Challenges in Deploying Steerable Wireless Testbeds

Eric Anderson, Caleb Phillips, Gary Yee, Douglas Sicker, and Dirk Grunwald

CU-CS-1058-09 December 2009

University of Colorado at Boulder

Technical Report CU-CS-1058-09
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Challenges in Deploying Steerable Wireless Testbeds

Eric Anderson, Caleb Phillips, Gary Yee, Douglas Sicker, and Dirk Grunwald

December 2009

Abstract

Phased array antennas enable the use of real-time beam-forming and null-steering to further increase control of signal and interference in wireless networks. Understanding the potential of this platform for both wireless mesh networks and single-hop networks is becoming more important as smart antennas begin to emerge in networking standards such as IEEE 802.11n and 802.16. Prior attempts to test non-standard antenna platforms have typically focused around simulations, fixed directional antenna testbeds that are unable to perform null-steering, and small scale temporary setups utilizing 1 or 2 phased array antenna nodes over the span of a few hundred meters.

This paper presents the challenges encountered – and solutions developed – in building WART, a permanent, campus-wide testbed for wireless networking with beam-forming antennas. We use affordable commercial off-the-shelf (COTS) hardware as both a measurement apparatus and the system under test. This approach makes it possible to develop and test networking protocols using equipment similar to what may be available operationally, but also presents difficulties beyond those typically encountered with specialized measurement hardware. We show that *system-level* techniques can adequately overcome those component limitations.

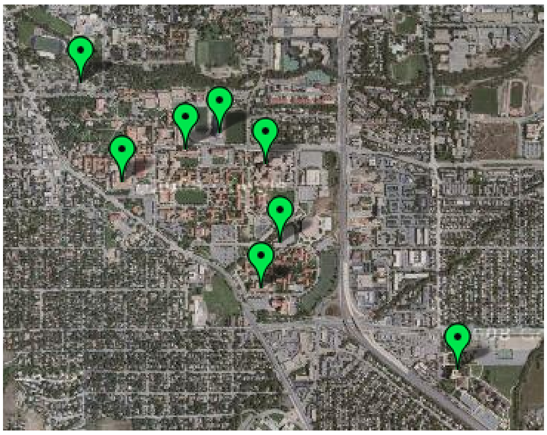
1 Introduction

Directional antennas, both fixed and steerable are emerging in the next generations of wireless networking because of their ability to give each node further control over signal strength and interference. Protocols incorporating directional or smart antennas have been proposed, but evaluation has been limited. Some work has used an ad-hoc experimental environment assembled to perform a small number of experiments [1], often at unrealistic scales [2, 3], while most rely solely on simulation or analysis.

In this paper we introduce the University of Colorado Wide-Area Radio Testbed (WART) as a platform for studying uses of directional, steerable, and smart antennas in wireless networking. Given the widely-recognized difficulty of accurately simulating radio environments, real-world experiments are essential for fully understanding wireless networking. The effects of antenna configuration are especially dependent on the vagaries of radio propagation, so physical fidelity is particularly important for this area of research [4].

WART is currently the only permanent facility for studying smart antennas over a significant area. The system consists of eight phased array antenna nodes mounted to the rooftops of the university and spans an area of 1.8 x 1.4 kilometers. The entire testbed is linked together via wired Ethernet and can be controlled from a single administration point. This architecture ensures that WART can not only offer the geographic scale and realism of large scale distributed testbeds [5], but can also give its users the degree of control and ease of management only seen in dense indoor testbeds such as ORBIT and Emulab [6, 7].

The production and deployment of such a testbed, however, is itself an engineering problem. In addition to describing the capabilities of WART, this paper describes some of the logistical challenges encountered in planning, installing, and maintaining a centrally controlled wide area rooftop network.



(a) Campus Testbed (1.8 x 1.4 km)

(b) Installed Antenna Node

1.1 Design Goals

WART is intended to be a dedicated experimental testbed for studying the impact of omni-directionality, directionality, null-steering and beam-forming throughout the network stack. Given this objective, there were three design goals for WART:

- The testbed must be able to perform outdoor omni-directional, fixed directional, and beam-forming experiments.
- The testbed must be able to test a diverse set of link distances of varying link qualities.
- WART nodes must be simple to reconfigure for varying experiments and provide an easy recovery mechanism in case of failure.

The environment chosen was the rooftops of several tall buildings at the University of Colorado, Boulder. These sites were chosen to provide a variety of link lengths, and line-of-sight between most, but not all, pairs of nodes. It was important to get a number of long links in order to study links with lower signal strengths at varying transmit powers. Note that this is in contrast to producing weak links by decreasing transmit power, which is only an approximation of long links. An indoor setting or an environment with a large number of reflections would not have been as appropriate for our directional studies due to the significant effect that would have had on beam patterns[8].

The remainder of this paper describes the hardware, software, and centralized architecture of WART that helps fulfill the design goals of easy maintenance and administration.

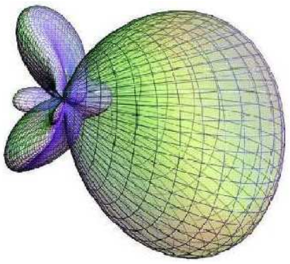


Figure 1: Unidirectional Pattern

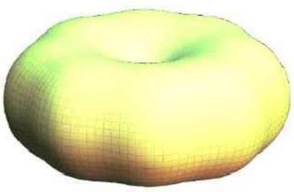


Figure 2: Omnidirectional Pattern

1.2 Smart Antenna System

In this section we describe the hardware and software that comprise WART. These components give it the unique ability to perform smart antenna research at all network stack levels and address challenges with its administration and experimental setup.

1.2.1 Hardware

Each smart antenna node consists of two major components: the phased array antennas and the embedded computer.

The phased array antennas used in our study were designed and constructed by Fidelity Comtech. The antenna operates in the 2.4GHz ISM band and uses an 8 element uniform circular array of dipole antennas that support a minimum 42 ° primary lobe when configured for a unidirectional pattern, as shown in Figure 1. The tight unidirectional pattern has a primary lobe gain of 18dBi. Additionally, the ratio of the lowest null to the highest peak is $\approx$ 40dB, which allows for selectively "nulling out" interfering signals.

Each dipole is controlled by a vector modulator which in turn is controlled by a distinct embedded processor. Intrinsic antenna reconfiguration time is $\approx 10\mu$seconds, although the interface with the transceiver boards limits the effective reconfiguration time to $\approx 100\mu$seconds. The transceiver boards are controlled by a series of phase-amplitude settings stored in flash memory, which allows fast reconfiguration between set patterns. For example, the antenna can quickly change the direction of the pattern shown in Figure 1, or switch to the omnidirectional pattern in Figure 2, by indicating the pre-computed configuration to be used.

