Design Considerations for a Multihop Wireless Network Testbed

Pradipta De, Ashish Raniwala, Srikant Sharma, and Tzi-cker Chiueh, Stony Brook University

ABSTRACT

Limited fidelity of simulators has prompted researchers to build wireless network testbeds for realistic testing. Unlike simulators, which have broad applicability, most of these testbeds are tailored to specific projects and cannot be used by a wider research community. Recognizing the growing importance of testbeds, this article is one of the first attempts to identify a comprehensive set of requirements for a general-purpose multihop wireless network testbed and the challenges therein. The issues range from initial testbed deployment and routine management to individual experimental configuration and data collection. We survey state-of-the-art wireless testbeds and highlight their salient features. The article is intended to provide an initial reference for researchers, application developers, and administrators dealing with various aspects of wireless network testbeds.

INTRODUCTION

Deriving accurate simulation models of wireless channel characteristics, such as radio propagation and channel errors, is inherently difficult and at best yields complicated models that are computationally inefficient. This, along with the gaining research impetus in cross-layer protocol design for wireless networks [1] that requires accurate characterization of the lower layers of the network stack, has prompted the development of a number of wireless network testbeds. Many of the wireless network testbeds built so far are designed for specific projects on which the researchers are working. Unlike simulation tools, which support a broad base of users, testbeds are often limited in their design and do not cater to the requirements of a wider research community. Hence, often the significant cost and management effort invested in building a testbed cannot be reused across multiple projects; nor can the testbed be easily reproduced. With a few exceptions, most published research on testbeds bears testimony to this observation [2, 3].

One of the well-known wireless network testbeds is the multihop ad hoc network testbed built at Carnegie Mellon University (CMU-DSR) for field evaluations of the Dynamic Source Routing (DSR) protocol implementation [2]. It comprised five mobile nodes and two static nodes. Mobility was introduced by placing laptops in cars. The testbed spanned an area of 700 m × 300 m. The large physical space and resource requirements make it difficult to replicate the CMU-DSR testbed for other projects. A testbed in a similar vein is the Ad Hoc Protocol Evaluation (APE) testbed at Uppsala University [3]. The testbed uses 37 nodes and incorporates mobility by strictly choreographing movement of volunteers carrying laptops. Roofnet [4] is another large-scale outdoor testbed used to study the characteristics of static mesh networks. Similarly, TAP [5] is an upcoming testbed designed for evaluating transit-access-point-based wireless backhaul network architecture. None of these testbeds is aimed at providing a common infrastructure for different wireless experiments. We believe that in order to make the use of testbeds more prevalent for wireless experiments, it is necessary to have shared remotely accessible testbeds, along the lines of PlanetLab [6] and Netbed [7] used for testing protocols for wired networks.

A recent extension to Netbed has built a mobile wireless testbed that can be shared by the research community [8]. Similarly, ORBIT [9] and WHYNET [10] projects are aimed at building shared wireless platforms for a broad range of mobile wireless networking research. The recent drive toward building these shared testbeds is a clear sign that wireless protocol development can benefit immensely from testing in real-world setting that a testbed provides.

Given this trend, it is worthwhile for the large community of researchers, application developers, and administrators dealing with various aspects of wireless networks, to understand the challenges underlying the design of a versatile wireless network testbed. It is our goal in this article to bring out the pros and cons of different testbeds that have been used over recent times, and in turn shed light on the choices one has in designing multi-hop wireless network experimentation testbeds. Although sensor networks form an active area of research in wireless domain, sensor network testbeds are beyond the scope of this article because sensor nodes (e.g. Berkeley Motes) are qualitatively different from a typical wireless testbed node in terms of resource constraints, like battery power, CPU speed and memory. A review of several existing testbeds helped us to derive a set of desirable features for wireless testbeds. We discuss the features of many of the existing testbeds and how each of them provides these features. This leads us to the design of our wireless testbed, called MiNT. MiNT is a miniaturized mobile multi-hop wireless network testbed that is designed to fit multiple nodes of the network on a tabletop. We use radio signal attenuation on the signal path to reduce the communication range of each node, thus setting up a wireless network testbed that can closely capture large-scale set-ups in a much smaller space.

The following is the outline of this article. We explore the requirements and challenges in realizing the necessary features in a wireless testbed. Then, we survey a large set of existing testbeds in light of these requirements. We also discuss our proposed miniaturized wireless network testbed (MiNT) architecture. Finally, we conclude the article.

