Designing High Performance Enterprise Wi-Fi Networks

Rohan Murty[†], *Jitendra Padhye*[‡], *Ranveer Chandra*[‡], *Alec Wolman*[‡], *Brian Zill*[‡]
[†]*Harvard University*, [‡]*Microsoft Research*

Use of mobile computing devices such as laptops, PDAs, and Wi-Fi enabled phones is increasing in the workplace. As the usage of corporate 802.11 wireless networks (WLANs) grows, network performance is becoming a significant concern. We have built DenseAP, a novel system for improving the performance of enterprise WLANs using a dense deployment of access points (APs). In sharp contrast with wired networks, one cannot increase the capacity of a WLAN by simply deploying more equipment (APs). To increase capacity, the APs must be assigned appropriate channels and the clients must make intelligent decisions about which AP to associate with. Furthermore, the decisions about channel assignment, and associations must be based on a global view of the entire WLAN, rather than the local viewpoint of an individual client or AP. Given the diversity of Wi-Fi devices in use today, another constraint on the design of DenseAP is that it must not require any modification to Wi-Fi clients. In this paper, we show how the DenseAP system addresses these challenges, and provides significant improvements in performance over existing WLANs.

1 Introduction

In a typical office environment, the wired network is generally well-engineered and over-provisioned [12]. In contrast, deploying 802.11 wireless networks (WLANs) in enterprise environments remains a challenging and poorly understood problem. WLAN installers typically focus on ensuring coverage from all locations in the workplace, rather than the more difficult to measure properties such as capacity or quality of service. Thus, WLAN users commonly experience significant performance and reliability problems.

The usage model for enterprise WLANs is currently undergoing a significant transformation as the "culture of mobility" takes root. Many employees now prefer to use their laptops as their primary computing platform, both in conference rooms and offices [18]. A plethora of handheld Wi-Fi enabled devices, such as PDAs, cell phones, VoIP-over-Wi-Fi phones, and personal multimedia devices are becoming increasingly popular. In addition to the scalability challenges that arise with increased WLAN usage, the applications for many of these new mobile devices require better QoS and mobility support.

The need to improve enterprise WLAN performance has been recognized by both the research community [20, 19, 7, 10, 24] as well as industry [3, 5, 2, 4, 1]. Upgrades at the PHY layer, such as the transition from 802.11g to 802.11n, are important steps along the path to increasing WLAN capacity, but they are not enough. Deploying more APs has the potential to improve WLAN capacity, but one must also address issues such as channel assignment, power management, and managing association decisions.

In this paper, we present a new software architecture called DenseAP, that supports a dense deployment of APs to significantly improve the performance of corporate WLANs. A key emphasis in our design of the DenseAP system is on practical deployability. Because of the incredibly wide diversity of existing Wi-Fi devices, DenseAP must provide significant performance benefits without requiring any modifications to existing Wi-Fi clients. Furthermore, we do not consider any changes that require hardware modifications or changes to the 802.11 standard.

The DenseAP architecture and design challenge two fundamental characteristics of most current enterprise WLAN deployments. First, existing WLANs are designed with the assumption that there are far fewer APs than clients active in the network. In the DenseAP architecture however, the APs are deployed densely – in the common case there may be an AP in every office. Second, in conventional WLANs clients decide which AP to associate with, whereas the DenseAP system uses a centralized association control.

The scarcity of APs in conventional enterprise WLANs limits their performance in a variety of ways. For example, with a large number of non-overlapping channels (e.g. 12 in 802.11a) but only a few APs, the

WLAN is unable to fully utilize the available spectrum at each location. Because radio signals fade rapidly in indoor environments, adding extra radios to existing APs is not as effective as deploying a larger number of APs in different locations. If APs are densely deployed, each client can associate with a nearby AP, and will see better performance. A dense deployment also reduces the impact of the "rate anomaly" problem [13] that hurts the performance of conventional WLANs.

With a dense deployment of APs, clients have many possible APs to choose from, and therefore access point selection policy is critical to achieving good performance. In conventional WLANs, clients select which AP to associate with using only locally available information. Most clients use signal strength as the dominant factor in selecting an AP, yet it is well-known that this behavior can lead to poor performance [14]. For example, when many clients congregate in a conference room, they all tend to select the same AP even when multiple APs operating on different channels are available. To improve performance in this scenario, clients must associate with different APs.

In the DenseAP architecture, a central controller gathers information from all APs, and then determines which AP each client should associate with. Simultaneously, the central controller also decides on the assignment of channels to APs. Even though Wi-Fi clients implement their own association policies and we do not modify these clients, the DenseAP controller effectively bypasses the client association policy by only exposing to each client the particular AP with which it wants the client to associate. Using a similar technique, the DenseAP controller also carries out periodic load balancing by seamlessly moving clients from overloaded APs to nearby APs with significantly less load.

The DenseAP architecture is quite versatile, and capable of improving many aspects of performance of enterprise WLANs. In this paper, we primarily focus on describing how DenseAP significantly improves the capacity of enterprise WLANs. We define capacity simply as the sum total of throughput all active clients in the network can potentially achieve. We will also briefly discuss how the architecture impacts other aspects of performance, such as quality of service for delay and jitter sensitive applications.

One obvious question that arises when considering a dense AP deployment is whether the performance gains justify the costs. One approach to reducing equipment costs is to leverage existing enterprise desktops and convert them to APs, similar to our previous work on DAIR [8]. However, the key concern of enterprise IT departments when deploying any new technology is typically the people costs associated with managing that technology. The DenseAP system is designed to require very little management overhead: the DenseAP nodes are self-configuring, and the redundancy available from the dense deployment means that AP hardware failures do not need to be addressed immediately.

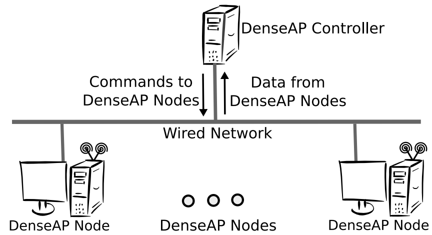

Figure 1. Overall architecture of the DenseAP system.

This paper makes the following new contributions. First, our system supports a high density of APs with off-the-shelf, completely unmodified clients. As a result, it provides performance benefits for all clients, including the *many* different types of handheld Wi-Fi devices that have recently appeared. Second, we demonstrate the performance benefits of our system at a significantly higher density than previous work. Third, we demonstrate that intelligent management of the association process is necessary even when you have a very high density installation of APs. Forth, we present a novel load estimation technique that allows our system to automatically factor in impact of external interference, such as traffic from nearby networks.

We have deployed the DenseAP system with 24 APs in our offices. The testbed can function in both 802.11a and 802.11g modes. Our experiments show that the system provides large improvements in performance over the existing corporate network. In specific cases, the improvement in throughput can be as large as 1250%. We present a series of experiments that show how various aspects of our system work together to provide these gains. We also show that our system is capable of handling nomadic and mobile clients.

2 Design Overview

Figure 1 is a high-level illustration of the DenseAP system. The system consists of several DenseAP nodes (DAPs) which provide wireless service and a DenseAP controller (DC) which manages the DAPs.

A DAP is a programmable Wi-Fi AP connected to the wired network. Each DAP periodically sends summaries to the DenseAP controller comprising of a list of associated clients, their traffic pattern summaries, RSSI values of a few packet samples from their transmissions, current channel conditions, and reports of new clients requesting service from the network. We classify DAPs into two categories: we refer to DAPs that do not have any clients associated with them as *passive*; those that have at least one associated client are called *active*.

The DC manages the DAPs. The periodic reports sent by the DAPs provide the DC with a global view of the network activity. Using this global view, the DC selects the right DAP for a client, allocates channels to DAPs, performs load balancing when needed, handles client mobility, and deals with DAP failures.

In the next section, we describe the mechanisms the DC uses to ensure that the client associates with the selected DAP. Then, we describe the algorithms involved in selecting the DAP and a channel for the DAP. We discuss power control and related issues in Section 5

3 Association Mechanism

In conventional WLANs, APs advertise their presence by sending out Beacon frames which include their SSID and BSSID. Prior to association, clients gather information about the APs by scanning the channels one by one, and listening for Beacons on each channel. This is called "passive scanning". The clients also perform "active scanning", whereby they send out a Probe Request frames on each channel. These are requests for APs to send out information about themselves. APs respond to Probe Requests with Probe Response frames, the contents of which are similar to Beacon frames. Once the client gathers information about all APs, it decides which AP to associate with.