The embedded computer is a single-board computer (SBC) based on the Intel XScale IXP425 processor. The entire system runs off 128 MB of memory and thus relies on the wired network connection for reading/writing to a long term storage device. The wireless interface card used is a Senao 5345MP MiniPCI adapter, which uses an Atheros chipset. The combined antenna and embedded computer is roughly 26x23x23 cm in size and can be mounted on vehicles, light poles and buildings.

1.2.2 Software

The default image used by each WART node is a standard OpenWRT Kamikaze distribution with some modifications to the default wireless drivers and startup scripts. This Linux distribution was selected because of its maturity, support for the embedded IXP425 processor, and standard tools such as python and tcpdump. The wireless driver is based on the Multi-band Atheros Driver (MADWiFi) version 0.9.4.5 and is augmented to support the smart antennas ability to change antenna patterns. Lastly, NFS is used to transmit data from the smart antenna node to long term storage.

2 Commodity Hardware as a Research Platform

In this section, we discuss limitations of commodity hardware with respect to research applications and the solutions we have developed to mitigate them. Principally, we want to:

- Be confident in the fidelity of physical-layer (PHY) measurements and settings
- Implement and study experimental medium access control (MAC) protocols
- Have precise control of timing and strict clock synchronization

2.1 Received Signal Strength Accuracy

To ensure that it is safe to use commodity IEEE 802.11x-based hardware to measure signal and interference levels, we calibrated the sensitivity of our radios against known signal sources.

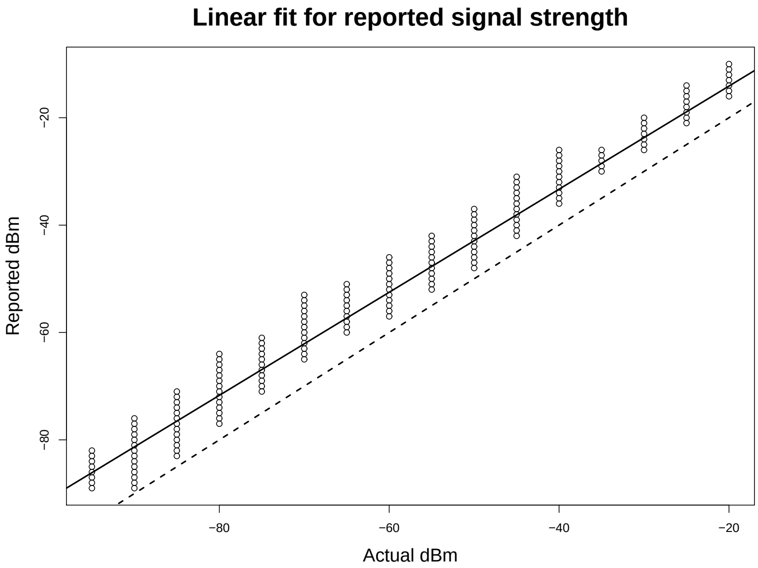


Figure 3: Linear fit of reported versus actual signal strength on commodity cards during calibration. The solid line indicates the regression fit, and the dashed line is perfect equality.

To get an idea of how accurate our commodity radios are in measuring received signal strength (RSS), we directly connected each of our radio cards to an Agilent E4438C vector signal generator (VSG). The VSG was configured to generate IEEE 802.11 frames and the laptop to receive them. For each of the cards we collected many samples while varying the transmit power of the VSG between -20 dBm and -95 dBm (lower than the receive sensitivity threshold of just about any commodity 802.11 radio) by 5 dBm increments. The resulting data is plotted in figure 3 along with a linear fit with a slope of 0.9602 and adjusted R-squared value of 0.9894 (indicating a strong fit to the data). The commodity radios perform remarkably well in terms of RSS measurement. To correct for the error they do exhibit, we use the slope and intercept of this fit to adjust our measurements.

After calibration, the residual error has nearly zero mean (-0.05 dBm) and a standard error of 1.7 dBm. The standard error of the sample mean varies as $SE_{\bar{x}} = \frac{s}{\sqrt{N}}$. This implies that any reasonable confidence level can be achieved by taking a practical number of samples. For example, 12 samples give a 95% confidence interval of ±1dBm, 45 samples gives ±0.5dBm, and so on.

2.2 Transmit Power Precision

Several studies have analyzed the fidelity of transmit power control in commodity wireless network interface cards (NICs) [9, 10]. Neither provides an exact calibration for our specific hardware, but they provide sufficient guidance for the type of experiments we have been performing. The devices studied offer a software API for setting transmit power, accepting settings in 1 dBm increments. These setting requests are implemented at a much coarser granularity by all of the hardware considered, including Atheros chipsets.[1] It is therefore not safe to assume that the requested power level matches the actual power level without first identifying the specific power levels supported by the hardware in use.

Because the phased array antenna provides additional – and relatively fine-grained – amplification, we are not particularly concerned with the absolute power level produced by the wireless NICs. Of more concern is the relative consistency. Shrivastava et. al. provide a conservative estimate: Their paper analyzes the combined variability of the transmitter, the channel, and the receiver. In the situation with the least expected exogenous variability (*LOS-light*), the apparent standard deviation of signal strength is less than 2 dBm. Additionally, their stationarity analysis shows a very low Allan deviation over both short (tens of packets) and long (thousands of packets) intervals [9].

This suggests that the sample sizes discussed for mitigating receiver measurement error are also reasonable for transmitter variability, and that samples separated by significant periods of time ought to be comparable.

2.3 MAC-Layer Flexibility

A challenge associated with using COTS wireless cards for research purposes is that the driver-card combination functions as a "black box." The exposed functionality is generally not sufficient for physical and MAC-layer experimentation.

One of the most basic requirements for a platform for experimental MAC design is the ability to send data frames exactly when and how the user wishes. There are several ways in which normal driver/hardware setups fall short:

- Not exposing information needed by experimental MAC protocols.
- Not offering a sufficient control interface for the physical parameters of interest.

[1] The Linux Wireless Extensions API allows device drivers to specify the set of supported power levels, but does not define the proper behavior for a device if an unsupported power level is requested. All of the hardware-driver combinations of which the authors are aware round to a supported level without returning an error code.

- Imposing unwanted aspects of an existing protocol (e.g. IEEE 802.11).

We addressed the first two with modest driver modifications. The chipset in the WiFi cards offers control over all the *IEEE 802.11 a*, *b*, and *g* PHY parameters on a per-frame basis, although channel changes cannot be made that quickly. The phased array antenna driver was originally coupled to the IEEE 802.11 protocol, but the two were fairly easy to separate. Harder than controlling *how* frames are sent is controlling *when*. Sections 2.5 and 2.6 discuss our approach to the timing problems in more detail.

