DAIR: A Framework for Managing Enterprise Wireless Networks Using Desktop Infrastructure

Paramvir Bahl[†], Jitendra Padhye[†], Lenin Ravindranath[†], Manpreet Singh[‡], Alec Wolman[†], Brian Zill[†]
[†]Microsoft Research, [‡]Cornell University

Abstract

We present a framework for managing and troubleshooting enterprise wireless networks using desktop infrastructure. The framework is called DAIR, which for *Dense Array of Inexpensive Radios*.

Prior proposals in this area either rely solely upon access points (APs) and mobile clients to monitor the wireless network, or augment them with dedicated sensor nodes. We believe that these approaches can be improved upon. One cannot cover the entire spectrum in a comprehensive manner using only APs and mobile clients. An ordinary, single-radio AP can not monitor multiple channels effectively, without adversely impacting the associated clients. Due to power constraints, mobile devices can not continuously monitor the wireless networks. Deploying dedicated sensor nodes is an expensive proposition.

Our solution is based on two simple observations. First, in most enterprise environments, one finds plenty of desktop machines with good wired connectivity, and spare CPU and disk resources. Second, inexpensive USB-based wireless adapters are commonly available. By attaching these adapters to desktop machines, and dedicating the adapters to the task of monitoring the wireless network, we create a low cost management infrastructure.

In this paper, we show how the DAIR framework is well-suited for solving many wireless management problems including detection of unauthorized access points, handling malfunctioning APs, and performance monitoring. In each case, we show how the DAIR framework takes advantage of the key attributes of the desktop infrastructure: dense deployment, stationarity, wired connectivity, and spare CPU and disk resources.

1 Introduction

DAIR is a framework for building wireless network management applications that benefit from dense RF sensing. Today's wireless LANs do not provide the same level of service as is currently provided by wired networks. We believe that better wireless management tools are an important step towards achieving that goal. Our initial target for DAIR is the enterprise: by building wireless management tools we believe we can significantly improve user productivity by reducing the costs of operations, increasing the reliability, and improving the security of enterprise wireless LANs.

The investment that a large corporation such as Microsoft makes into wireless infrastructure is substantial: Microsoft's current wireless network consists of approximately 5000 access points (APs); it supports 25,000 users each day in 277 buildings, covering more than 17 million square feet [4]. Beyond the equipment costs, the costs of planning, deploying, and maintaining wireless networks are also substantial. Thus, it is important to develop infrastructure that improves the ability of Information Technology (IT) departments to manage their wireless networks.

The DAIR approach is based on two observations. First, in most enterprise environments one finds plenty of desktop machines. The machines are generally stationary and are connected to wall power. They have good wired connectivity, spare CPU cycles, free disk space, and high-speed USB ports. Second, inexpensive USB-based wireless adapters are readily available and prices continue to fall[1]. By attaching USB-based wireless adapters to desktop machines, and dedicating the adapters to the task of monitoring the wireless network, we create a low cost monitoring infrastructure.

The low cost of the USB adapters provides us with a key advantage: dense deployment. The effectiveness of any management solution for wireless networks depends upon the ability to perform RF sensing from a large number of physical locations. Our solution provides a low-cost way of densely deploying RF sensors that cooperatively perform various management and diagnosis tasks.

The second advantage of our approach is that in a corporate environment, desktops are usually stationary. The stationarity allows us to ensure that coverage of the area being managed is adequate. Additionally, it also eases the problem of location determination, which is a useful technique for pinpointing many wireless management problems. Finally, the stationarity of the sensors allows our wireless management system to maintain meaningful histories of wireless network observed seen at specific locations.

The third advantage of our approach is that desktop machines generally have good wired connectivity. As we shall see later in the paper, having access to the corporate wired network is critical, and allows us to do a better job of monitoring and diagnosing the wireless network.

The final advantage of our approach is that apart from providing spare CPU cycles, the desktop machines also offer access to wall power (and hence no power constraints). This permits more comprehensive monitoring of the wireless network.

In this paper, we present an overview of the DAIR architecture and outline several management applications we are building using the DAIR framework. In each case, we show how the DAIR framework takes advantage of the key attributes of the desktop infrastructure: dense deployment, stationarity, wired connectivity, and spare CPU cycles.

[1]On July 28th, 2005 at http://www.anandtech.com/, we found a sale price of $6.99 for an 802.11g USB adapter.