The 802.11 standard allows APs to beacon with the SSID field set to null – this is referred to as a hidden SSID. A client that wishes to associate with an AP using a hidden SSID must first send out a Probe Request that contains the SSID of that network, which will then cause the AP to provide a Probe Response. For any client that does not provide the correct SSID, the AP does not respond.

The DC performs association control by exposing DAPs to clients on a "need to know basis". This is achieved as follows. First, the passive DAPs (i.e., those that do not have clients associated with them) in the network do not send out any beacons. The active DAPs do send out beacons but with a hidden SSID. Second, each DAP maintains a local access control list (ACL) of client MAC addresses. On receiving a probe request from a client, the DAP replies with a probe response message only if the client's MAC address is in its ACL. If a DAP receives a probe request (it may be a broadcast request) from a client whose MAC address is not in its ACL, it sends a message to the DC informing the controller that a client might be requesting service. The DC determines which, if any, DAP should respond to the probe request and adds the MAC address of the client to the ACL of that DAP.

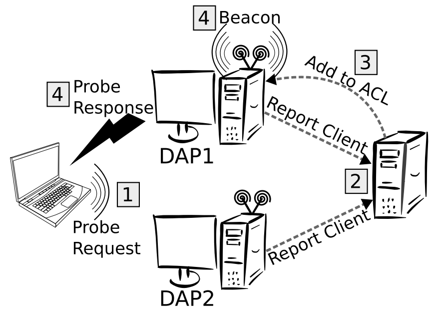

Figure 2. Association in the DenseAP system

By adding the MAC address of a client to only one DAP's ACL at a time, the DC ensures that for the SSID associated with the DenseAP network, *only one* DAP is visible to the client at any given time.

Note that traditional MAC address filtering could not have achieved this. MAC address filtering only prevents association, not probe responses. With traditional MAC address filtering, a client would discover several DAPs, and it may not even try to associate with the one that the DC has chosen for it.

We will now illustrate how these two techniques are used when a client associates with the system for the first time, and handing off a client from one DAP to another.

3.1 Associating New Clients

Figure 2 illustrates the association mechanism instrumented in DenseAP. **(i)** A new client C with mac address C^{mac} broadcasts probe requests. **(ii)** DAPs DAP_1 and DAP_2 receive probe requests and inform the DC of C^{mac}. **(iii)** The DC executes the association algorithm and determines which AP the client should associate with. Assuming it picked DAP_1, the DC then sends a message to DAP_1 to add C^{mac} to its access control list (ACL_{DAP_1}). **(iv)** DAP_1, on receiving the next probe request from C, checks to see if $C^{mac} \in ACL_{DAP_1}$. If so, it responds to C with a probe response thus initiating the association process. If DAP_1 was not previously beaconing, it now begins.

The reader may wonder why DAPs beacon at all. One reason is that beacons are essential for allowing the clients to enter power-save mode. In addition, certain popular drivers automatically disconnect from an AP if they do not receive periodic beacons.

We note a few points about this mechanism. First, we have verified that this mechanism works with the device drivers of a variety of popular 802.11 chipsets including Atheros, Intel Centrino, Realtek, Ralink, and Prism2. Second, if a client fails to associate with the assigned DAP (e.g. due to interference near the client), the DC detects this since DAPs periodically report back information about their associated clients. The DC then reassigns the client to a different DAP. Third, ACL entry for a client is maintained only as long as the client is as-

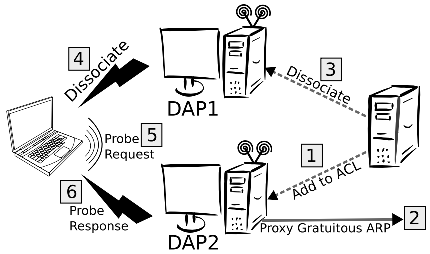

Figure 3. An example handoff in the DenseAP system. The client is switched from DAP_1 to DAP_2.

sociated with the AP.

3.2 Client Handoffs

The DC may sometimes want to handoff clients from one DAP to another, for load balancing or because the client's location has changed. Figure 3 illustrates the sequence of steps that lead to a seamless handoff in DenseAP.

Assume that client C has successfully associated with DAP_1. The following steps are taken when the DC decides to handoff C from DAP_1 to DAP_2. **(i)** The DC adds C_{mac} to the ACL on DAP_2. **(ii)** To ensure any further traffic flowing towards C is routed via DAP_2, DAP_2 sends out a gratuitous proxy ARP message containing C_{IP} and DAP_2^{mac} to the wired subnet. **(iii)** The DC asks DAP_1 to send a *disassociate* frame to C. **(iv)** DAP_1 removes C^{mac} from the local ACL and sends a disassociate frame to C. It also cleans up all local association state pertaining to C. **(v)** C receives the disassociate frame and immediately begins to scan for other DAPs by sending out probe requests. **(vi)** Upon receiving C's probe request, DAP_2 responds with a probe response.

After associating with DAP_2, C does not send out a DHCP request since the time taken to re-associate did not cause a local `media-disconnect` event. Therefore, it is important to ensure that the DC only hands off clients between DAPs that are on the same subnet. In practice, we expect that all DAPs managed by a DC will be on the same wired subnet. We present results illustrating the efficacy of our handoff mechanism in Section 6, along with time required for each step.

Having described the mechanisms by for enforcing the association decisions, we now describe the policy for making these decisions.

4 Association Policy

The goal of the association policy is to improve the overall system capacity. We do this by picking the "right" DAP for a client to associate with, and when needed we select the "right" channel for that DAP to operate on.

Intuitively, the way to improve overall system capacity is to have each client to associate with a nearby, lightly-loaded AP. Furthermore, APs that are close to each other should operate on orthogonal channels. We will formalize these notions in the rest of the section.

We do not claim that our association policy is optimal. Rather, our aim is to provide a significant improvement over existing WLAN networks, by taking advantage of the dense deployment of DAPs, and without requiring any changes to the clients.

In this section, we first present a metric we call *Available Capacity* to rank all possible DAPs that a client can associate with. We then describe the association policy for four scenarios: (i) when a new client shows up, (ii) when the wireless channel conditions change, (iii) when clients move, and (iv) when DAPs fail. A more detailed description of our association policy and related issues is available in [22].

4.1 The *Available Capacity* Metric

When we select a DAP for a client to associate with, we want to pick a DAP where a client can expect to get good throughput. Unfortunately, it is not easy to determine what throughput a client can expect to get when associated with a DAP. The value depends on several factors, such as quality of the channel between the DAP and the client, presence of other traffic/interference, autorate algorithms in use and the CCA thresholds used by the client and the AP. Rather than estimating each of these factors, we focus on the two that affect the expected throughput the most, namely: the transmission rate the client and the DAP can use to communicate with each other; and how busy the wireless medium is in the vicinity of the client and the AP.

The transmission rate is a function of, among other things, how well the signal propagates between the client and the DAP. For example, if a DAP and a client are far away from each other, they generally won't be able to communicate at high transmission rates, since the wireless signal degrades with distance. Alternatively, when a DAP and a client are close to each other, they will be able to communicate at higher transmission rates. However, if the wireless channel is busy with other traffic, the expected throughput will be lower, since the client and the DAP will have fewer opportunities to transmit packets.

The combined impact of the transmission rate and the busy medium is approximated by the *Available Capacity* (AC) metric as follows. Given a channel (C), a DAP (D) and a client (M), AC_{DM}^{C} is the product of *free air time* on C in the vicinity of D and M, and the *expected transmission rate* between the D and M. The free air time is simply the percentage of time when the wireless medium is not in use. Our notion of load at a DAP on a particular channel is defined by (1 - free air time).

The intuition behind this metric is that the DAP with the highest available capacity will allow the client to send the most data, while simultaneously reducing the impact on other clients in the network. For example, if a client and DAP can communicate at a high transmission rate, then each frame will consume less air time, and the client will be able to send more data. Furthermore, other clients on the same channel in the vicinity will have more opportunity to transmit. On the other hand, if the channel has low utilization then it is acceptable for the client and the DAP to communicate at a low transmission rate because it will have little impact on other clients.

This metric is similar to the one proposed in [10], where clients associate with the AP which is the least loaded and offers the best data rate. We compare and contrast our work with [10] in more detail in Section 8.