There are several important aspects of the IEEE 802.11 protocol which tend to be implemented in hardware, making it challenging to use that hardware to explore significantly different protocols. In our WiFi chipset, these include MAC-layer retries and acknowledgements, carrier sense multiple access collision avoidance (CSMA/CA) back-off, and frame checksums. The rationale for implementing these functions in hardware is presumably speed: the turn-around time for raising an interrupt, sending information from an expansion card to the processor, waiting for the kernel to handle the interrupt and so on can be significant. One study found that doing acknowledgements in software took over 150 microseconds while the hardware implementation took less than 10 microseconds [11]. Such hardware-implemented features need to be either disabled or tolerated. Retries turn out to be easily disabled: There is a flag in the frame descriptor (`HAL_TXDESC_NOACK`) that causes the hardware not to wait for an acknowledgement after transmitting a frame. The frame checksums, and a few other mandatory header bits, we just accept. They are at worst overhead: The receiver can be configured to pass frames up the stack even if they are not addressed to that device or fail the hardware checksum test, so experimental protocols are not constrained to obey the semantics of those mandatory fields, only to fill them with values that the hardware will accept.

2.4 Implementing non-CSMA MACs

Suppressing CSMA/CA is critical for exploring non-contention-based MACs. In a few scenarios, such as a time division multiplexing (TDM) MAC with no outside noise sources, the medium should always be free whenever any node senses it and so CSMA/CA is harmless. In others, especially any system with intentional spatial reuse, multiple nodes may legitimately be active at the same time.

We developed a series of driver modifications to control CSMA/CA-related functions in the Atheros AR5212 chip set. Unlike retry-less transmission, which is already used for various broadcast frames in IEEE 802.11, CSMA-less operation is not an intended function of WiFi hardware. Consequently, this behavior has to be specified indirectly, and the necessary steps are not part of the documented public interface to the hardware[2]. Our group, with help from the broader Free Software community, reverse-engineered a procedure for practically disabling (and re-enabling) clear-channel assessment (CCA) in the cards we are using. Credit for analyzing closed-source driver behavior to identify registers touched during normal operation is due to the members of the `madwifi-devel` and `ath5k-devel` mailing lists.

Our patch to the MADWiFi driver changes three main parameters in the AR5212 chip. They seem somewhat redundant, but empirically the desired behavior is not always achieved without all three:

[2] As of 29 November 2008, Atheros Corporation has released the source code to their Hardware Abstraction Layer and announced that the free Linux drivers will be their public reference platform. This is likely to increase the publicly-available documentation significantly.

- *Diagnostic/Debugging Mode:* Set ignore bits for the Network Allocation Vector (NAV) in overheard packets, and physical carrier sensing.
- *Inter-Frame Spacing:* Configure the card to use the smallest possible durations for the gaps between frame transmission used in IEEE 802.11. If carrier-sensing is not being performed, these introduce pointless delay.
- *Disable Queue Backoff:* Prevent the card from backing off after draining a single hardware queue if there are other hardware queues with packets.

A patch adding this CCA control to MADWiFi is publicly available as part of our Commodity Atheros Research Platform (CARP) project[3].

2.4.1 Evaluation

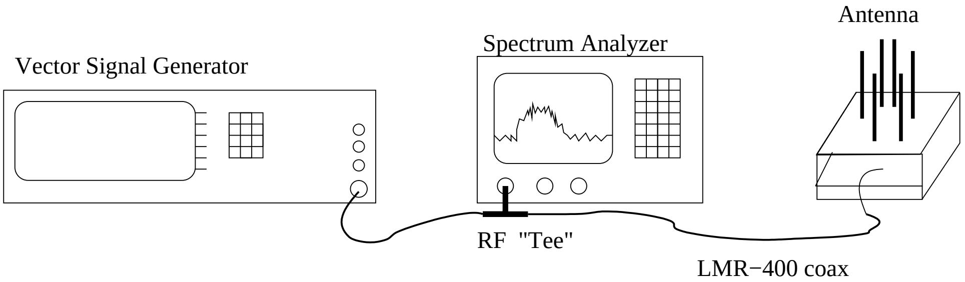


Figure 4: CSMA/CA Evaluation Apparatus

To verify that CCA has effectively been disabled, the WiFi card in the phased array antenna node is disconnected from the antenna and connected to the test equipment shown in Figure 4. The embedded computer is configured to produce a continuous stream of packets, and the vector signal generator (VSG) is used to create a competing signal on the same channel. The spectrum analyzer is used to determine whether the expected packet transmissions from the computer are occurring.

The testing procedure is shown in Algorithm 1. For each type of VSG signal, the experimenter verifies that packets are sent despite the interfering signal *only when CCA is disabled*. There is reason to believe that different mechanisms and thresholds are used for detecting different types of signals. In particular, IEEE standards define different power thresholds for deferring to signals recognized as valid PLCP headers and other “generic” signals. Further, a patent issued to Atheros describes their apparent approach to interference mitigation in more detail [12]. The mechanism employs a general power measurement component and specific detectors for OFDM and CCK modulations. Additionally, signal detections which correlate with successful packet reception are treated differently than those which do not. To address all of these cases, we tested with the following signal types:

- Sine wave (carrier only)

[3] Available at `https://systems.cs.colorado.edu/projects/carp/`. Based on personal correspondence, we know this is being used by researchers at IIT Delhi, the Dublin Institute of Technology, Communications Research Centre Canada, the University of Wisconsin, the University of Pittsburgh, WINLAB at Rutgers, and Stony Brook University.

Algorithm 1 CSMA/CA (CCA) testing procedure

```
for all VSG signal types do
    Configure vector signal generator
    Turn VSG RF output off
    for all $CCA$ in {on, off} do
        Set system CCA ← $CCA$
        Start computer sending packets
        for $power$ in RANGE($-100dBm \cdots +10dBm$) do
            VSG RF output power ← $power$
            Check for IEEE 802.11-like signal and VSG signal on spectrum analyzer
        end for
    end for
end for
```

- FM-modulated carrier
- Continuous (“unframed”) DSSS/CCK/DQPSK modulated carrier
- Continuous (“unframed”) OFDM/QAM-16 modulated carrier
- Framed complete packets: IEEE 802.11b 11 Mbps DSSS/CCK/DQPSK beacon frames
- Framed complete packets: IEEE 802.11g 54 Mbps OFDM/QAM-64 beacon frames

The last four were produced using Agilent Signal Studio and then replayed on the VSG.