2 Design and Architecture

In this section we provide a brief description of the DAIR framework. We skip many details, since the focus of the paper is on wireless management applications that use the DAIR framework. DAIR is designed for enterprise environments – we assume that the primary users of desktop computers do not require an incentive to have their machines participate in a corporate-wide system that is deployed to improve the performance and reliability of the wireless network. The architecture of the DAIR framework conforms to the following five guidelines:

(1) Light Monitoring Load: Since the monitoring nodes are employee desktop computers, it is imperative that monitoring occur only if it does not adversely impact the computer user's experience. A corollary to this requirement is that DAIR should adjust gracefully to failures or stoppage of any monitoring nodes.

(2) Secure: As new pieces of software are added on employees work machines, DAIR should not add new security vulnerabilities. The communication between the various DAIR components should be authenticated and encrypted. We assume that users do not have administrative privileges on their desktop machines, so they can not interfere with the DAIR system.

(3) Low Cost of Deployment: DAIR should be easily and rapidly deployable. Ideally, the corporate IT department should be able to bring the system on-line by simply asking the users to plug a USB wireless dongle into their computers and then remotely installing the software.

(4) Remote Management: DAIR should be easy to configure and control remotely. It should not require any attention from the owner of the desktop machine. When a problem is detected, the IT department should be automatically alerted with minimal false alarms.

(5) Scalability: DAIR should scale easily for large wireless networks. The DAIR software should allow multiple instances of each component to avoid bottlenecks.

A high-level illustration of the DAIR architecture is provided in Figure 1. The DAIR system has two kinds of monitoring nodes, *AirMonitors* and *LandMonitors*. The AirMonitors are ordinary desktop computers belonging to employees equipped with inexpensive USB wireless cards. AirMonitors monitor wireless traffic that is "in the air". The LandMonitors are computers that monitor traffic on wired networks. For example, a LandMonitor may be used to monitor DHCP requests on each subnet. LandMonitors are not as densely deployed as AirMonitors.

The data gathered by the monitoring nodes is stored in one or more of the database servers. It is analyzed by one or more *Inference Engines*. The inference engines control the monitors by assigning them requests from several different inference engines. The ability to perform multiple monitoring tasks at the same time is fundamental to ensure scalability of the DAIR architecture.

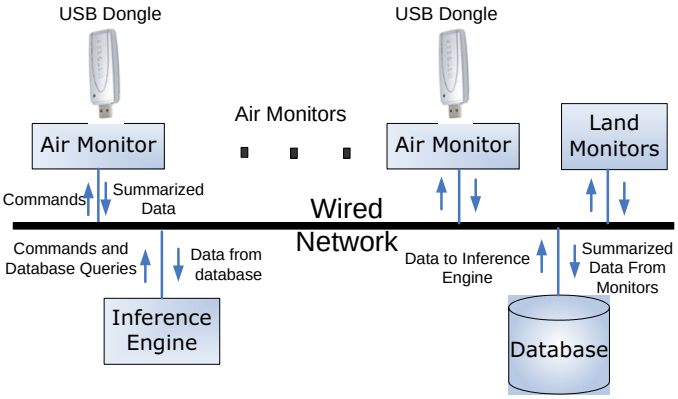

Figure 1: DAIR Architecture

Before accepting a request from the inferencing engine, the monitor checks to see if it can fulfill the request. For example, if an AirMonitor receives a new request to monitor a specific channel different from the one it is already monitoring, it will refuse that new request. Similarly, if the AirMonitor determines that the additional request will place undue burden on the host, it will refuse the request. While the precise definition of what constitutes undue burden varies based on circumstances, parameters such as history of CPU and memory usage are taken into consideration [12].

The monitor nodes filter and summarize the data before reporting it to the database. For example, if an inference engine is interested in monitoring the presence of unauthorized APs on a specific channel, it will issue a request to the AirMonitors to switch to that channel, and periodically report all the unique SSIDs (wireless network names) and BSSIDs (MAC addresses of APs [6]) that they have heard. The inference engine can then look through the data to detect unknown SSIDs or BSSIDs that may signal presence of unauthorized APs.

The monitor nodes are not limited to passive observations. They can also send packets. For example, the inference engine may request one of the AirMonitors to attempt to associate with an unknown AP in order to gather more information. This requires the AirMonitor node to send association requests and to process incoming responses.

3 DAIR Applications

In this section, we provide a high-level overview of a number of wireless management applications that we plan to build using using the DAIR framework. We will also describe one of these applications in detail.