We now describe how we estimate the free air time and the expected data rate for a given DAP/client pair. We stress that we do not expect these calculations to be precise. Our intention is to provide a reasonable ordering of DAPs, in order to pick a good AP for the client to associate with. The load balancing process described later in the section can reassign the client to a different DAP, should the conditions change and the initial choice is no longer appropriate.

4.1.1 Estimating Free Air Time

Given a dense deployment of DAPs, it is likely that the DC will associate a client with a nearby DAP. Hence, it is also likely that the wireless channel conditions near the DAP and the client associated with it will be similar. Thus, we only estimate the free air time on a channel in the vicinity of the DAP.

The amount of free air time around the DAP depends on the traffic generated by the DAP itself (i.e. downlink traffic), the uplink traffic and any background traffic/interference. The background traffic includes traffic generated by other DAPs and clients in the vicinity that are functioning on the same channel. Of these quantities, the DAP can easily determine the amount of air time consumed by the traffic it generates. To determine the rest, we have devised a method inspired by the ProbeGap technique [16].

Each DAP periodically sends a small broadcast packet at a fixed transmission rate on the highest priority driver queue, which is usually reserved solely for high priority PSM packets. The packets in this queue are sent even if the DAP has other packets pending in the normal data queue.

The DAP records the difference in time from when the packet was queued to when the transmit completion interrupt signals that the frame was successfully sent. When the channel is idle, the packet will be transmitted immediately and the measured delay will be the sum of frame transmission time plus various OS overheads. We denote this value by δ_{min}. Because the channel is idle, δ_{min} does not depend on channel conditions. The transmission time of the frame is determined by the transmission rate and the size of the packet, both of which are constant. The OS overheads depend on the hardware and software configuration of the DAP. If the channel is busy, the packet will be delayed by more than δ_{min}. This additional delay depends on several factors such as number of contending stations, the size of packets they are sending and their transmission rates. To obtain a good estimate of the fraction of time the channel is busy, we send a number of such probe packets and count the fraction of packets that experienced a delay of more than δ_{min}. Because δ_{min} does not depend on channel conditions, it can be calculated apriori by performing simple calibration experiments.

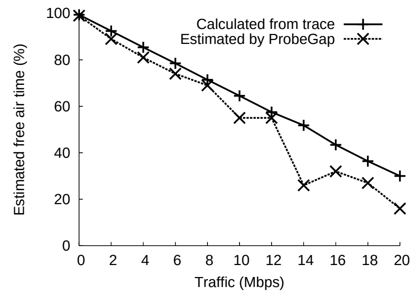

Figure 4. Validation of ProbeGap method

This technique works quite well in practice, as shown in Figure 4. The experiment was performed as follows. We set up a DAP with no associated clients on an otherwise idle channel in the 802.11a band. We had previously determined that δ_{min} was 250 microseconds. Using another AP and a client, we generated different volumes of CBR UDP traffic on the channel. We had also set up a sniffer to capture every packet. This graph shows the amount of free air time estimated by the DAP, as well as the "actual" free air time calculated using the trace captured by the sniffer. We see that the estimate provided by our method is a good approximation, although we tend to slightly underestimate the free air time.

Note that the probe packets are not delayed by the normal data traffic generated by the DAP itself (i.e. downlink traffic). So, to estimate the free air time, the DAP adds the air time consumed by the traffic it generates to the estimate obtained by the ProbeGap method before reporting it to the DC.

If a DAP has clients associated with it, it only needs to report the free air time for the channel it is currently on. If a DAP has no associated clients, it can scan all channels and report to the DC the channel that has the most free air time.

In our current implementation, each DAP sends the load estimation probes every 200ms. The free air time is estimated over a period of 20 seconds.

4.1.2 Estimating Expected Transmission Rate

It is difficult to accurately predict the transmission rate a client will achieve when communicating with a DAP (or vice-versa). The rate primarily depends on how well the DAP receives the client's signal. However, the rate also depends on a variety of other factors such as the autorate algorithm implemented by the client, power levels used by the client, and channel conditions near the client. Of these factors, we can only estimate how well the DAP receives a client's signal.

When attempting to associate, clients send out probe request messages which are overheard by nearby DAPs who then inform the central controller. We estimate the quality of the connection between the client and the various candidate DAPs using the signal strength (RSSI) of the received probe request frames at the various DAPs. We convert these observed signal strengths into estimates of expected transmission rate by using a mapping table. The mapping table bucketizes RSSI values into fixed-size buckets, and assigns an expected rate to each bucket. We assume that the same transmission rate will be used by both the client and the AP. We call this the *rate-map* approach. The mapping table is initially generated by manual profiling using a few clients at various locations. It can then be refined as actual data from more clients is gathered during live operation.

At first glance, it may appear that extrapolating the signal strength observed in the uplink direction to an expected transmission rate in both directions could result in inaccurate estimations and/or poor performance, especially considering the other factors that are ignored. Yet, in our system, we find that it provides reasonable results for the following reasons. First, given the density of access points, a client generally associates with a nearby DAP. For such short distances, we find that signal strength measured in one direction is a good approximation of signal strength seen in the other direction. Second, because the client and the DAP are usually close to each other, we generally see good signal strength in both directions. Most commercial Wi-Fi cards behave similarly in such conditions. Finally, note that we do not need the exact transmission rates used by either the client or the DAP. The conversion table is merely a way to ranking the relative importance of the observed signal strength. In Section 6, we will present results that demonstrate the usefulness of the rate-map approach.

We now describe how the AP selection algorithm uses the available capacity metric.

4.2 Associating a New Client

When a new client first appears in the network, it scans on all channels and sends out probe requests. Because this client has not yet been added to the ACL of any DAP, all DAPs that hear the probe requests simply report them to the DC. To calculate reasonable signal strength estimates, the DC waits for a short while (10-30 seconds in our current implementation) after the first report of a new client is received. During this interval it continues to collect reports of probe request packets from DAPs. At the end of this interval the DC calculates the average signal strength of all the probe request frames seen by each DAP. The rest of association algorithm is illustrated by the following example.

Assume two DAPs, A and B hear probe requests from a client M. Assume that A is active i.e. it already has other clients associated with it, whereas B does not (passive). For both A and B, the DC first calculates the expected rates with M, R_{AM} and R_{BM}, using both the observed signal strengths and the rate map table. Then, the DC considers the amount of free air time at each DAP. A already has clients associated with it, and therefore it is operating on some channel X. Hence, A has already been reporting free air time for that channel. We denote this by F_{XA}. Using the most recent report, the DC calcuates the available capacity at A on channel X by $AC_{AM}^{X} = F_{XA} * R_{AM}$.

Now let us consider DAP B. It has no clients associated with it. Let us assume that DAP B has recently seen the highest available free air time on channel Y, denoted by F_{YB}. The DC calculates the available capacity at B on channel Y as $AC_{BM}^{Y} = F_{YA} * R_{BM}$. The DC then compares AC_{AM}^{X} and AC_{BM}^{Y}, and picks the higher of the two. If they are equal, it decides in favor of B, since B has no clients associated with it. In general, whenever available capacity of several DAPs are equal, the DC always picks one that has the fewest clients associated with it. If the DC picks A, it adds M's mac address to A's ACL. If it picks B instead, it first instructs B to stop scanning and to stay on channel Y. It then adds M's mac address to B's ACL. In both cases, the rest of the association process unfolds as described in Section 3.1.

Note two key aspects of this algorithm. First, we never move existing clients to another DAP as a result of a new client association. Second, DAPs are only assigned channels on an *on-demand* basis, as part of the association process. A DAP is assigned a channel only when a client in its vicinity requests service from the network. When a DAP becomes passive, it no longer has an assigned channel.

4.3 Load Balancing

The goal of the load balancing algorithm is to detect and correct overload situations in the network. We expect that such situations will be rare in an environment with a dense deployment of access points, and with numerous available orthogonal channels (e.g. 12 in 802.11a). However, it is important to watch for, and correct the overload situations if and when they occur.

For example, an overload situation might occur if many clients congregate in a conference room, and the network conditions are such that the algorithm described in Section 4.2 assigns several of them to a single DAP. In such a situation, all clients simultaneously transmitting or receiving data can cause an overload at the DAP.