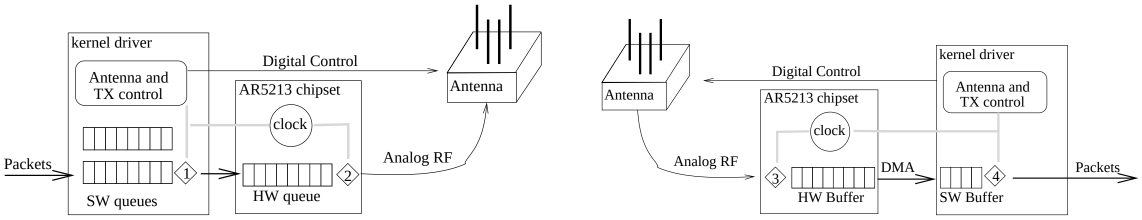
In all cases, the system performs as expected. With CCA suppression activated, the test computer produces a steady stream of packets regardless of the background signal from the VSG. Without CCA suppression, two different effects are seen: the valid packet streams cause the test computer to back off indefinitely and simple wave forms produce a more complex behavior. At low power levels, when the signal is initiated, the test computer stops sending packets for several seconds and then resumes. At high enough power levels, however, the test computer stops producing packets and does not resume. This behavior likely represents the “adaptive interference immunity control” described in the patent, whereby signal measurements which do not correlate with actual packet reception are identified as “false positives” and the threshold required to induce back off is adjusted. We did not identify the specific power thresholds or delay periods associated with this function.

2.5 Precise Timing Control

Precise timing is important for both efficient experimentation and a variety of MAC protocols. We are interested in both when packets are sent and when experimental antenna equipment changes state. We have developed infrastructure for quickly switching states in a coordinated manner across the entire system. There are two main challenges: (1) To interpret the results, it must be possible to match each packet sent or received to the antenna configuration in effect at the time. (2) To conduct experiments involving multiple nodes, it must be possible to synchronously change states so that the system state remains consistent.

We address both of these challenges by clocking our system off the high-resolution clock included in the adapter’s chipset. Most of the difficulty in connecting packets to antenna states comes from non-deterministic timing: On the sending side, the host can know when a packet is passed to the hardware (diamond 1 in Figure 5a), but it cannot know exactly when the packet will leave the antenna, especially if the card performs CCA and CSMA/CA backoff. Similarly, there is a variable delay between when the packet passes through the receiving antenna and when the host's interrupt handler is called to service the packet (diamond 4 in Figure 5b).

While there is a large margin of error associated with the *system time* when the packet was actually sent, the *MAC time* at reception can be known much more precisely. The MAC time, used for calculating retransmission timeouts and back-offs, is maintained by a high-resolution clock on the interface card. Packets are stamped by the hardware with the MAC time upon arrival (diamond 3), so there is almost no non-deterministic delay between the actual reception and the time-stamp. Since the AR5212 chipset also makes this time available, antenna transitions are scheduled relative to the MAC time.



(a) Transmission chain architecture

(b) Reception chain architecture

2.6 Time Synchronization

Using the on-chip timer helps with clock synchronization between nodes. MAC time synchronization is already required by the IEEE 802.11 protocol and is done in the interface hardware. In both BSS and IBSS (ad-hoc) modes, stations include their MAC time in beacon packets. Listening stations then set their own clocks off the beacons. Since this is done in the chipset (diamonds 2 and 3), the variability in delay is much lower – and thus the synchronization is much tighter – than what can be achieved using software on the end hosts.

3 Administration and Maintenance Infrastructure

The previous sections have discussed challenges related to using commodity equipment as a research platform. This section focuses on generic challenges likely to face any distributed wireless testbed.

Operational and maintenance issues become increasingly important as the number of nodes, their geographic distribution, physical inaccessibility, and heterogeneity of network connections all increase. The next several sections will describe the design decisions and support infrastructure developed to make the testbed as useful as possible. In 2004, our experimental procedure consisted of an operator with a laptop controlling each physical node, and human-layer signaling with cell phones or FRS radios. Experimental equipment was pre-configured in the laboratory before being transported to the test sites. Experiments were controlled and monitored by the operators, and results were downloaded onto the local laptops for later analysis. The subsequent testbed design has been driven by the need to address problems with that approach.

3.1 Centralization

The simple approach described above might be sufficient for small experiments if everything worked as intended. However, experimental hardware and software is almost inevitably flawed, and faults which escape notice during testing regularly cause problems during live experiments. When problems do occur, equipment needs to be rebooted, experiments need to be re-started, scripts need to be edited, and sometimes new software needs to be installed.

The (human) communication overhead of trying to identify and correct problems across all test locations quickly becomes prohibitive, even when the necessary fixes are small. We found that – even when nothing went wrong – coordinating a four-node experiment required at least a half-hour of overhead for setup, configuration checks, synchronization, starting the experiment, downloading the data afterwards, and running basic sanity checks on the data. Overall, the ratio of time expended to successful experiment time was very high.

Our primary requirement for the testbed infrastructure was that it enable centralized management. In particular, it is necessary at a minimum to be able to perform the following tasks, for all of the experimental nodes, from a single location:

- Configure, start, and stop experiments
- Gather and analyze data
- Replace experimental software
- Reboot crashed equipment

Additionally, it is not strictly necessary but very useful to be able to:

- Monitor the progress of experiments
- Actively identify crashed or mis-configured nodes
- Replace all system software

Our testbed infrastructure is designed to provide these capabilities. At its core, this infrastructure consists of a control plane network, a “management box” connected to each experimental antenna unit, and a collection of software tools. All of these will be described in detail in upcoming sections.

3.2 Management System

Every experimental antenna unit is directly connected to a management box, depicted in Figure 5. Each box contains a flexible single-board computer (SBC) along with hardware required for remote power control. These serve multiple purposes, the most basic of which is connecting the research equipment into the control plane network. The phased array antenna systems have built-in Ethernet, but the management boxes provide a number of critical services which are not possible without them.

Besides providing network connectivity, the management boxes also provide network booting to the antenna units. This approach greatly simplifies reconfiguration: Any software change, from one configuration file to a new operating system, can be made by uploading a new image to the management system and rebooting the experimental equipment. The equipment could boot from a remote network server, but only if the network to which they were attached had both the configuration and performance to support it, which would limit options substantially.