3.1 Overview of DAIR Applications

We have identified several applications that can leverage the DAIR platform. Below, we provide a brief description of each application, and describe how these applications take advantage of the key attributes of DAIR the platform.

Rogue Access Point detection: One common security problem facing large organizations is the attachment of unauthorized (or rogue) APs to corporate networks [1, 2]. Once an unauthorized AP is set up, unauthorized clients may gain access to the corporate network without having physical access to the premises of the corporation. Thus, the ability to detect these unauthorized or "rogue" APs is a "must-have" feature for any wireless network management system. We will describe the problem in more detail in Section 3.2, and present several detection techniques that leverage the stationarity and the wired connectivity of the desktop infrastructure.

Rogue ad-hoc network detection: The problem of detecting rogue ad-hoc networks is similar to that of rogue AP detection. The key distinction that the unauthorized device attached to the wired network is operating in ad-hoc mode and acting as an IP forwarder between the wired network and the ad-hoc wireless network [7]. We use techniques similar to those presented in Section 3.2 to detect the presence of rogue ad-hoc networks.

Helping disconnected clients: A mobile client may find itself unable to connect to a wireless network for a variety of reasons. First, it may be out of range of all the corporate APs. Second, the AP may be rejecting association requests due to overload. Third, the AP may not be able to hear the client because of noise near the AP. Finally, the client may not have the appropriate security certificates installed.

In [7], the authors proposed a mechanism called Client Conduit to allow limited network access to such disconnected mobile clients. They use nearby connected mobile clients as relay nodes to tunnel the packets to the wired network. We can provide similar service using our wired AirMonitors. The AirMonitors can also maintain a log of failed association attempts by mobile clients. Such logs can be useful to pinpoint RF holes and aid future placement of APs.

Note that the stationarity and the wired connectivity of the desktop infrastructure play a key role in our solution to this problem. Since the desktops are stationary, they can maintain meaningful past history of performance of nearby APs. The wired connectivity of the desktops is leveraged to provide temporary access.

Network Performance Monitoring: Since the AirMonitor nodes are stationary, and they do not rely on the wireless interface as their primary means of communication, they can easily collect meaningful long-term historical statistics about performance of the wireless network. The performance data can be gathered using both passive measurements and active probing. Passive measurement can include numbers such as average utilization of different channels, number of unique clients seen, signal strength from nearby APs, average rate at which different clients connect to different APs etc. The AirMonitor nodes can also perform regular active measurement by associating with nearby APs to measure parameters such as throughput and loss rate. This data can be used to drive long-term network planning activities such as determining placement and channel assignment of APs. Neither the APs nor the mobile clients are in a position to collect such extensive network performance data.

Detecting Denial-of-Service attacks: Several vulnerabilities in the 802.11 management and media access services have been recently pointed out [10, 11]. DoS attacks on 802.11 networks are usually implemented by circumventing the normal operation of the firmware in commodity 802.11 devices. For example, an attacker may spoof a deauthentication packet. This will force both the client and the AP to exit the authenticated state. Another possibility is to send a packet with a very large NAV value [6]. The large NAV value will force other stations in the area to withhold their transmissions for extended periods of time. By deploying multiple AirMonitor nodes inside an enterprise, and correlating the traffic observed by various AirMonitors, we can detect such DoS attacks. Once again, we point out that the stationary nature of the AirMonitors and the fact that they do not reply on the wireless interface as their primary means of communication allows them to perform this task easily and comprehensively.

Fast hand-offs and roaming: A lot of research has gone into the problem of ensuring a smooth handoff when a mobile client moves from one AP to another. Many of the proposed solutions depend on being able to predict the client's movement trajectory [20]. The DAIR framework can enhance the trajectory prediction process. Since the AirMonitors themselves are stationary, and have wired connectivity, they are well-equipped to track a mobile client. The tracking data can be used to enhance trajectory prediction, and hence a better handoff experience for the client.

Recovering from malfunctioning APs: The DAIR infrastructure can serve as a backup service for the wireless network infrastructure. If an AP crashes or starts experiencing performance problems, a nearby AirMonitor can temporarily take over the job of providing wireless connectivity to the mobile clients in that area.

In summary, we have shown that the DAIR framework allows us to build applications that address several important problems in wireless network management. We now describe one these problems, namely, the rogue AP detection problem, in more detail.