The load balancing algorithm works as follows. Once every minute, the DC checks all DAPs to see if any are severely overloaded. Recall from Section 4.1.1 that the busy air time (load) calculation incorporates the impact of traffic/interference near the DAP and the downlink traffic generated by the DAP. We consider a DAP to be overloaded, if it has at least one client associated with it, and it reports free air time of less than 20%. In other words, the channel is more than 80% busy in the vicinity of this DAP. The DC considers the DAPs in the decreasing order of load. If an overloaded DAP (A) is found, the DC considers the clients of A as potential candidates to move to another DAP. Recall that the DAPs send periodic summaries of client traffic to the DC. These summaries include, for each client, a smoothed average of the sum of uplink and downlink traffic load generated by the client during the previous interval. The load is reported in terms of air time consumed by the traffic of this client, and the average transmission rate of the traffic.

For each client $M \in A$, the DC attempts to find a DAP B such that the expected rate M will get at B is no less than the average transmission rate of the client at A, and the free air time at B is at least 25% more than the air time consumed by M at A. If such a DAP is found, M is moved to B using the process described in Section 3.2. Note that if B had no clients associated with it, the DC will also assign it a channel (the one B reported to have the most free air time on), just as it would do when associating a new client.

The load balancing algorithm moves at most one client that satisfies the above criteria during each iteration. Furthermore, once a client M has been handed off from A to B, it is considered ineligible to participate in the next round of load balancing. These hysteresis techniques are intended to prevent oscillations.

We note a few things about the load balancing algorithm. **(i)** Our algorithm is conservative. Moving clients from one AP to another is a potentially disruptive event, and we try to minimize how often we force such reassociations to occur. **(ii)** The load balancing algorithm improves overall system throughput in two ways. First, the client that is moved to the less-loaded AP can ramp up and consume more bandwidth. Second, the clients that stayed with the previously overloaded AP now have one less client to contend with, and they can also increase their throughput. **(iii)** It is sometimes possible to do load balancing by changing the channel of the overloaded DAP. This technique is useful only if the background traffic/interference (potentially from other DAPs) on the channel is significantly higher compared to the traffic sent/received by the overloaded DAP itself. However, the drawback of this technique is that all clients associated with the DAP will have to to re-associate. Since we consider client re-associations to be disruptive events, we do not to use this technique.

4.4 Mobility

The DC keeps track of a client's location, using the algorithm described in [11]. The algorithm takes into account the signal strength of a client's transmissions as reported by various DAPs, and the location of those DAPs, to determine the approximate location of the client. The median location error is about 1.5 meters. This is sufficient for our purpose because we only need to detect that the client's location has changed significantly.

The DC updates the locations of clients in the system every 30 seconds. When a client's location changes by more than 10 meters, the DC finds another DAP for the client to associate with, using the criteria described in the previous section. If such a DAP is found, the client is handed off to the new DAP. A client that undergoes handoff is considered ineligible to participate in the subsequent round for load balancing to prevent oscillations. It is, however, eligible to participate in another, mobility-related handoff.

4.5 Fault Tolerance

DAPs send periodic reports to the DC, so it is easy for the DCto detect when a DAP fails. In our current implementation, if the DC does not receive any reports from a DAP for up to one minute, it flags the DAP as a possible failure and does not assign any new clients to it. The clients associated with the failed DAP get disconnected. These clients immediately begin scanning for other DAPs in the vicinity by sending out probe request messages. Other DAPs in the vicinity pick up these probe messages and alert the central controller, which assigns these clients to other DAPs, as per the association policy.

5 Power Control

In a dense deployment of DAPs, transmit power control can mitigate the effects of interference between DAPs on the same channel [20], and increase spatial reuse. Since we do not wish to modify clients, we must do do power control at DAPs alone. However, such unilateral power level can cause the clients and APs to operate at different transmit power levels. Prior work [20] has shown that

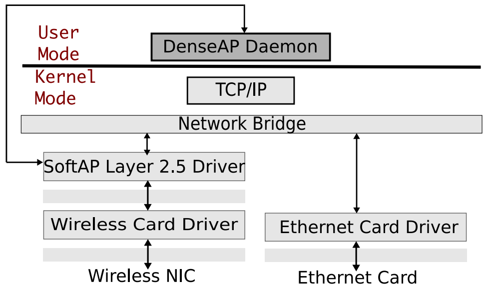

Figure 5. The network stack on each DenseAP node

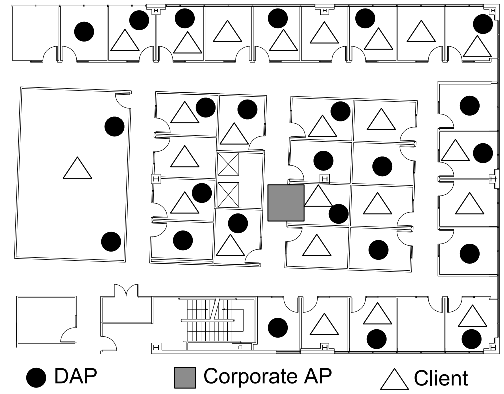

Figure 6. The testbed. The area is roughly 32m x 35m. The rooms have full walls, and solid wood doors.

asymmetric transmit power levels can increase the number of hidden terminals in a WLAN coverage area.

We have implemented and tested several adaptive power management schemes in our testbed. We do not present detailed results due to lack of space. Briefly, our results confirm the observations in [20]. Based on these results, as well as those reported in prior work, we conclude that unilateral power control at DAPs is undesirable, and the best policy is to simply use the maximum power level. A secondary benefit of this scheme is that it provides better coverage in the intended coverage area.

Two other parameters that can also affect the overall WLAN capacity are the Clear Channel Assessment (CCA) threshold used by each DAP [19], and the autorate algorithm implemented on each DAP. The wireless cards we used in our testbed do not allow us to change the CCA threshold. Auto-rating algorithms have been studied extensively by prior research. DenseAP nodes use the autorate algorithm described in [25].

6 Evaluation

We have built a prototype implementation of the DenseAP system. Figure 5 illustrates the network stack on each DAP. The network stack enables AP functionality on ordinary desktop machines. An integral part of the stack is our software AP (SoftAP), a fully programmable AP for the Windows platform. The wired and wireless interfaces are bridged. Each DAP also runs a DenseAP daemon, a user-level service responsible for managing local access point functionality. The service periodically queries the SoftAP driver, and sends summaries of client statistics to the DC. It also receives commands from the DC and sets appropriate parameters in the driver.

The DAPs are off-the-shelf PCs running Windows Vista, and the networking stack described above. Each machine is equipped with a Netgear JWAG511 wireless NIC. The wireless NICs are based on a chipset from RealTek. They support operation in 802.11 a/b/g modes, with one limitation: in the 802.11a mode, they can operate only on the lower 8 channels (channels 36-64). We also found that the RealTek cards do not work reliably in promiscuous mode, so we use an additional radio on each DAP to simulate promiscuous mode. This second radio is not fundamental to our approach, and is used only to compensate for the shortcomings of the RealTek card. All of the DAPs are connected to the same IP subnet on their wired Ethernet link. The DC is an ordinary desktop-class machine.

The DenseAP testbed is deployed on a portion of our office floor, as shown in Figure 6. The testbed consists of 24 DAPs. The DAPs are deployed roughly in every other office. Within each office, the machine is placed on the floor; the exact location determined by the consent of the occupant. This area of our building is served by a single corporate WLAN AP. The AP is located roughly at the center of the area, and is placed on the ceiling. Note that the DenseAP deployment is 24 times denser than the corporate WLAN. In addition to the DAPs, we have also deployed 24 machines to serve as clients. The clients are a mix of ordinary desktop and laptop machines, equipped with a variety of off-the-shelf wireless NICs.

Most of the the experiments reported in this paper were run in 802.11a mode (5 GHz band). There is very little corporate traffic in the 802.11a band. Thus, for most experiments, the background traffic is negligible, and we did not need to run the load estimation algorithm. The 802.11g band does see a fair amount of usage during normal office hours, but to avoid impacting corporate traffic, we were limited to conducting 802.11g experiments outside of normal work hours. We now turn to evaluating the performance of the DenseAP system using this testbed. We begin by validating the rate-map approach (section 4.1.2), which lies at the heart of our association and load balancing algorithms.

6.1 Usefulness of Rate-Map Approach

The rate-map approach (section 4.1.2) is the foundation of the association and load balancing algorithms. This approach is based on the hypothesis that the signal strength of a client's probe request packets, as observed by a DAP (i.e. *uplink* packets), is a good approximation of the transmission rate a client can expect in *both the up link and downlink* directions in a dense DAP deployment. Higher transmission rates, generally imply higher throughput between the corresponding DAP and the client. Hence, the objective of the rate-map approach is to pick a DAP such that a client will get good throughput in both directions.