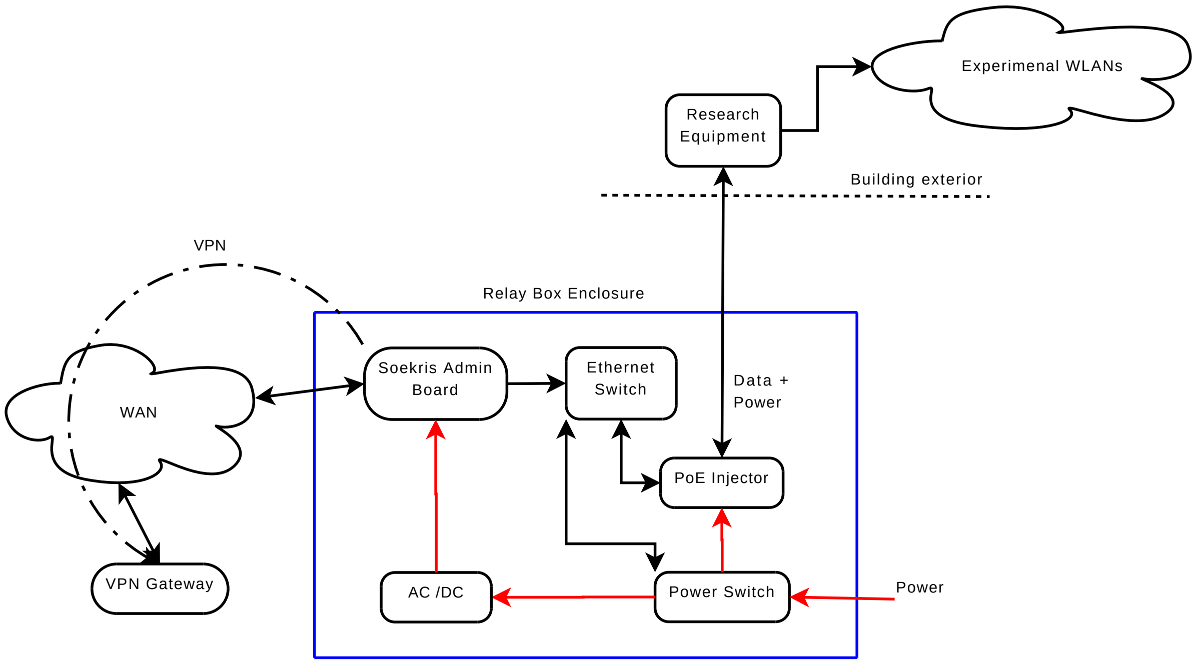


Figure 5: Management box configuration

3.3 Infrastructure Configuration

We designed the infrastructure with the goal of having as few “moving parts” as possible, because configuration errors are easy to introduce and can be very difficult to remedy once equipment has been deployed. To minimize opportunities for error, much of the system configuration is fixed, both between nodes and over time on any given node. We tried to identify the unavoidable sources of variability and isolate them so that as little of the overall system as possible has the potential to handle it incorrectly. The *unavoidable variability* comes from the address and configuration available on the outside (Internet) network link, the need to distinguish between units, new software images for the experimental equipment, and the passage of time.

The computer in each management box has one inward-facing network interface, a range of software processes, and one or more outward-facing interfaces. Except for time, which is rather pervasive, and the boot image, which is limited to one file served up by the TFTP daemon, the variability can be localized to the software directly interacting with the outward-facing network interface. The network configuration for the inward-facing interface and the devices on the internal network (the network-controlled power switch and the research equipment boot loader) is hard-coded and identical between units.

The organization of the control plane network relies heavily on the use of a virtual private network (VPN) and network address translation (NAT). On each management computer, the external IP address, DNS servers, and default routes are automatically configured by DHCP. Those are the only aspects that need to “know” anything about the network to which the box is attached. DNS is used to locate our VPN server, although the current IP address is also configured as a fall-back. Every management system is loaded with a different private key and X.509 certificate for connecting to the VPN, and this is the *only* hard state difference between boxes. The VPN daemon on the management board attempts to connect to the server on boot, or if it becomes disconnected for any reason.

3.4 Reliability and Availability

A key characteristic of a large testbed is that physical access to the equipment is likely to be difficult and time-consuming. In our testbed, the experimental equipment is mounted on rooftops and in several cases requires a ladder or safety equipment for access. At night or in inclement weather, on-location maintenance is effectively limited to swapping out the entire unit. The management boxes are indoors, but access is often difficult for administrative reasons, and is inconvenient under the best circumstances.

Availability is generally defined as $A = \frac{MTBF}{MTBF+MTTR}$, where $MTBF$ and $MTTR$ are mean time between failures and mean time to repair, respectively. Many of the design and configuration decisions described in sections 3.2 and 3.3 are intended to avoid failures, but the primary goal is to minimize the set of failures which require on-site physical intervention to repair, should they occur. A secondary goal is to make such intervention as quick and simple as possible.

3.5 Remote Repair

The most common significant failure in our testbed is a kernel hang in one of the phased array antenna units. A large portion of our experimental code has to run in kernel space, either for performance reasons or because it is an integral part of a device driver. The IXP425 platform includes a watchdog timer, and it is enabled, but some errors (especially acquiring locks and failing to release them) render the kernel effectively useless while still allowing the watchdog process to keep resetting the timer. Additionally, this platform has a limitation that the soft reset instruction resets the CPU but does not always reset the peripherals correctly, meaning that the device can reboot directly into a bad state.

We address this by including a network-controlled power switch in the management box. The experimental equipment and management computer are on separate switched circuits, and either can be turned off or power-cycled remotely using this switch. A limitation of this design is that the switch is only reachable if the computer is forwarding packets, so it cannot be used to address a hung management system.[4]

Another possible failure is corruption of the operating system on the experimental systems. This could easily result from either a kernel error, an intentional upgrade that proved to be faulty, an interrupted upgrade, or other circumstances. We considered several possibilities involving fail-safe operating system images and similar approaches, but always booting from the network sidesteps the entire issue: Nothing important is installed or stored on the experimental system except for the boot loader. As long as that remains intact, it is always possible to restore or replace the system software by simply rebooting.

3.6 Interchangeable Parts

On-site repairs, besides being time consuming, take place in less-than-ideal environments. It can be loud, windy, cold, hot, vibrating, high off the ground, or otherwise physically awkward.

[4] A previous version of the management box design used a power switch which was *itself* prone to hanging, a situation with little hope for remote repair.

The person making the repair has far fewer resources than would be available in the lab. Consequently, it is beneficial to make the repair process as simple as possible, and especially to avoid the need for on-site configuration and testing as part of the repair process.

This was a significant reason for the fixed-and-uniform configuration approach described in section 3.3. Every phase array antenna unit or network power switch has exactly the same hardware and configuration as every other one. Every management computer is the same as every other except for the contents of a removable compact flash card. This makes it easier to develop testing processes for each component, and means that a faulty or suspect component can be replaced with no thinking or configuration required. In fact, it is often easiest to replace the entire management unit as a whole – except for the flash card – and then diagnose the faulty one in the comfort of the lab.

3.7 Security

Since WART nodes are connected to untrusted networks, they are potentially susceptible to the same attacks that many other machines on the University of Colorado network experience on a day-to-day basis. Several steps have been taken to ensure that only authorized access is given to both the phased array antenna node and management board.

First, communication to the WART management nodes is restricted to nodes that are part of the same VPN. This requires having a certificate signed by the certificate authority, a process which is performed off-line. Once this trust has been established, we utilize SSH keys to allow remote logins directly to the phased array antenna nodes.

