ROHAN MURTY, JITENDRA PADHYE, RANVEER CHANDRA, ALEC WOLMAN, AND BRIAN ZILL

designing high-performance enterprise Wi-Fi networks

Rohan Murty received his BS in Computer Science from Cornell University in 2005. He is currently a PhD student in the Computer Science Department (School of Engineering and Applied Sciences) at Harvard University.

rohan@eecs.harvard.edu

Jitendra Padhye received his BE in Computer Engineering from Victoria Jubilee Technical Institute, Mumbai, India, in 1992, his MS in Computer Science from Vanderbilt University in 1995, and his PhD in Computer Science from the University of Massachusetts, Amherst, in 2000. He then spent two years working at ACIRI, now called ICIR. He has been at Microsoft Research since April 2002.

padhye@microsoft.com

Ranveer Chandra is a researcher in the Networking Research Group at MSR. He completed his undergraduate studies at the Indian Institute of Technology, Kharagpur, and holds a PhD in Computer Science from Cornell University. He was the recipient of the Microsoft Graduate Research Fellowship during his PhD and his dissertation on VirtualWiFi was nominated by Cornell for the ACM Dissertation Award.

ranveer@microsoft.com

Alec Wolman is a researcher in the Networking group at Microsoft Research, Redmond. He received a PhD in Computer Science from the University of Washington in 2002. Before graduate school, he worked for Digital at the Cambridge Research Lab.

alecw@microsoft.com

Brian Zill is a Senior Research and Software Design Engineer at Microsoft Research, Redmond. Brian is the author of Microsoft's IPv6 stack as well as the Mesh Connectivity Layer (MCL).

bzill@microsoft.com

WE ARE STARTING TO SEE A SIGNIFICANT increase in the use of mobile computing devices such as laptops, PDAs, and Wi-Fi enabled phones in the workplace. As the usage of corporate 802.11 wireless networks (WLANs) grows, network capacity is becoming a significant concern. In this paper, we propose DenseAP, a novel architecture for increasing the capacity of enterprise WLANs using a dense deployment of access points (APs). In sharp contrast with wired networks, one cannot automatically increase the capacity of a WLAN by simply deploying more equipment (APs). To succeed in increasing capacity, the APs must be assigned the appropriate channels and power levels, and the clients must make intelligent decisions about which AP to associate with. Furthermore, these decisions about channels, power 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. We outline the challenges faced in solving these problems and the novel ways in which DenseAP addresses them.

In a typical office environment, it is relatively easy to deploy a wired Ethernet network. These networks are generally well-engineered and overprovisioned. In contrast, deploying WLANs in enterprise environments is still 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, it is common for WLAN users to 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, in both conference rooms and offices [12]. 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. These changes are leading enterprise network administrators to question the assumptions made during the design and deployment of their existing WLANs [1]. 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 the research community [2, 13, 14] as well as by industry. 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, yet in sharp contrast to wired networks, one cannot automatically increase the capacity of a WLAN by simply deploying more equipment. To succeed in increasing capacity, intelligent software control of the WLAN devices is needed to deal with such issues as channel assignment, power management, and managing association decisions.

We present a new software architecture called DenseAP, with the goal of significantly improving the performance of corporate Wi-Fi networks. A key emphasis in our design of the DenseAP system is on practical deployability. For example, 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, as a consequence of these concerns, we do not consider changes that require hardware modifications or changes to Wi-Fi protocols. Although these constraints do limit the design space to a certain extent, we found they also open up a set of interesting research challenges.

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, whereas with the DenseAP architecture the common case will be that APs outnumber clients. Second, in conventional WLANs clients decide which AP to associate with, whereas the DenseAP systems use centralized control of the association process.

The scarcity of APs in conventional enterprise WLANs limits their performance in a variety of ways. For example, with a large number of nonoverlapping 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 thus see better performance. A dense deployment also ameliorates the “rate anomaly” problem [8] that hurts the performance of conventional WLANs.