To validate our hypothesis we demonstrate a positive correlation between the RSSI of the probe request packets from the client to both uplink and downlink throughput via the following experiment. We set up a client laptop at a fixed location. The client attempts to associate with each of the 24 DAPs in turn. Prior to each association attempt with a DAP, we measure the signal strength of the client's probe request packets as observed at that node. This is the uplink signal strength (USS). After associating with a DAP, the client contacts a server on the wired network, and carries out a 2 minute TCP download, followed by a 2 minute TCP upload. We carry out this experiment from 6 different locations, and repeated the entire process 5 times. The experiment was performed on channel 64 of 802.11a band.

We found a correlation of 0.71 between USS and upload throughput and 0.61 between USS and the download throughput. These results indicate that USS can be indeed be used as a good predictor of upload and download throughput. This correlation is much stronger if we look at the throughput numbers against bucketized USS values. The rate-map approach bucketizes USS values and assigns a rate to each bucket (section 4.1.2). Figure 7 illustrates the strong correlation between throughput numbers and these bucketized USS values. The error bars indicate one standard deviation. It can be seen that the bucketized USS values are a good predictor of both upload and download throughput.

We have conducted these measurements over a number of clients and consistently found positive correlations thereby validating the hypothesis that USS values of probe packets from clients, can be used as good proxies for transmission rates between a DAP and the client.

Note that we have demonstrated a correlation between USS and uplink and downlink *throughput*. Detailed results that demonstrate a correlation between USS and *transmission rates* are available in [22].

6.2 DenseAP Performance

We now present results that demonstrate the performance of the DenseAP system. We first establish the baseline for all our experiments. We then demonstrate the overall gains achieved by DenseAP and present a series of experiments that delineate the contribution of various facets of the DenseAP system to those gains. We also demonstrate graceful degradation of the system's performance when the number of clients increases or when number of DAPs decrease. Since a large majority of the traffic in WLANs tends to be downlink [11], we only present downlink numbers for most experiments. We have conducted uplink experiments in each case, and found the uplink results to be very similar to the downlink results.

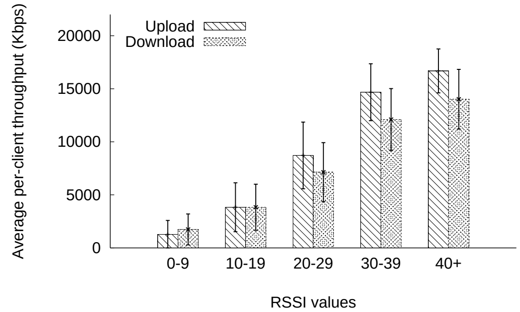

Figure 7. Correlation between bucketized uplink RSSI and upload/download throughput

6.2.1 Establishing the Baseline

We begin by evaluating the performance of the corporate WLAN, to provide a baseline against which we can compare the performance of our system.

As mentioned earlier, the testbed area is served by a single corporate AP. To establish the baseline performance, we had a group of clients associate with the corporate AP. The clients then simultaneously carried out a one minute TCP download from a server on the wired network. We varied the group size (the number of clients) from 2 to 12. The experiment is repeated 10 times for each group size. Each time, the group members are selected at random from among the available clients. We performed similar experiments for upload.

The results of this experiment are shown in Figures 8 and 9. Each point represents the median per-client throughput, and the error bars show SIQR.

For both 802.11a and 802.11g, the median per-client throughput drops as number of simultaneously active clients increases. However, the 802.11g numbers are substantially lower than the 802.11a numbers. This is because in 802.11g mode, the corporate AP sends out a CTS-to-self before every packet to avoid interfering with 802.11b clients. This is a well-known, and well-studied issue. Since we have not implemented the CTS-to-self feature for DAPs, we will refrain from directly comparing the performance of DenseAP and the corporate network in 802.11g mode.

6.2.2 Overall DenseAP Performance

We now repeat the experiment described in the previous section, but using the DenseAP system instead of

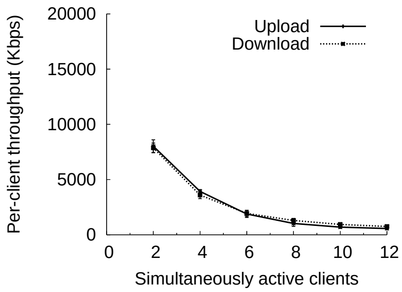

Figure 8. Baseline performance: 802.11a

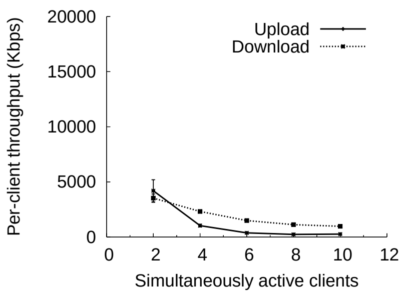

Figure 9. Baseline performance: 802.11g

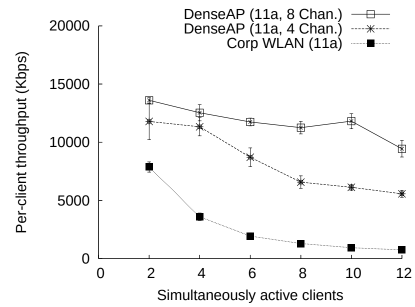

Figure 10. DenseAP performance: 802.11a

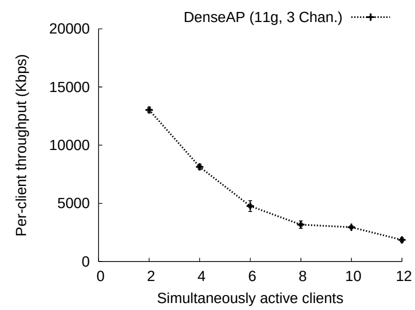

Figure 11. DenseAP performance: 802.11g

corporate WLAN. All features such as channel assignment, association policies and load balancing were enabled. We repeated the experiment twice in 820.11a mode, once using 8 channels (channels 36-64) and once using 4 channels (channels 40, 48, 56 and 64) We also ran the experiment with the DenseAP system in 802.11g mode using 3 orthogonal channels (channels 1, 6 and 11).

Figure 10 illustrates the performance of DenseAP in the 802.11a band. Figure 11 illustrates the performance of DenseAP in the 802.11g band. The graph does not have a baseline, since we do not wish to compare performance of corporate WLAN and DenseAP in 802.11g mode, as explained earlier. Let us focus on the 802.11a results.

We see significant performance gains over the corporate network. For example, with 8 simultaneously active clients, the median download throughput on the corporate network was 1.3Mbps. On the other hand, the median download throughput with DenseAP when using 8 channels, was 11.25Mbps. This represents an improvement in capacity by a factor of 868% over the corporate WLAN. Similarly, for 12 clients in the system, the median download throughput for corporate WLAN is 750 Kbps and for DenseAP it is 9.4 Mbps, which is an improvement of over 1250%.

The the comparison with the corporate WLAN may seem unfair, because we are comparing the 8-channel, 24-AP DenseAP system against a single-channel, single-AP baseline. However, the only purpose of these results is to show the full benefit of the DenseAP approach in our testbed. The next step is to separate out the impact of various factors that contribute to these results. As described earlier, the gain in throughput comes from four factors. These are: (i) use of orthogonal channels (ii) dense deployment of APs (iii) use of intelligent associations and (iv) load balancing.

We note that though enabled, the load balancing algorithm played no role in these results. The main reason is that the clients are scattered uniformly across the floor. Thus, in most cases, each client associated with its own DAP. Further, since all clients started at the same time and they all saturated their respective channels, there was no opportunity for our load balancing algorithm to move a client from one DAP to another since all channels were equally loaded. We consider the impact and efficacy of the load balancing algorithm later in Section 6.3.

It is easy to see that more orthogonal channels are better, since the median throughput is higher with 8 channels than with 4 channels. But the important question is whether the DenseAP system derives all its benefit from using more orthogonal channels? That is, can we isolate the impact of the dense deployment of DAPs and our centralized association policy?

To isolate the impact of DAP density, we need to ensure that the number of channels and the association policy play no role in the performance. The way to do this is to evaluate the performance of the DenseAP system with all DAPs operating on the same channel. This experiment is described next.