3.2 Detection of Rogue Access Points

There are many scenarios scenarios whereby rogue wireless equipment may be connected to a corporate network. For example, an employee might bring in a wireless AP from home, plug it in to the corporate network without configuring it to require the necessary authentication. Or, a disgruntled employee may deliberately attach an unauthorized AP to the corporate network. Note that once an unauthorized AP is attached to the corporate network, the security of the network is compromised even if all the *authorized*

APs are configured to use appropriate authentication mechanisms. Thus, detecting these unauthorized or "rogue" APs is an important challenge.

It may appear at first glance that to solve this problem, an organization simply needs to maintain a database of all authorized APs, including their SSIDs and BSSIDs. An alarm is raised whenever an unknown SSID or BSSID is heard by a wireless sensor – this sensor can be an AP, a mobile client, or a dedicated sensor node. This is the basic mechanism proposed in previous research [7], and many wireless management companies offer rogue AP detection as part of their product offerings [1, 2]. Unfortunately, this straightforward approach is susceptible to both false negatives and false positives. We now discuss how the DAIR architecture help us improve upon the basic approach.

3.2.1 Guarding against false negatives

A malicious user may configure a rogue AP to advertise the same SSID and BSSID as one of the authorized AP devices, in which case the above simple strategy will not flag a problem. To guard against such false negatives, DAIR uses the observed signal strength of packets received at the different AirMonitors to determine the approximate location of the device in question. DAIR uses this information, along with the fact that the 802.11 beacon sequence numbers are different [6], as indication that there are multiple devices pretending to be one. DAIR also uses historical information to assist with this process: for example, a set of AirMonitors suddenly hears an "authorized" AP with strong signal strength, when for the past three months they have never heard that AP. At this point, the network administrator can use the location information to look for the rogue AP, as the location of the legitimate AP is known. Both the stationarity of the AirMonitors and their ability to continuously monitor the wireless spectrum allow DAIR to gather the historic data necessary to eliminate this type of false negative.

3.2.2 Guarding against false positives

In many office buildings, one is likely to overhear APs deployed by other nearby corporations. The fact that a sensor can hear an AP that is not in the database of authorized APs is not necessarily a cause for alarm. DAIR prioritizes the alarms by determining whether the "suspect AP" (hereafter referred to as the *suspect*) is attached to the corporate network. If DAIR determines that the suspect is indeed attached to the corporate network, the alarm is assigned a higher priority. While it is not always possible to definitively determine whether the suspect is connected to the corporate network, we have implemented a number of tests to answer the question in many situations.

Before describing our tests, we note that the term "Access Point (AP)" is used rather loosely in practice. As per the 802.11 standard, the AP is a device that acts as a bridge between the wireless network and the wired backhaul. In other words, it is a "layer 2" device, like an Ethernet switch. This is the functionality that most commercial-grade APs provide. On the other hand, the wireless devices designed for home networking are generally called *wireless routers*, which combine AP and router functionality, usually along with NAT capabilities. The importance of this distinction will become clear later in the section.

We first describe a test that can reduce false positives regardless of whether the suspect is an AP or a wireless router. Then we describe two tests that are useful when the suspect is really an AP. We then consider the case where the suspect is, in fact, a wireless router.

Association Test

To determine if the suspect is connected to the corporate network, one of the AirMonitor nodes attempts to associate with it. If the association is successful, the AirMonitor then attempts to communicate with (e.g. ping) one or more well-known entities that are only accessible from within the corporate network. If this test succeeds, then we know the suspect is attached to the wired network. If the attempt to associate or ping fails, perhaps because the AP has MAC address filtering or WEP enabled, then we must run more tests.

The Suspect is an Access Point

MAC Address Test: This test is used when an AirMonitor can hear data packets that are either destined to or transmitted from the suspect. These packets can yield clues about whether someone is using the suspect as an entry point to the corporate network. If the packets are not encrypted, we look at the destination IP addresses to see if any device associated with the suspect is communicating with hosts inside the corporate network. If the packets are encrypted, we look at the source or destination MAC address of these packets. If a device associated with the suspect is communicating off the subnet that the suspect is connected to, then the destination (or source, depending on direction of communication) MAC address in their packets will be the MAC address of the subnet router. To implement this test, we need a database of the MAC addresses of routers within the corporate network. This table can be automatically filled in by the AirMonitors. Due to space constraints we omit the details of this process.