It is important to note that this last security stage is not without its weaknesses. This is due to the fact that the phased array antenna nodes run off a ramdisk and are thus without any real permanent storage. This forces each node to regenerate their SSH keys upon every reboot. This makes the nodes susceptible to man-in-the-middle attacks should an attacker obtain access to the VPN via a trusted certificate. One possible remedy to this challenge could be to embed the SSH keys directly into the OS image, which would allow anyone with an OS image to impersonate any antenna node, but would still be an improvement.

Another possible attack could stem from the wireless interface side. Should an attacker associate with a node, the node could potentially begin routing packets from unauthorized users. For now, we have disabled all routing services, but this remains a risk for future multi-hop experiments.

4 Deployment Logistics

Deploying a physically large testbed, especially with outdoor equipment, involves a number of challenges outside the traditional realm of computer science. There is a modest inherent engineering component that is significantly compounded by the need for approval and cooperation from various outside parties. All of the WART nodes are located on University of Colorado property, meaning that we only had to interact with a single owner, but it is a very large and bureaucratic one. We suspect that broadly similar issues would be likely to arise in working with another large organization, and possibly with multiple smaller ones.

In practice, deploying and operating equipment indoors in laboratory and office spaces has required only the informal approval of the research groups using that space. There may be relevant building codes or university policies, but there is no enforced approval process.

However, equipment installed on the outside of buildings, or visible from the outside, requires the involvement of the campus-wide organizations responsible for all construction projects. Fundamentally, there seems to be no administrative category for a project which spans a large area but with very minimal requirements. Building an outdoor testbed therefore becomes a university construction project with all of the overhead that entails.

Some of the more prominent logistical challenges encountered were:

- *Architectural Approval:* The aesthetic impact on campus buildings had to be approved by the campus architect.
- *Antenna Siting and RF Interference Approval:* A separate antenna committee had to be convinced that the proposed sites would not interfere with existing radio equipment.
- *Electrical Design and Installation:* The electrical requirements of the testbed equipment are extremely low; each node uses less power than a desk lamp. However, all construction projects involving new electrical connections are subject to the same approval process, regardless of the actual load. This means that an electrical design for each node had to be completed and signed off by a certified electrical engineer, and installation of the electrical components had to be performed by licensed electricians. Both had to be done by outside contractors hired through the office of facilities management, requiring an additional round of financial approvals before work could begin. Additionally, the waterproof plastic enclosures we had designed and fabricated for the management boxes had to be scrapped and replaced with metal enclosures specifically rated for containing electrical equipment.
- *Environmental Health and Safety:* All construction projects have to be audited for safety risks to both the workers and the campus in general. The primary concern was pre-existing asbestos building materials, although we also had to vouch for the microwave radiation levels.
- *Roof Integrity:* Because the equipment was to be mounted on the outside of buildings, both the attachment methods and cable connections had to be evaluated for waterproofing, fire sealing, and structural impact. In the cases where new holes had to be made through the roof, the penetration and waterproofing had to be installed by campus roofing services.
- *Antenna Structure:* Local building codes and campus design rules establish standards for wind, snow, and ice tolerance. The university requirements were the more stringent in this case, requiring that equipment be designed for 120 mph wind load. Very little antenna mounting equipment, especially in the WiFi market, meets those requirements. While commercial options do exist, we found it more cost-effective to design and construct our own in-house.
- *Financial Approvals:* After our research group and department decided to allocate money for the testbed, there were still a significant number of delays waiting for work orders and payments to be approved by other university entities. In particular, payments from the computer science department to facilities management, and from facilities management to outside contractors all required administrative approval before the payee could begin work.

4.1 Timeline

The testbed deployment process has required a total of two years. Most of that time has consisted of waiting for some necessary action by parties outside our department. Within that waiting, most of the time has been for administrative approvals, with actual design and construction requiring relatively little. Table 1 shows our actual timeline; with more foresight it probably could have been compressed.

The architectural and RF approval steps are an unavoidable bottleneck, as they determine whether and where equipment can be installed. In our case, it required approximately 9 months from the first informal proposals to a preliminary approval of the sites chosen. Once those decisions had been made, several of the remaining steps could probably have proceeded at once.

The obvious deployment tasks, namely physically installing the antenna node and management box, and running conduit and Ethernet cable between them, required on the order of one week per node.

Date	**Task**
12/2006	Initial talks with campus architect, campus network admin., and facilities management
01/2007	Initial proposal to campus architect Preliminary approval from campus network admin.
05/2007	Preliminary approval from campus architect
08/2007	Preliminary approval from facilities management
09/2007	Environmental health and safety approval
04/2008	Electrical plans completed Begin wired control plane install
05/2008	First WART node installation
06/2008	Electrical installation done
08/2008	Wired control plane done
11/2008	All WART nodes operational

Table 1: Deployment Timeline

4.2 Costs

Table 2 presents an approximate breakdown of the expense incurred *per node* in building this testbed. The dominant cost is not the research equipment itself but rather labor required for regulatory and school policy compliance. This includes both the electrical work mentioned earlier and the time spent by university employees on evaluation and project oversight.

5 Related Work

In this section we will give a high level overview of other wireless testbeds, both indoor and outdoor, and discuss how they compare to CU-WART.

Description	Cost
Phased Array Antenna Node	\$3,000
Management Box and other Control Plane Equipment	\$1,200
Installment Materials	\$300
External Labor and Fees	\$5,780

Table 2: Cost of labor and parts per WART node. The labor of research group members is not considered.

5.1 Outdoor wireless testbeds

The existing outdoor testbeds generally have more operational emphasis and less experimental control and management support than WART or the indoor testbeds. Most use stock IEEE 802.11 at the MAC and physical layers, although additional low-layer information is gathered to inform higher-layer research. This may in part reflect their designers' research interests, and may also reflect limitations resulting from the lack of a stable separate control network.

Roofnet: Roofnet is probably the first distributed testbed for IEEE 802.11 mesh networking [13, 14]. It consists of 20-40 nodes mounted on the rooftops of mostly residential buildings in Cambridge, MA. The entire network spans over an area of 1.5 x 1.5 kilometers. Unlike WART, Roofnet is unable to experiment using IEEE 802.11g modulation schemes, and is restricted to experiments involving omni-directional beam patterns. Roofnet is also a dual-purpose network; in addition to being a research testbed it also acts as a multi-hop backbone that provides Internet access. In contrast, WART is a dedicated experimental platform.

Rice/TFA Mesh: In terms of practical challenges, the RICE/TFA mesh is the most similar to our testbed. The physical size is similar: 2.12 km diameter for TFA, 2.36 km for WART. TFA has 14 nodes[5], WART has 7. The TFA-Rice mesh appears to involve equipment located on property with a variety of owners, suggesting similar access difficulties. There is little published information about the design and operation of the network, but it seems likely that their project and ours face similar issues. The deployment approach – in terms of choosing sites, not the logistics – is described in [15]. There are two primary differences: First, WART is focussed on experimental techniques and equipment at the physical layer, while the TFA mesh is not designed for experimentation at this layer. Second, the TFA mesh has a large operational component, while WART is purely experimental.