To fully benefit from a dense deployment of APs, the clients must associate with the right AP. In conventional WLANs clients select which AP to associate with. Typically, the association policy is implemented within the device driver for most Wi-Fi clients, and it uses only locally available information. For example, most client drivers tend to use signal strength as the dominant factor in selecting an AP, yet it is well known that this behavior can lead to poor performance [9]. For example, when many clients congregate in a conference room, they all tend to choose the same AP. To improve performance, when multiple APs are available clients must associate with different APs. In the DenseAP architecture, the central controller gathers information from all the APs and then determines which AP a particular client should associate with. Using a novel way to manipulate the 802.11 association process, the central controller ensures that a specific Wi-Fi client will only discover the AP that the controller has chosen, thus ensuring that the clients associate with this AP. Using a similar mechanism, the controller also carries out periodic load balancing by seamlessly moving clients from overloaded APs to nearby APs with significantly less load. The controller achieves all this without requiring any changes to the association control software that runs on the clients.

The DenseAP architecture is quite versatile and is capable of improving many aspects of performance of enterprise WLANs. In this paper, we focus on describing how DenseAP helps significantly improve 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 also briefly discuss how the architecture can improve other aspects of performance, such as quality of service for delay- and jitter-sensitive applications such as VoIP.

We describe the DenseAP architecture, algorithms, interesting research challenges, and open issues that arise when deploying APs in a very dense manner. These challenges include the need for appropriate channel and power assignment, ensuring that clients associate with the appropriate AP, and the need to make the system self-managing and easy to deploy. We will point out how the performance could be further improved if we could modify the end clients or count on their cooperation in some manner. We view the current DenseAP architecture not as the final word but as a practical first step toward exploring many ways of improving the performance of Wi-Fi networks.

Architecture

Figure 1 presents a high-level illustration of the DenseAP system architecture. Broadly, the system consists of DenseAP nodes (DAPs) connected to the wired network and controlled by a central server. Each DAP has a programmable software AP running on it. The DAP sends periodic reports to the DenseAP Central Controller (DC). These reports consist of the list of clients associated with it and the amount of traffic sent to or received from each client. The DC aggregates reports received from all DAPs and uses this information to send commands to DAPs to control their behavior.

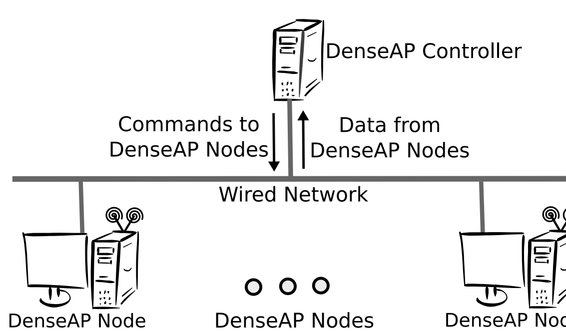

FIGURE 1: OVERALL ARCHITECTURE OF THE DENSEAP SYSTEM

In the context of this architecture, we need to answer the following three questions: First, what information does the DC need, and what "knobs" can it tune to improve the capacity of the system? Second, how densely should the DAPs be deployed? Third, is the architecture scalable and cost-effective?

Role of the DenseAP Central Controller

Given a set of DAPs and clients, the overall capacity of the WLAN depends on several factors. Since we do not wish to modify the clients, the set of performance "knobs" available to us is somewhat limited. In our current implementation, the DC attempts to improve capacity by controlling the channel each DAP operates on, the power with which each DAP transmits, and the DAP with which each client associates.

Two other knobs that can also affect the overall WLAN capacity are the Clear Channel Assessment (CCA) threshold used by each DAP and the autorate algorithm implemented on each DAP. The CCA threshold determines the level of background noise an 802.11 transmitter will consider acceptable before transmitting. If set to a high value, the transmitter is more likely to cause interference with other transmissions [13]. We do not modify the CCA threshold since most off-the-shelf wireless cards do not allow modifications to this value. The second knob is the autorate algorithm, which determines the transmission rate used by the DAP to communicate with the clients. Autorating algorithms have been studied extensively by prior research. DenseAP nodes use the autorate algorithm described in Wong et al. [17]. In the future, we plan to investigate whether the autorate algorithm can benefit from the network information gathered by the DC.