ARP Test: The MAC address test only handles the case where a device associated with the suspect is communicating with an entity off the local subnet. To handle the case where communication is only within the local subnet, DAIR uses an ARP LandMonitor that listens for ARP requests which are broadcast on the wired network. Remember that with switched ethernet, it is often not easy to observe arbitrary traffic on the wired network, but it is always easy to observe traffic sent to a broadcast address, as long as the listener is on the same subnet. The ARP LandMonitor periodically summarizes the list of MAC addresses that issued the ARP requests, and submits those summaries to the central data collection server. Whenever an AirMonitor detects the MAC address of a device that is communicating with the suspect, it checks whether or not that MAC address is

on the list of MAC addresses that have been seen issuing ARP requests on the wired network. If so, then the suspect is attached to the wired network.

The Suspect is a Wireless Router

Another challenging scenario arises when someone attaches a wireless router to the corporate network. The problem is that these devices break the previous two tests. When the AP and IP routing functionality are implemented in the same device, the destination MAC address of the wireless traffic will simply be the wired MAC address of the wireless router. Furthermore, any ARP requests that go out on the wired network will be just using the source MAC address of the wireless router, not that of the wireless device associated with the router. To handle the case of wireless routers, we have two additional tests.

DHCP Signature Test: A wireless router device that wants to communicate with other devices on the wired network is likely to issue a DHCP request shortly after it is plugged in to the wired network. We use a DHCP LandMonitor which listens to broadcasts of DHCP requests on the wired network. We detect the type of device that issues the DHCP request by parsing the contents of the DHCP requests. Our studies indicate that contents of the DHCP option field can be used as a fingerprint to determine the type and the manufacturer of the device that issued the request. For example, we can distinguish between requests that come from a Windows clients, and those that come from wireless routers. In many cases, we can also determine the manufacturer of the wireless router (e.g. DLink, NetGear etc.) If a LandMonitor detects a DHCP request whose fingerprint does not match any of the device types that are usually connected to the corporate network, then it can raise an alarm.

Correlation Test: Our final test is perhaps more reliable, but significantly harder to deploy. For this test, we correlate packets sent on the air with packets sent on the wired network, using both the length of the packets and the times at which they were sent. By observing the same traffic on the wireless and wired networks, DAIR can detect that the suspect is attached to the wired network. The key problem with this technique is visibility: ideally you either need ethernet repeaters rather than switches, or you need to enable port mirroring on the ethernet switch that the device is directly attached to. In other words, the challenge is getting access to unicast traffic on the wired network generated by the wireless router, especially when you don't know where or even whether the device is attached to your wired network.

3.2.3 Summary

We have shown that the seemingly simple problem of detecting rouge APs is, in fact, quite challenging. We also described how the DAIR architecture leverages the unique attributes of the desktop infrastructure to limit the number of false negatives and false positive alarms. Our techniques are not foolproof, and we do not guarantee that a suspect is *not* connected to the corporate network. However, we do provide the network administrator with more information, without many false positives and false negatives.

4 Related work

Network diagnostics and management is an active area of research. Much of the published literature has focused on wired networks in general, and on wide area Internet failures in particular [22, 21, 16, 13]. The problem of detecting and diagnosing faults in wireless networks has received comparatively less attention from the networking research community. Recently, the problems associated with securing and managing wireless networks have become more prominent [10, 11, 8], and there is a lot of commercial interest in this area.

There are numerous commercial offerings in the area of wireless network management [3, 1, 2, 5]. Most products use one of the two approaches. They either rely on APs for monitoring, or to use dedicated and often expensive custom hardware sensors for RF monitoring. Very few details of algorithms or heuristics used by these products are available in the marketing literature provided by these companies. As we described in Section 3.2, there are many different levels of sophistication that one can provide when solving the security problems of rogue wireless equipment.

Some commercial products [3] rely on APs for monitoring wireless networks. This approach is certainly cost effective, but it has several limitations. First, a single-radio AP can not easily monitor multiple channels, or associate with other nearby APs, since its primary function requires it to spend most of its time on one specific channel serving associated clients. Second, the APs usually have limited CPU power and memory resources, compared to desktop machines, so we cannot poll them (i.e. issue SNMP queries) too frequently. Third, the level of detail that typical commercial APs provide with their SNMP counters is quite limited. Fourth, APs tend to be closed platforms so one cannot load and run third-party code on them, making it difficult to quickly deploy new functionality. Finally, an AP only provides a view of one end of the wireless communication, so an AP-based solution can not be used to detect problems such as RF holes or excessive interference that primarily affect client end of the communication. To overcome these limitations, some vendors [1, 2] augment the AP-based monitoring by deploying special sensor nodes throughout the organization. However, such specialized sensors are expensive.