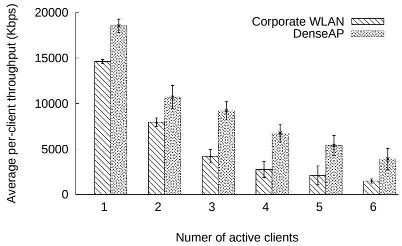

Figure 12. Benefits of density: DenseAP with 1 channel

6.2.3 Using Only 1 channel: Impact of Density

We repeated the previously described experiment with only one channel, and we varied the number of active clients from one to six. We did the experiment for the corporate WLAN, and then repeated it for DenseAP with all DAPs set to use the same channel.

Note that association policy plays very little role in this setting. Our testbed is small, and all DAPs interfere with one another. As a result, load on all DAPs is the same, so the association policy is reduced to simply selecting a DAP that hears the client with reasonable signal strength. For similar reasons, load balancing does not play a role either.

Thus, the only factor providing gains for DenseAP in this setting is the density of the DAPs. The reason density provides performance gains in this setting is the following. As more clients are added, the performance of the corporate WLAN is dominated by the client with the worst connection quality, which is usually the client that is the farthest away from the AP. Due to poor connection quality, such clients use lower transmission rates, thereby consuming more airtime. This, in turn, hurts performance of all other clients. This is known as the rate anomaly problem [13]. With DenseAP however, each client generally talks to a nearby DAP. As a result, clients and DAPs can communicate at higher data rates, thereby reducing the impact of the rate anomaly problem.

The results of this experiment are shown in Figure 12. We see that DenseAP performs better than WLAN even in this setting. The results highlight the benefit of the dense AP deployment. They also explain why, in the previous section, we saw gains of more than 800% with 8 channels!

The results lead us to ask: can a system administrator significantly improve capacity by simply adding more APs to the network? In other words, what is the contribution of the association policy to the overall gain? We examine this in the next section.

6.2.4 Benefits of Association Policy

In this section we demonstrate the benefits of the DenseAP association policy over the client-driven method of association used in conventional WLANs.

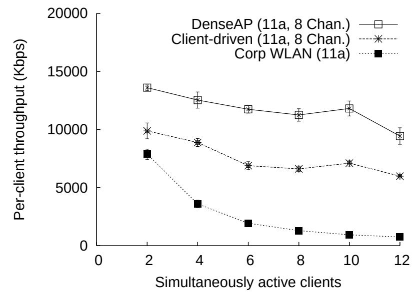

Figure 13. Benefits of the association policy

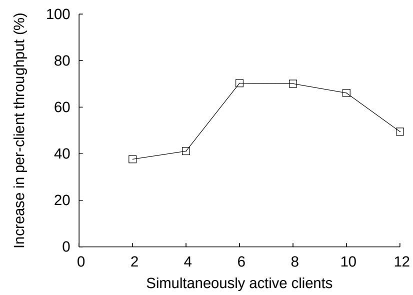

Figure 14. Gains over the client-driven approach

We carried out the following experiment in the 802.11a band, using 8 channels. We disabled the DC. We first assigned channels to all DAPs using the channel assignment algorithm described in [15]. We then disabled the ACLs, and allowed clients to associate with the DAP of their choice. In other words, the association decisions were left to the clients (as it is in today's WLANs). This setup represents a dense deployment with a conventional WLAN approach. We then carried out the experiment described in the previous section.

We note a few points about this particular scenario. There was no pre-existing traffic on any of the channels. Also, the clients were generally evenly distributed across the testbed, and so were the DAPs. Each client then picked the DAP to associate with based on the local client driver implementation policies.

The results of the experiment are compared with the result of running a full fledged DenseAP system with the same deployment and 8 channels. These results, along with the baseline, are shown in Figure 13. The results show that while simply deploying more APs and doing intelligent channel assignment in a conventional WLAN will be beneficial, the benefits will be higher if associations are controlled in a centralized manner.

In other words, the fact that the the line labeled "DenseAP" is above the line labeled "Client-driven" is what demonstrates the benefits of the DenseAP approach. The extra gain is due to the intelligent, centralized association control used in the DenseAP system. The magnitude of the extra gain is illustrated in Figure 14. In fact, as we shall see later, the centralized controller can provide roughly the same gains with fewer

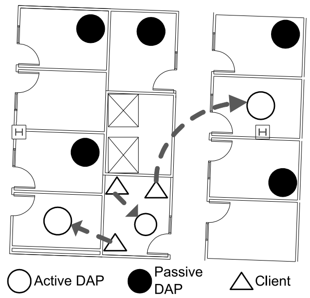

Figure 15. A conference room scenario

APs.

To drive home the point about the benefits of association policy, we consider which DAPs the clients associated with when left to decide by themselves. For example, in the case of 12 active clients, the clients used only 6 channels and 10 APs. On the other hand, by using the association policy, the DenseAP system used all 8 channels, and 11 APs.

One may argue that in the above experiment, the "Client driven" approach performed worse than the DenseAP approach simply because the specific static channel assignment we used for the "Client driven" approach was a bad one. However, we note that *any* static channel assignment algorithm that does not take into account the actual location of clients in the system, is always likely to underperform a dynamic, on-the-fly channel allocation algorithm. We demonstrate this with a simple experiment.

We set up three clients in a small conference room, as shown in Figure 15. There were no other clients in the system. We disabled DC and instead let the three clients pick the DAP to associate with. Unsurprisingly, they all associated with the AP located in the conference room. Note that no static channel assignment algorithm can remedy this situation: the clients *must* associate with different DAPs for channel assignment to have any impact. We repeated the same experiment with the DC enabled and the association policy ensured that the three clients associated with three separate DAPs.

Note however that the association policy alone is not effective. It delivers an improvement in capacity in conjunction with a higher density of DAPs. To demonstrate this, we consider the performance of the system with fewer DAPs.

6.2.5 Performance with Fewer DAPs

We repeated the experiment described in the previous section, but using only 12 of the 24 DAPs deployed in our testbed. The 12 DAPs were selected at random. We used all 8 channels. The results are shown in Figure 16. For comparison purposes, we have also included lines showing the baseline corporate network performance, and the DenseAP performance when using 24 DAPs and 8 channels. As expected, the results show that using fewer DAPs hurts performance.

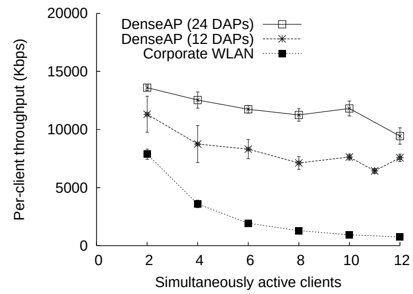

Figure 16. Performance with fewer DAPs

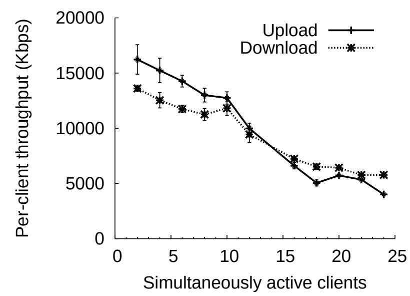

Figure 17. Performance with more clients

These results demonstrate the following: (i) More DAPs are beneficial and (ii) the performance of the DenseAP system degrades gracefully, if DAPs were to fail. We have already described how the DC can detect that a DAP has failed, and re-assign its clients to other DAPs. (iii) Note that the performance of the DenseAP system with 12 nodes is similar to the performance of the client-driven approach (Figure 13), with 24 nodes! In other words, the association policy can deliver similar performance with only half as many DAPs.

6.2.6 Performance with More Clients

In all the experiments so far, we have not used more than 12 clients. Since we have deployed 24 DAPs, it is important to consider the performance of the system with more clients. To do this, we extend the experiment described earlier to use up to 24 clients. We use all 24 DAPs, and 8 channels. The upload and download throughputs are shown in Figure 17. We see that the per-client throughput gradually decreases with increasing number of clients. This shows that DenseAP system can gracefully handle the extra load.

6.2.7 Performance with On/Off Traffic

We now turn to more complex traffic patterns as opposed to the throughput of 1-minute TCP flows. Corporate network traffic can be modeled as a series of short

	Mean Time b/w requests	Load Offered per node	Corporate WLAN	DenseAP
Overload	0.5s	2Mbps	828ms	46ms
Full load	1s	1Mbps	187ms	46ms
Low load	2s	0.5Mbps	60ms	46ms

Table 1. Performance with On/Off traffic

flows arriving at various times [12]. The metric of interest for such traffic patterns is the flow completion time.