We now describe how the DC performs association and channel management in the system. We also address power control, mobility, and fault tolerance in the system.

ASSOCIATION CONTROL

In the DenseAP system, the DC decides which client associates with which DAP. This is achieved by limiting the visibility of DAPs to the clients. We first describe the mechanisms involved and then describe the algorithm used to decide which DAP is made visible to which client.

LIMITING DAP VISIBILITY TO CLIENTS

In conventional 802.11 networks, APs advertise their presence by sending out beacons, 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 message on each channel. This message is a request for APs to send out information about themselves. APs respond to a probe request message with a probe response message, the contents of which are similar to the beacon frame. Once the client gathers information about all APs, it decides which AP to associate with, and it carries out the association handshake.

The DC performs association control by limiting the visibility of DAPs to clients, exposing DAPs on a "need to know basis" to a particular client. This is achieved via two techniques. First, since 802.11 networks are identified by their SSIDs, DAPs are set to beacon with the SSID field set to NULL. Second, each DAP maintains a local access control list (ACL) of client MAC addresses that is solely managed by the DC. 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 (i.e., if the DC has previously added the MAC address to the ACL of this DAP). If a DAP receives a probe 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 reason to use hidden SSID beacons is to keep the network "hidden" and prevent clients from associating with any other DAP in its vicinity. Another alternative is for DAPs not to beacon at all; however, beacons are essential for clients to use power save mode. We have also found some client drivers that disconnect if they don't receive periodic beacons from the access point they are associated with.

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 might not even try to associate with the one the DC has chosen for it.

This method of association control has two key advantages. First, it requires no changes to the client. Second, the DC has a comprehensive view of the traffic in the network and hence can make an informed decision when choosing a DAP for a client.

We have verified that most, if not all, wireless drivers and cards available on the market today perform active scanning and hence they are able to discover DAPs. DenseAP is also designed such that if a client fails to associate with the assigned DAP (say, because of interference near the client), the DC detects this since DAPs periodically report back information about associated clients. The DC then reassigns the client to a different DAP.

DAP SELECTION POLICY

We now consider how the DC determines which DAP a client should associate with. Our intuition is to take into account both the load on the DAP and the quality of the connection between the client and the DAP. We capture this in a metric called the available capacity, which is calculated as follows: Available Capacity = Free Air Time × Expected Data Rate. Free Air Time is the fraction of time during which the DAP and the channel it is on are not busy. Expected Data Rate is an estimate of the transmission rate the DAP and the client will achieve when communicating with each other. This is primarily determined by the quality of the connection between the client and a prospective candidate DAP. The client is assigned to the DAP with the most available capacity. In other words, the DAP with the most free capacity will allow the client to send the most data, while minimizing the impact on other clients.

Channel assignment is tied into the association scheme. Since we only have a limited number of channels, those DAPs not servicing clients do not need to beacon and hence we don't assign channels to them. Therefore a DAP is assigned a channel on an on-demand basis (i.e., only when at least one client is associated with it). When a DAP does not have a channel assigned, it scans all channels and estimates load on the various channels in its vicinity. This information is sent to the DC. When assigning a channel to a DAP, the DC picks the channel with the least load.

One could propose other association policies, depending on the end goal for which the system is optimizing. For example, it can take into account the number of clients associated with the DAP, or it can try to balance the load across all DAPs. It may be possible to anticipate and factor in the future load generated by the client (e.g., demand from VoIP clients is generally predictable). We are actively exploring this research space.

We now describe how we compute the free air time at the DAP and the expected data rate. We do not expect these calculations to be precise, particularly when it comes to estimating the expected data rate. However, our intention is to provide a reasonable ordering of DAPs and to recover from any mistakes via load balancing. Hence, even if a client were to be assigned an "incorrect" DAP, the system will, at some point, hand off the client to a more suitable DAP.

