International Colloquium on Automata, Languages and Programming*, pages 237–249, July 2001.

[36] M. Shreedhar and G. Varghese. Efficient Fair Queuing using Deficit Round Robin. In *Proc. of ACM SIGCOMM*, August 1995.

[37] D. Tang and M. Baker. Analysis of a local-area wireless network. In *Proc. of the Sixth Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*. ACM Press, Aug. 2000.

[38] D. Tang and M. Baker. Analysis of a metropolitan-area wireless network. *Wireless Networks*, 8(2/3):107–120, 2002.

[39] N. H. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless LAN. In *Mobile Computing and Networking*, pages 167–178, 2000.

[40] P. White. RSVP and integrated services in the Internet: A tutorial. *IEEE Communications*, pages 100–106, May 1997.