Two previous research efforts have proposed addressing similar problems [7, 19]. The key differentiator of our work is the approach to deployment. In [7], mobile clients were expected to perform the majority of the management tasks. It was not clear how an IT manager could be assured of reasonable coverage at any given point in time. In [19], APs are expected to perform additional monitoring functions to detect greedy or malicious behavior in hotspots. Compre-

hensive coverage of wireless spectrum using single-radio APs is not feasible because the APs must primarily use their wireless interface for the task of serving associated mobile clients. Multi-radio APs can overcome this limitation to some extent. However, the DAIR approach can provide much higher density of RF sensors.

Several research papers that monitor and characterize the behavior of wireless networks rely on polling of APs [15, 14, 9]. Although this is a useful way of obtaining data, and in fact, in our architecture we plan to use this approach to augment and validate the information collected by our AirMonitors, it suffers from the disadvantages that we have discussed earlier.

While our paper is focused on diagnosing faults in infrastructure wireless networks, researchers have also proposed diagnostic systems for ad-hoc wireless networks. For example, Qiu et. al. [18] present a system in which nodes in a multi-hop ad-hoc network gather trace data that is later analyzed with a simulator to detect faults and perform root-cause analysis. Marti et. al [17] propose a watchdog mechanism to detect network unreliability problems stemming from selfish nodes in an ad-hoc network.

5 Conclusion

We presented DAIR, a framework for monitoring and diagnosing faults in enterprise wireless networks using existing desktop machines. The DAIR architecture takes advantage of the key attributes of the desktop infrastructure: dense deployment, stationarity, wired connectivity, and spare CPU and disk resources. We described in detail how DAIR leverages the desktop infrastructure to reduce false negatives and false positive alarms when tackling the problem of detecting rogue APs. We also described how the DAIR framework can be used to address several other wireless network management problems.

We have started building the DAIR system. At present, we have a small deployment of AirMonitor nodes, equipped with NetGear WG111U USB wireless adapters. We have implemented most of the tests required for rogue AP detection, as well as some support for network performance monitoring. Our initial results from this small deployment are quite encouraging. In the near future, we hope to expand this small initial deployment to cover an entire floor of our office building.

References

[1] AirDefense: Wireless LAN Security. http://airdefense.net.

[2] AirTight Networks. http://www.airtightnetworks.net.

[3] AirWave Management Platform. http://airwave.com.

[4] Private communication with Microsoft IT depratment.

[5] Symbol Technologies: SpetcrtumSoft Wireless Management System. http://www.symbol.com.

[6] *IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification, P802.11*, 1999.

[7] A. Adya, P. Bahl, R. Chandra, and L. Qiu. Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks. In *MOBICOM*, 2004.

[8] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self Management in Chaotic Wireless Deployments. In *MOBICOM*, 2005.

[9] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *MOBISYS*, 2003.

[10] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, 2003.

[11] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker. Security flaws in 802.11 data link protocols. *Communications ACM*, May 2003.

[12] J. R. Douceur and W. J. Bolosky. Progress-based regulation of low-importance processes. In *SOSP*, 1999.

[13] A. Feldmann, O. Maennel, Z. Mao, A. Berger, and B. Maggs. Locating internet routing instabilities. In *SIGCOMM*, Sep 2004.

[14] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *MOBICOM*, 2004.

[15] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *MOBICOM*, 2002.

[16] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. In *SOSP*, Oct. 2003.

[17] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MOBICOM*, 2000.

[18] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Troubleshooting multihop wireless networks (extended abstract). In *SIGMETRICS*, 2005.

[19] M. Raya, J.-P. Hubaux, and I. Aad. DOMINO: A System to Detect Greedy behavior in IEEE 802.11 Hotspots. In *MOBISYS*, 2004.

[20] M. Shin, A. Mishra, and W. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *MOBISYS*, 2004.

[21] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *SIGCOMM*, 2005.

[22] R. Teixeira, T. G. Griffin, G. Voelker, and A. Shaikh. Network sensitivity to hot potato disruptions. In *SIGCOMM*, 2004.