ESTIMATING FREE AIR TIME

We can estimate the free air time in the vicinity of a DAP with varying degrees of accuracy. The DAP could simply add up the air time used by all packets that it has sent and received, over a unit period of time. The remaining time is the free air time at the DAP. However, such an approach ignores the effects of interference. Part of the interference can be accounted for by adding up any traffic sent or received by nearby DAPs that are on the same channel. The DC can perform this calculation using the information submitted by each DAP. A much more accurate method of estimating the free air time is for each DAP to use the ProbeGap technique, as proposed by Lakshminarayan et al. [11], and report the information to the DC. This technique directly estimates the free air time by computing the delay experienced by small probe packets. We currently use a variant of this method in our implementation. Further details are provided in our paper [15].

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 drops 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 given 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 of ranking the relative importance of the observed signal strength. We present further details in our paper [15] on this approach as well as its efficacy.

A typical wireless network tends to be dynamic in that the available capacity of a DAP will change as clients enter or leave the network and as the traffic load changes. We aim to make a reasonably good and efficient choice when initially assigning the client to a DAP and then to adapt to the changes in the environment via load balancing as we now describe.

LOAD BALANCING VIA HANDOFFS

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 used when associating new clients 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 earlier 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%—if, in other words, the channel is more than 80% busy in the vicinity of this DAP. The DC considers the DAPs in 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 at 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 by using a mechanism similar to the association mechanism described earlier [15]. Note that if B had no clients associated with it, the DC would 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 these 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 fewer 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 reassociate. Since we consider client reassociations to be disruptive events, we do not use this technique.

POWER CONTROL

We have three options when it comes to power control in the DenseAP system. The first is to perform no power control at all, the second is to perform unilateral power control at the DAP, and the third option is to perform co-ordinated power control between clients and the DAPs. We rule out the third option at present, since it requires client modifications.

In our current testbed, we use the first option: We allow all DAPs to transmit at maximum power. This increases the coverage area, but it minimizes the potential for spatial channel reuse. We are also experimenting with unilateral power control at the DAP. The idea is that, given a set of clients associated with the DAP, the DAP transmits at the minimum power necessary to provide “good” service to all clients. The “goodness” of the service is defined in terms of loss rate and transmission rate. The problem, however, is that the clients are free to transmit at any power they choose, which reduces the potential for spatial reuse. Our preliminary experiments have uncovered other problems with this option. Our results show that unilateral power control at the DAPs results in increased instances of hidden terminal and capture-effect problems. We are investigating this issue further.

MOBILITY

To handle client mobility, the DC keeps track of a client’s location, using the technique described in Chandra et al. [7]. When the location changes significantly, the system hands off the connection to a suitable DAP located nearby. The handoff is performed as previously described, and the DAP is selected using the available capacity metric. A client that undergoes handoff is considered ineligible to participate in load balancing for some period of time, to prevent oscillations. It is, however, eligible to participate in another, mobility-related handoff.

FAULT TOLERANCE

We want DenseAP to be a self-configuring and self-managing system. Hence, when a DAP goes offline or is rebooted, it no longer sends periodic load information to the DC. The DC detects this, 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 DC, which assigns these clients to other DAPs, as per the association policy.

What Is the Desired Density of DAPs?

So far we have focused on the role of the central controller. The other key factor that affects the performance of the DenseAP system is the DAP deployment density. Several important questions need to be studied in this regard. For example: (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? (iv) Since wired networks tend to have well-defined SLAs, how does one specify an acceptable level of service for WLAN?

Guidelines developed for traditional WLANs offer little help in answering these questions, since these guidelines are generally developed with the aim of using as few APs as possible while maximizing the coverage area. As such, the questions pertaining to density present interesting research challenges, and we are actively working to answer them. Mhatre and Papagiannaki [13] have explored a closed-form solution for optimal AP density by varying the CCA threshold, which in turn affects the throughput and coverage of the network.