To compare the performance of DenseAP system with corporate WLAN with such on/off traffic, we carry out the following experiment. We use 12 clients, all of which are active simultaneously. Each client downloads 2000 files from a central server on the corporate network. The sizes of files are chosen from a Pareto distribution with mean of 125KB and shape parameter of 1.5. The time between start of successive downloads is chosen from an exponential distribution with a given mean. By changing the mean time between successive requests, we can control the amount of offered load generated by each client.

We consider three scenarios. In the first scenario, the mean interarrival time between successive downloads is 0.5 seconds. This corresponds to a mean offered load of about 2Mbps per client. In Figure 8, we see that when 12 clients are simultaneously active on the corporate WLAN, the median per-client throughput is 750Kbps. Thus, the 2Mbps offered load represents a heavy overload of the corporate network. We similarly construct a fully loaded, and highly loaded scenario using mean interarrival times of 1 and 2 seconds, respectively. The details are shown in Table 1.

We repeat the experiment on corporate WLAN, as well as the DenseAP system using 24 APs and 8 channels. The median flow completion times under corporate WLAN, and DenseAP system are shown in Table 1. We see that the median flow completion time for corporate WLAN is very high under the overload and full load scenarios. These high flow completion times are detrimental to user experience. On the other hand, in DenseAP the median flow completion time is essentially equal in all three cases, since the load on the system is substantially lower than its capacity.

6.3 Load Balancing

As we have discussed earlier, in our system, a dense deployment of DAPs, coupled with the association policy limits the need for frequent load balancing. Our system uses load balancing only to correct severely imbalanced client-DAP assignments, rather than as a means to achieve "optimal" performance. The reason for this is simple: every client-AP reassignment, no matter how carefully done, carries with it the potential to disrupt a client's performance.

To illustrate the load balancing capabilities of DenseAP, we carry out the following experiment. We use three clients, situated in a conference room, as shown in Figure 15. We force the DC to initially assign all three clients to the conference room DAP. The clients start simultaneously downloading files from a server on the wired network. The DC correctly recognizes the overload situation, and at 1 minute intervals, reassigns two of the associated clients to two nearby APs, and sets them to operate on different channels. As a result, the throughput of all three clients improves substantially. The results are shown in Figure 18.

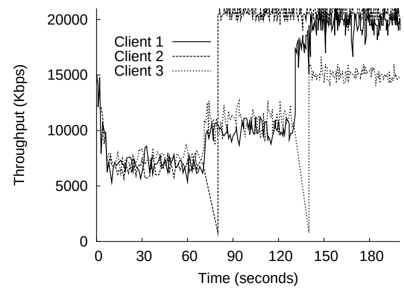

Figure 18. Load balancing: 3 TCP downloads

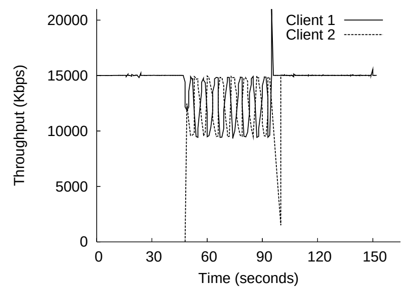

Figure 19. Load balancing: 2 CBR UDP flows

A similar scenario is shown in Figure 19. We use only two clients. We force the DC to associate both clients to the same AP. The first client starts a CBR UDP download that consumes 15Mbps. This roughly simulates streaming playback of a high quality video. However, this is not enough to saturate the DAP, and hence the DC does not move either client. At time 50 the second client starts a movie download as well. After one minute, the DC detects that an overload situation has occurred and moves the second client to a nearby DAP, and assigns it another channel. The 1-minute hysteresis interval is a tunable parameter of our system, and depending on system configuration and desires of the user population, can be set to a smaller or a larger value.

We now examine the time taken for a handoff from one DAP to another during load balancing. As we will see in the next section, it has significant implications for handling of mobile clients. Recall the sequence of steps for a handoff illustrated in Figure 3. The breakdown of time taken by each of these steps is shown in Table 2. As it has been observed in prior work [23], we see that client scanning is the most expensive step during a hand-

Step	Time (ms)
Disassociation	0
Scanning time	1487.6
Authentication	00.381
Association	00.689
Total handoff time	1488.67

Table 2. Breakdown of a typical handoff in DenseAP.

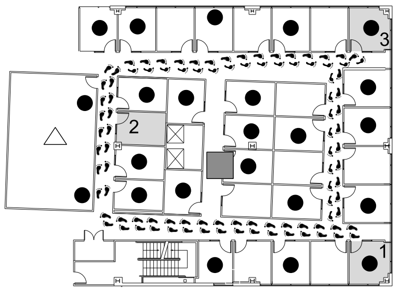

Figure 20. Locations for mobility experiment

off. The 1.5 second delay may cause a TCP timeout, but won't break an existing connection. To mitigate the impact of such disruptions, our load balancing algorithm is very conservative, and moves clients only if they are associated with an overloaded DAP. Such clients would generally be experiencing poor performance in the first place.

6.4 Mobility Experiments

In all the experiments described so far, the clients have been stationary. In this section, we consider how the DenseAP system performs with non-stationary clients. Non-stationary clients fall in two categories, nomadic and mobile.

Nomadic clients move from place to place, but spend significant time being stationary at each place. In corporate WLANs, most non-stationary clients are nomadic clients. A typical example of a nomadic client is an employee who takes her laptop to various meetings. For nomadic clients, the quality of connection they receive when they are "on the move" is less important than the quality of connectivity they receive when they are stationary.

The other type of non-stationary clients are mobile clients. A Wi-Fi VoIP phone user falls in this category. Such clients are rare in current WLANs, but are likely to become more prominent in future [18]. These clients need seamless connectivity as they move. For such clients, metrics such as delay jitter and smoothness of handoff are more important than throughput. Providing good service to mobile clients in a Wi-Fi network is an active topic of research.

Our system can handle both nomadic and mobile clients. We present results for nomadic clients here. The

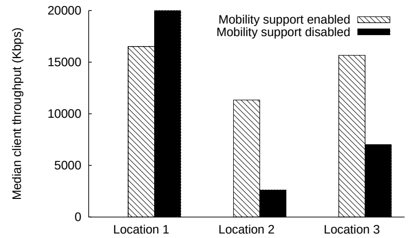

Figure 21. Performance of nomadic client

results for mobile clients are omitted due to lack of space, and are available in [22].

Our system periodically determines the location of each client, and triggers reassociation if the client's position has changed substantially. This works quite well for nomadic clients, since the clients are stationary for most of the time. We demonstrate this with the following experiment.

We setup a client at location 1 on the map shown in Figure 20, and carried out ten 2MB TCP downloads from a server on the wired network. We then walked to location 2, waited for two minutes, and carried out the downloads again. We performed a similar experiment at location 3. We repeated the experiment twice: once without mobility support and once with the support enabled.

The median throughput of the downloads, with and without mobility support is shown in Figure 21. We see that with mobility support enabled, the DC correctly reassociates the client at each location, so its performance does not suffer. Without mobility support, the client continues to be associated with the AP near location 1, and its performance suffers at locations 2 and 3.

6.5 System Scalability

Our architecture uses a central controller (the DC) to manage the DAPs. Each DAPs sends periodic reports to the DC. This raises scalability concerns. To address these concerns, we note that our DC was able to easily manage a network of 24 DAPs and 24 clients, without any special optimizations. The CPU load on the DC never exceeded 30%. We estimate that the amount of control traffic generated by each DAP was less than 20Kbps. Thus, we estimate that a slightly more powerful DC could easily handle a network of about 100 DAPs, without any special optimizations. This should be enough to cover a floor of our office building.

We note here that it is not strictly necessary to use a single central controller. What is necessary is the use of global knowledge while making association and channel assignment decisions. In theory, the functionality of the central controller can be either be replicated, or even implemented in a fully distributed manner. The DAPs can exchange information with each other to gain a global view of the network, and make appropriate decisions.

However, this approach is more complex to implement, and has its own set of scalability concerns.

Another issue we must address is the impact of several DAPs in close vicinity, beaconing and sending probe packets. Our measurements show that in the common case, the impact on performance is less than 1%. This is due to two reasons. First, only active DAPs send beacons, and second, when we use multiple channels, the number of DAPs on any one channel is small.

7 Discussion

We now discuss some issues related to the DenseAP architceture.