Mesh at Purdue (MAP): The MAP network is a primarily indoor research network which uses several fixed directional antennas for point-to-point links between adjacent buildings [16]. There are two approximately 20 meter links and two approximately 60 meter links.

RuralNet / Digital Gangetic Plains (DGP): The RuralNet deployment is an experiment in using IEEE 802.11 equipment for very long range point-to-point communication [17]. The operators use fixed directional antennas on traditional radio towers and buildings to form multi-kilometer links.

[5] Based on TFA public wiki as of 13 December, 2008.

Ad-hoc Protocol Evaluation testbed (APE): The basic design of the APE project is for humans to carry laptops that are pre-loaded with scripts to control the experiments. Node placement and mobility are controlled by "monkey walks" – human operators following directions displayed on the laptops. The APE software packages include modifications to their wireless network interface cards to collect signal strength information for all received packets [18, 19]. (This information is available as part of the Prism or Radiotap headers reported by many wireless NIC drivers).

5.2 Indoor wireless testbeds

There are a large number of indoor wireless testbeds, emphasizing a variety of technologies and design objectives. In general, the indoor testbeds are more compact (dense) than the outdoor testbeds. They also benefit from a much more controlled environment: the problems of remote repair and establishing and maintaining a reliable communication infrastructure, which have been at the forefront of our design challenges, are largely non-issues.

Many of the indoor testbeds have at least an order of magnitude more nodes than any of the outdoor ones: There are 400 nodes in the ORBIT testbed, over 400 (both wired and wireless) in Emulab, and 210 in Kansei [6, 20, 7, 21]. Much of the infrastructure developed for the indoor testbeds is oriented toward automating the process of configuring, controlling, and aggregating data from such a large collection of devices.

ORBIT: The ORBIT testbed consists of a "main grid" of 400 nodes arranged in a 20 by 20 grid, and several smaller "sandboxes" for development and testing [6]. The nodes consist of single-board computers with IEEE 802.11 NICs and omni-directional antennas. The experimenter can install arbitrary software on the nodes, but there are standard operating system images which include a specialized measurement and control framework [22, 23].

Emulab wireless extensions: Emulab is a well-known testbed for emulating arbitrary wired network topologies. It uses a variety of resource allocation and virtualization mechanisms to support many concurrent – but isolated – experiments [20]. Emulab has recently been extended to include several classes of nodes with wireless networking capabilities: Rack-mounted PCs with WiFi radios, PCs with GNU Software Radio and Ettus Research USRP hardware, Mica2 sensor motes, and mobile robots [7]. The non-mobile nodes operate very similarly to the wired-only Emulab nodes, with a dedicated Ethernet control plane, while the robots have significant mobility-specific support infrastructure. The mobile-node tracking and control infrastructure is conceptually similar to that described in [24]. Most Emulab nodes allow the user to install arbitrary code, down to the OS level. Because the mobile nodes do not have an out-of-band control and reprogramming mechanism, users are significantly constrained to avoid breaking the necessary on-board infrastructure. The mobile nodes do have attached Mica2 motes, over which the user has complete control.

UCR Testbed: The UCR testbed consists of single-board computer with stock IEEE 802.11 NICs spread throughout several floors of a single office building. The devices are powered via power-over-Ethernet from a set of PoE-enabled switches, providing a simple interface for power-cycling nodes [25]. Although not mentioned in the paper, the project web site indicates that they have added several PCs with USRP hardware to the testbed.

Hydra: Hydra is an indoor testbed for SDR experimentation. The physical layer is implemented with GNU software radio and USRP hardware, while higher layers are implemented with Click. The design work seems to be focussed on the prototyping platform, not the testbed aspect facility [26]. No information is given about the size or infrastructure of the testbed.

TRNC/ESPAR: TRNC/ESPAR is a hardware platform for evaluating directional MAC protocols using Electronically Steerable Parasitic Array Radiator (ESPAR) antennas [2, 3]. The authors refer to the system as a testbed, but it is in the sense of prototyping equipment, not a specific facility.

UCLA UnWiReD Laboratory: The UnWiReD testbed is a two-node facility for physical-layer experimentation with MIMO systems. The testbed is distinctive in that it provides a very flexible SDR platform for four-way MIMO at both the transmitter and receiver, and includes remotely-controlled mechanical actuators to adjust the antenna positions [27].

Miniaturized Network Testbed: MiNT: MiNT is an effort to simulate wireless networks with mobility using as little space as possible. It is conceptually very similar to the mobile nodes in Emulab, although developed independently [7]. Nodes have multiple wireless interfaces for various purposes; the ones used for the protocol under test are highly attenuated to simulate the loss of much larger areas. Additionally, the MiNT platform integrates with ns-2 to provide a hybrid simulation/emulation environment [28]. In the initial version, the mobile nodes were simple antenna platforms connected by RF cables to PCs where the actual processing took place. The MiNT-m paper describes improvements to dispense with the stationary PCs, along with additional management tools. The testbed infrastructure consists mainly of mechanisms for node tracking, positioning, control, state logging, and state rollback [24].

Kansei: Kansei is another testbed aimed at using high density to emulate a large system in a small area [21]. The system consists of 210 Stargate SBCs arranged in a grid with a wired control interface.

MeshTest: MeshTest uses standard PCs and an emulated RF environment. Each computer is connected into an RF matrix switch, allowing for programmable attenuation between nodes [29]. This provides significant flexibility in a very small physical size, although it entails some loss of fidelity. The infrastructure consists of the RF switch, the ORBIT software tools, and a custom-developed application for configuring the switch.

EWANT: Emulated Wireless Ad-hoc Network Testbed: EWANT uses standard PCs and a partially-emulated RF environment. Each PC is connected to one or more antennas through a combination of fixed attenuator and an RF multiplexer. The antennas are all positioned within small area, adding giving some measure of propagation realism [30].

6 Conclusion

This paper has presented WART, a testbed that will facilitate future networking research by providing unique physical layer capabilities not seen in any other outdoor networking testbed.

While the testbed covers an entire university campus, it is easy to manage and administer due to its wired control plane, which is remotely accessible from anywhere on the Internet.

The research motivation for building WART was to study the use of directional, steerable, and adaptive antennas. The prominent issues encountered in creating the testbed proved to be only indirectly related to that objective. The direct causes were *using commodity equipment*, *supporting low-level experimentation*, and *spanning a large geographical area*.

Commodity equipment: The research equipment (phased array antenna nodes) is comparatively affordable at $3000 per node, while specialized test and measurement equipment could easily cost an order of magnitude more. The consequences of using commodity hardware have been the need for significant calibration and testing, and extensive software hacking to make the hardware operate in unintended ways.