In our current deployment, described earlier, we use ordinary user desktop machines to serve as DAPs. If this approach is followed, then question (i) need not be answered. Every desktop machine (or most of them) can serve as a DAP. However, this raises a different question: How do we know we have adequately covered the given area? Administrators of traditional WLANs use expensive site-surveying tools to determine the AP placements. Inspired by DAIR [3], we are exploring methods to automatically determine whether we have left any gaps in our coverage.

Scalability

Our architecture uses a central controller (the DC) to manage all DAPs. Each DAP sends out 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 20 kbps. Thus, we estimate that a slightly more powerful DC could easily handle a network of about 100 DAPs, without any 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 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 proximity, beaconing and sending probe packets. Our measurements show that, in the common case, the impact on performance is less than 1%. This is because only those DAPs servicing clients send beacons and because, when we use multiple channels, the number of DAPs on any one channel is smaller.

Implementation

Our implementation of the DenseAP system is deployed on a portion of our office floor. Our current testbed consists of 24 desktop-class PCs serving as DAPs. (See Figure 2.) The DAPs are deployed at a density of roughly one machine in every other office. The area is normally served by three of our corporate WLAN's APs. As illustrated in Figure 2, the current DenseAP deployment is nine times more dense than the corporate WLAN deployment.

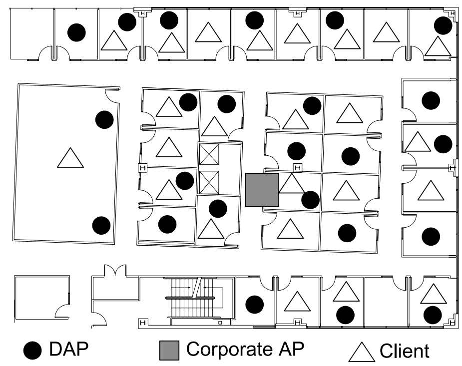

FIGURE 2: THE TESTBED. THE AREA IS ROUGHLY 32 × 35 M.

DAPs are constructed entirely from commodity hardware. We use off-the-shelf PCs. To each PC we add a Netgear JWAG511 wireless interface card. These are multiband 802.11 a/b/g radios using a RealTek chipset. The access point functionality is provided in software through a combination of a device driver and a system service. Having the AP functionality implemented in software was critical to our efforts, as it allowed us to easily modify the AP behavior to our specifications.

The DC also runs on an ordinary (but dedicated) desktop machine. All DAPs are connected to the same IP subnet on their wired Ethernet link. The DenseAP WLAN runs in the 5 GHz band (802.11a) on the lower 8 channels. The corporate network also operates in the same band.

Our paper [15] has further details regarding the evaluation of the system as well as an extensive discussion on various open questions and issues.

Related Work

Prior academic work on either improving capacity or managing dense deployments has focused on channel assignment, power control, and associations, most of which have required modifications to clients. Fundamentally, DenseAP differs from all prior work as follows:

- Practicality: Of the host of prior work in this area [4, 6, 10, 13, 14, 16], to our knowledge, DenseAP is the first system to be designed and deployed in a real-world scenario.
- Intelligent association and load balancing: To our knowledge, none of the prior proposals is capable of intelligent associations or able to deal with a dynamic operating wireless environment without requiring client modifications. In most such systems, associations tend to be static or solely driven by the clients.
- No modifications to clients necessary: Most approaches require modifications to the clients [4, 5, 6, 9, 16] or the clients to cooperate in some manner that breaks the prevalent 802.11 standards.

SMARTA [2], like DenseAP, addresses the problem of managing dense AP deployments to increase capacity or lower latency, without modifying clients. There are several key differentiators between it and DenseAP. First, SMARTA does not account for a dynamic operating environment. It lacks the ability to load-balance clients and hence the need to assign "correct" channels and power levels at the outset is greatly magnified. Second, unlike DenseAP, SMARTA relies entirely on the clients to make their own association decisions, which by prevalent standards are agnostic to network load. In a dense deployment, this approach can very easily lead to lower throughput for all clients [9]. Third, it is unclear how clients maintain persistent connections when SMARTA performs channel or power assignments. The system does not make an effort to sustain such connections at the client. Most of these differentiators arise from SMARTA having been studied almost completely in simulations.