Density Re-visited: The density of DAP deployments affects the performance of DenseAP. This raises some important questions that need to be addressed, (i) Where should the DAPs be placed? (ii) Is there a point at which adding more DAPs to the system can hurt performance? (iii) How do we determine the minimum necessary density for a required level of service in a given environment? Guidelines developed for traditional WLANs offer little help in answering these questions, since they are generally developed with an aim of using as few APs as possible while maximizing the coverage area. Question (i) In our current testbed, we distributed the DAPs roughly uniformly in the given area. However, it may be beneficial to deploy more DAPs near "hotspots" such as conference rooms. We are studying this question further. For (ii), thus far, we have demonstrated exploiting density to yield higher gains in capacity. However, with only a finite number of channels and no power control, we expect the benefits from density to diminish beyond a certain point. Mhatre et al. [19] have presented a closed form solution for optimal AP density by varying the CCA threshold, and we are working on validating it on our testbed. To address (iii), we can integrate DAIR [8] with DenseAP to automatically determine RF Holes, i.e. regions with no coverage.

Hidden Terminal: The DenseAP system might exacerbate the hidden terminal problem due to a greater number of parallel transmissions. We have not noticed this effect in our testbed where all DAPs interfere with each other. However, hidden terminals might be a concern in larger testbeds. We are expanding our deployment to investigate this issue in detail. However, our preliminary insight is that the hidden terminal problem might not be severe in the DenseAP scenario because of the capture effect [17]. In a dense deployment of DAPs, the clients are generally located very close to the DAPs they are associate with. Furthermore, the signal in the 5 GHz band fades rapidly in indoor environments thereby reducing the interference from far-away transmitters. Therefore, we expect the capture effect to reduce the impact of hidden terminal problems.

Spatial Reuse of Channels: When assigning a channel to a DAP, our algorithm can take into account the load on all available channels. The load includes background noise, as well as traffic generated by other DAPs. Thus, we achieve spatial re-use whenever possible. Our algorithm, however, is not optimized to maximize spatial re-use.

Co-existence with Other Wi-Fi Networks: Since we can take the load on a channel into account while assigning channels to DAPs, it is easy to see that DenseAP can co-exist with other Wi-Fi networks. For example, if a nearby network is generating heavy traffic on a particular channel, the DenseAP system can detect it, and avoid assigning that channel to DAPs that are likely to be affected by that network.

What is the Ideal Client-AP Assignment?: The ideal client-AP assignment depends on several issues, including traffic, background noise and environmental factors that affect radio signal propagation. Currently, the DenseAP algorithm ignores the impact of hidden terminal issues, and focuses on avoiding problems such as rate anomaly and AP overloading. We make no claims that our algorithm is optimal. In future, we plan to study the optimality of our algorithm using simulations.

8 Related Work

There has been much prior work on WLAN channel assignment and power control. Several of them [9, 24, 14, 10, 7, 21] either require modifications to the client or to the 802.11 standard. This makes them difficult to deploy. To the best of our knowledge, ours is the first proposal to be built and deployed that performs intelligent associations and deals with a dynamic operating wireless environment *without* requiring client modifications. Of the prior work in this area, we address two systems in particular that come closest to DenseAP.

Similar to DenseAP, MDG [10] identifies intelligent channel assignment, power control and client association as being key components of a systematic approach to increase the capacity of an 802.11 wireless network. It studies the interdependencies between these three knobs and identifies various situations in which a correct order of their application can increase network capacity. Furthermore, MDG modifies clients, and uses explicit feedback and cooperation from them to perform efficient channel assignment, power control and association. In contrast to MDG, DenseAP does not require any modifications to the clients, and therefore explores a different design space.

SMARTA [7] is similar to DenseAP in that it uses a centralized server to increase the capacity of a dense AP deployment without requiring client modifications. However, it uses a different approach. The central controller builds a conflict graph among the APs, and uses this graph to tune the AP's channel and transmit power. It does not manage client associations. There are two main differences between SMARTA and DenseAP. First, DenseAP relies on correctly managing client associations. We have shown that the benefits of a dense AP deployment is limited if clients are allowed to take association decisions. We have also shown that unilateral power control (without client cooperation) can hurt the performance of the system. We also note that since SMARTA is evaluated entirely in simulations, we are unable to do a fair comparison of SMARTA with our scheme.

In [6], the authors propose using a centralized scheduling mechanism to schedule downlink traffic in a dense deployment of APs. The overall goal is to efficiently manage the data plane of an 802.11 deployment. The work is in progress and at the time of this submission, the authors have not proposed a solution for managing the uplink traffic.

A host of products by networking startup companies [3, 5, 2, 4, 1] are designed to manage AP deployments in the enterprise. The exact details about how their products work are difficult to obtain. However, most systems seem to either ignore association control and load balancing, or they address such challenges by requiring users to install custom drivers.

9 Conclusion

We have demonstrated that DenseAP improves the capacity of an enterprise network. It achieves this by exploiting DAP density via an intelligent association process that encompasses load balancing and dynamic channel allocation. We have described the algorithms and mechanisms necessary to support unmodified clients, and shown significant benefits in a real testbed deployment.

Acknowledgments

We thank Victor Bahl, Ratul Mahajan, Matt Welsh, Aditya Akella, Jon Crowcroft, Lili Qiu, Dina Katabi and the anonymous reviewers for their helpful comments on early drafts of this paper. We thank Mitesh Desai for helping with the SoftAP implementation.

References

[1] Autocell, http://www.autocell.com.

[2] Bluesocket, http://www.bluesocket.com.

[3] Enterprise solutions from aruba networks, http://www.arubanetworks.com/solutions/enterprise.php.

[4] Extricom, http://www.extricom.com.

[5] Meru networks, http://www.merunetworks.com.

[6] N. Ahmed, S. Banerjee, S. Keshav, A. Mishra, K. Papagiannaki, and V. Shrivastava. Interference Mitigation in Wireless LANs using Speculative Scheduling . In *MobiCom*, 2007.

[7] N. Ahmed and S. Keshav. SMARTA: A Self-Managing Architecture for Thin Access Points. In *CoNEXT*, 2006.

[8] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. Enhancing the Security of Corporate Wi-Fi Networks Using DAIR. In *MobiSys*, 2006.

[9] P. Bahl, M. Hajiaghayi, K. Jain, V. Mirrokni, L. Qiu, and A. Seberi. Cell Breathing in Wireless LANs: Algorithms and Evaluation. *IEEE Transactions on Mobile Computing*, 2006.

[10] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre. MDG: Measurement-driven Guidelines for 802.11 WLAN Design. In *MobiCom*, 2007.

[11] R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-based Management System for Enterprise Wireless LANs. In *NSDI*, 2007.

[12] J. Eriksson, S. Agarwal, P. Bahl, and J. Padhye. Feasibility Study of Mesh Networks for All-Wireless Offices. In *MobiSys*, 2006.

[13] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Infocom*, 2003.

[14] G. Judd and P. Steenkiste. Fixing 802.11 Access Point Selection. In *SIGCOMM Poster Session*, Pittsburgh, PA, July 2002.

[15] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed channel assignment in multi-radio 802.11 mesh networks. In *WCNC*, 2007.

[16] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth Estimation in Broadband Access Networks. In *IMC*, 2004.

[17] J. Lee, W. Kim, S.-J. Lee, W. Kim, D. Jo, J. Ryu, T. Kwon, , and Y. Choi. An Experimental Study on the Capture Effect in 802.11a Networks. In *WINTECH*, 2007.

[18] M. Lopez. Forrester Research: The State of North American Enterprise Mobility in 2006. December 2006.

[19] V. Mhatre and K. Papagiannaki. Optimal Design of High Density 802.11 WLANs. In *CoNEXT*, 2006.

[20] V. Mhatre, K. Papagiannaki, and F. Baccelli. Inteference Mitigation through Power Control in High Density 802.11 WLANs. In *Infocom*, 2007.

[21] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *Infocom*, 2006.

[22] R. Murty, R. Chandra, J. Padhye, A. Wolman, and B. Zill. Designing High Performance Enterprise Wi-Fi Networks. Technical Report MSR-TR-2008-28, 2008.

[23] I. Ramani and S. Savage. SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks. In *Infocom*, Miami, FL, March 2005.

[24] A. Vasan, R. Ramjee, and T. Woo. ECHOS - Enhanced Capacity 802.11 Hotspots. In *Infocom*, 2005.

[25] S. Wong, S. Lu, H. Yang, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *MobiCom*, 2006.