Low-level experimentation: Many of the experiments we wish to conduct are low-level, both in the sense of being at the physical and MAC layers of the OSI hierarchy, and in the sense of requiring “close to the metal” system implementation. This implies the need for easy reprogramming and crash recovery, high-volume data collection, and a flexible control interface. In practice, these in turn require a control connection that is separate from the experimental wireless system.

Large geographical area: It has been amply demonstrated that radio propagation in general, and directionality in particular, are very environmentally dependent [4]. Consequently, it was important that WART encompass a range of node densities and environmental features of interest. However, covering a large area implies *physical distance* and often *administrative diversity*, each of which contribute significant design challenges. Physical distance effectively precludes running dedicated cables from a central location to all of the nodes, which implies that power and network connectivity (if needed) must be supplied using resources available on site. It is this constraint which leads us to the “management box” design, with network support, power conversion, and power switching co-located with every measurement node. It is worth noting that a large testbed without the focus on low-level experiments may be able to dispense with the dedicated control plane and remote-reprogramming capabilities, significantly relaxing these requirements.

Covering a larger area often implies involving more administrative domains. Our sites are all owned by the same university, but building at a campus-wide scale requires the involvement of many departments – administrative and academic – and the approval of several levels of hierarchy. The practical impact of this cannot be overstated. The approval processes, and the cascade of design decisions made in order to secure those approvals, account for at least half of the total time and cost for this project.

This testbed was developed to study particular physical layer technologies, but the design lessons are not specific to that objective. Most of the challenges encountered in designing this testbed – and the solutions developed – are likely to apply to other outdoor and wide-area testbeds. We have developed an infrastructure for deploying nodes at widely separate, minimally provisioned sites and connecting them into an easily-managed unified research system.

References

[1] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, “Ad hoc networking with directional antennas: a complete system solution,” *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 496– 506, March 2005.

[2] H. Mitsuhashi, M. Watanabe, S. Obana, M. Bandai, and T. Watanabe, "A testbed with a practical smart antenna for directional mac protocols in ad hoc networks," in *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, (Washington, DC, USA), pp. 731–736, IEEE Computer Society, 2007.

[3] N. Kohmura, H. Mitsuhashi, M. Watanabe, M. Bandai, S. Obana, and T. Watanabe, "Unagi: a protocol testbed with practical smart antennas for ad hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 1, pp. 59–61, 2008.

[4] E. Anderson, C. Phillips, K. Bauer, D. Sicker, and D. Grunwald, "Modeling directionality in wireless networks [extended abstract]," in *ACM SigMetrics*, 2008.

[5] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. Sigcomm 2004*, ACM, August 2004.

[6] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1664–1669 Vol. 3, 2005.

[7] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile emulab: A robotic wireless and sensor network testbed," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, April 2006.

[8] E. W. Anderson, C. T. Phillips, D. Grunwald, and D. Sicker, "Modeling environmental effects on directionality in wireless networks," Tech. Rep. CU-CS-1044-08, Department of Computer Science, University of Colorado at Boulder, July 2008.

[9] V. Shrivastava, D. Agrawal, A. Mishra, S. Banerjee, and T. Nadeem, "Understanding the limitations of transmit power control for indoor wlans," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 351–364, ACM, 2007.

[10] F. Ben Abdesslem, L. Iannone, M. D. de Amorim, K. Kabassanov, and S. Fdida, "On the feasibility of power control in current ieee 802.11 devices," in *PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops*, (Washington, DC, USA), p. 468, IEEE Computer Society, 2006.

[11] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "Softmac - flexible wireless research platform," in *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[12] Atheros Communications, Inc., P. J. Husted, H. Ye, and A. Singla, "Adaptive interference immunity control." United States patent. World Intellectual Property Organization (WIPO) publication number WO/2005/048473.

[13] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 121–132, ACM, 2004.

[14] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05*, ACM, August 2005.

[15] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement driven deployment of a two-tier urban mesh access network," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, (New York, NY, USA), pp. 96–109, ACM, 2006.

[16] S. M. Das, H. Pucha, D. Koutsonikolas, Y. C. Hu, and D. Peroulis., "DMesh: Incorporating practical directional antennas in multi-channel wireless mesh networks," *Journal on Selected Areas in Communications*, vol. 24, November 2006.

[17] B. Raman and K. Chebrolu, "Experiences in using WiFi for rural internet in india," *IEEE Communications Magazine*, Jan 2007. Special Issue on New Directions In Networking Technologies In Emerging Economies.

[18] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom, and C. Tschudin, "A large-scale testbed for reproducible ad hoc protocol evaluations," *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1, pp. 412–418 vol.1, Mar 2002.

[19] E. Nordstrom, P. Gunningberg, and H. Lundgren, "A testbed and methodology for experimental evaluation of wireless mobile ad hoc networks," *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pp. 100–109, Feb. 2005.

[20] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, (Boston, MA), pp. 255–270, USENIX Association, Dec. 2002.

[21] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, "Kansei: A testbed for sensing at scale," in *IPSN/SPOTS*, 2006.

[22] D. Raychaudhuri, M. Ott, and I. Secker, "Orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities Tridentcom 2005*, pp. 308–309, 2005.

[23] M. Ott, I. Seskar, R. Siracusa, and M. Singh, "ORBIT testbed software architecture: Supporting experiments as a service," in *Proceedings of IEEE Tridentcom 2005*, (Trento, Italy), Feb 2005.

[24] P. De, A. Raniwala, R. Krishnan, K. Tatavarthi, J. Modi, N. A. Syed, S. Sharma, and T. cker Chiueh, "Mint-m: an autonomous mobile wireless experimentation platform," in

MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services, (New York, NY, USA), pp. 124–137, ACM, 2006.

[25] I. Broustis, J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, "A blueprint for a manageable and affordable wireless testbed: Design, pitfalls, and lessons learned," in *TridentCom*, 2007.

[26] K. Mandke, S.-H. Choi, G. Kim, R. Grant, R. Daniels, W. Kim, R. Heath, and S. Nettles, "Early results on hydra: A flexible mac/phy multihop testbed," *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pp. 1896–1900, April 2007.

[27] W. Zhu, D. Browne, and M. Fitz, "An open access wideband multiantenna wireless testbed with remote control capability," *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pp. 72–81, Feb. 2005.

[28] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Mint: a miniaturized network testbed for mobile wireless research," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, pp. 2731–2742 vol. 4, March 2005.

[29] B. D. Walker, I. D. Vo, M. Beecher, and M. Seligman, "A demonstration of the meshtest wireless testbed for delay-tolerant network research," in *Proc. CHANTS*, pp. 105 – 107, ACM, September 2008. ACM 978-1-60558-186-6/08/09.

[30] S. Sanghani, T. Brown, S. Bhandare, and S. Doshi, "Ewant: the emulated wireless ad hoc network testbed," *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1844–1849 vol.3, March 2003.