Mhatre and Papagiannaki [13] propose varying the (CCA) threshold on APs to increase capacity in 802.11g mode. The system has been designed and studied within the confines of the Opnet simulator. Similarly, other proposals involve varying the receiver sensitivity as well as the CCA threshold [18].

A host of products by networking startup companies are designed to manage AP deployments in the enterprise. Although practical, most systems tend to ignore association control and load balancing, or they address such challenges by requiring users to install custom client drivers. Further references are provided in our paper [15].

Discussion and Conclusion

Use of the DenseAP system thus far has been focused on improving capacity in the enterprise. It can also improve other dimensions of WLAN performance such as lowering latency, separating voice and data traffic, and providing QoS. Each one of these goals entails tweaking the association and handoff policies. For example, in the case of handoffs, we could pick only those clients that appear to be experiencing low transmission rates to the DAP they are associated with. Another possibility is to try to determine client traffic patterns and aggregate all VoIP clients during the association process to a group of DAPs to provide better QoS. We are continuing to investigate these avenues with the overall goal of using DenseAP to improve WLAN performance along multiple dimensions.

REFERENCES

[1] R. Ahlawat and C. Canales, User Survey Report: Wireless LANs, North America and Europe, Gartner Report, 2005.

[2] N. Ahmed and S. Keshav, "SMARTA: A Self-Managing Architecture for Thin Access Points," *CoNEXT*, Lisboa, Portugal, December 2006.

[3] 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," *MobiSys*, Uppsala, Sweden, June 2006.

[4] 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.

[5] A. Balachandran, P. Bahl, and G. Voelker, "Hot-Spot Congestion Relief and Service Guarantees in Public-Area Wireless Networks," *SIGCOMM Computer Communication Review,* 32(1), 2002.

[6] Y. Bejerano and R.S. Bhatia, "MiFi: A Framework for Fairness and QoS Assurance in Current IEEE 802.11 Networks with Multiple Access Points," *Infocom*, Hong Kong, March 2004.

[7] R. Chandra, J. Padhye, A. Wolman, and B. Zill, "A Location-Based Management System for Enterprise Wireless LANs," *NSDI '07*, Cambridge, MA, April 2007: http://www.usenix.org/events/nsdi07/tech/chandra.html.

[8] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *Infocom*, San Francisco, CA, March 2003.

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

[10] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks," *Infocom*, Anchorage, Alaska, May 2007.

[11] K. Lakshminarayanan, V.N. Padmanabhan, and J. Padhye, "Bandwidth Estimation in Broadband Access Networks," *IMC*, Sicily, Italy, Taormina 2004.

[12] M. Lopez, The State of North American Enterprise Mobility in 2006, Forrester Research document, December 2006.

[13] V. Mhatre and K. Papagiannaki, "Optimal Design of High Density 802.11 WLANs," *CoNEXT*, Lisbon, Portugal, December 2006.

[14] V. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference Mitigation through Power Control in High Density 802.11 WLANs," *Infocom*, Anchorage, Alaska, May 2007.

[15] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing High Performance Enterprise Wi-Fi Networks," *NSDI '08*, San Francisco, CA, April 2008: http://www.usenix.org/events/nsdi08/tech/murty.html.

[16] A. Vasan, R. Ramjee, and T. Woo, "ECHOS—Enhanced Capacity 802.11 Hotspots," *Infocom*, Miami, Florida, March 2005.

[17] S. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," *MobiCom*, Los Angeles, California, September 2006.

[18] J. Zhu, B. Metzler, Y. Liu, and X. Guo, "Adaptive CSMA for Scalable Network Capacity in High-Density WLAN: A Hardware Prototyping Approach," *Infocom*, Barcelona, Spain, April 2006.