Wireless Network Testbed Design Challenges

Cost

One of the foremost steps in setting up a wireless testbed is choosing appropriate hardware that has favorable cost and performance trade-off. Some of the key components required to build a multihop wireless testbed are:

- The basic hardware platform, which could be a laptop, a desktop PC, any other small form-factor PC, or an embedded device such as a Linksys WRT54GS wireless router that runs a standard OS like Linux
- Wireless interface cards that give the right degree of control in terms of adjusting different configurable parameters
- External antennas (omnidirectional or directional)
- Other accessories, like RF signal attenuators, RF cables, steering devices for directional antennas, and battery packs
- A platform for introducing mobility, which could be anything from cars to paid volunteers to mobile robots capable of carrying payloads

Any of these components can be either custom-made or commercial off-the-shelf products. Per node cost against flexibility of operation is usually the guiding factor in this decision. For example, commercial wireless cards, although inexpensive, often provide limited configurability: it may not be possible to finely configure transmit power or reception sensitivity. It is therefore a challenging task to choose the right set of hardware for putting together a wireless testbed node.

Management

The management of a testbed begins with the initial setup comprising *hardware/software configuration* and *deployment* of the nodes, and continues through the entire lifetime of the testbed in terms of *monitoring* the status of each node and *maintaining* them. Unlike a wired network testbed, the setup of a wireless testbed involves finding suitable locations for placing each node so that a true multihop network can be created. Thereafter, managing a testbed, which is usually spread over a large geographical area, requires a remote monitoring tool that provides constant feedback on various node parameters and changing wireless link conditions.

A remote monitoring tool in turn requires:

- A channel for communicating control information that does not interfere with experiment channel
- A set of node/link/channel parameters to be monitored to provide a comprehensive view of the testbed

The monitored parameters are fed to a monitoring agent on a central controller, thus supporting remote administration of the testbed. 24 × 7 operation of a mobile testbed additionally requires an automatic battery recharging facility for different nodes.

Resource Sharing

Building testbeds that could be shared by multiple users is receiving significant attention in the network research community. The wired network research has been spurred by the development of shared testbeds such as Netbed [7] and PlanetLab [6]. Resource sharing in *wired* network testbeds involves allocating a subset of nodes from a pool to individual experiments (Netbed), or multiplexing several experiments on each node (PlanetLab). It is not possible to borrow these paradigms unmodified for sharing a wireless testbed among multiple users. Since a wireless channel is a shared resource, it must be ensured that multiple experiments conducted on a testbed are isolated in the spatial, frequency, or temporal domain.

Experimental Control

Defining an experiment on any testbed involves several steps. Some of these steps, such as application and node configuration, are similar to an experiment setup in a wired testbed, while others, such as topology configuration and mobility configuration, are more specific to wireless experiment setup.

Topology configuration: Configuring a wireless network's topology involves placing the nodes in such a way that each node-pair satisfies some specified link property, such as signal-to-noise ratio (SNR) or link error rate. A manual topology configuration, where the user determines the correct location of all nodes to satisfy the specified link properties, becomes tedious as the number of experiment nodes increases. Ideally, the user should just declaratively specify the topology constraints, while the locations of individual nodes are automatically computed based on a priori measurements calibrating the testbed.

Application configuration: This step involves setting up the traffic generators and traffic sinks, and can be done in two ways. The user can write her own applications. Alternatively, a library of applications could be provided from which the user chooses the appropriate applications. This could be useful in reducing setup overhead of different experiments.

Defining an experiment on any testbed involves several steps. Some of these steps, such as application and node configuration, are similar to an experiment setup in a wired testbed, while others, such as topology configuration and mobility configuration, are more specific to wireless experiment setup.

Debugging any wireless protocol on a testbed involves solving all the difficulties of distributed debugging. Hence, a facility for introducing controlled faults, to uncover bugs in protocol implementations, could provide an added advantage to wireless testbed users.

Mobility configuration: In wireless network experimentation, a user should be able to configure node mobility by specifying:
- Trajectories
- Target locations
- Mobility models

In a simulation environment, models such as Random Waypoint and Brownian are used to specify mobility patterns. Similar models can be used to describe mobility of testbed nodes. Tools like CAD-HOC [11] can also be used to generate more realistic mobility patterns.

An associated problem with node mobility in a testbed is to plan the motion of the nodes such that collision among nodes can be avoided. Also, as the nodes move around, their locations must be updated on the management GUI. A tracking system is required for updating node location in real time.

Setting node/card properties: The user may require privileged access to nodes to allow changing node/card configurations, as well as to install kernel modifications implementing the protocol under test. Providing privileged access to users makes it necessary to be able to restore vanilla conditions after each experiment. It is also important to save a user's experiment configurations, such as node parameters and locations, and allow reloading the configurations to make rerunning of an experiment faster.

Experiment execution: The next step in experiment control is providing the user with ways to fine-tune an experiment by observing the results during execution. In addition to simultaneous start/stop of experiments on all the testbed nodes, an ability to pause experiments, modify parameters on the fly, and then continue experiments could substantially reduce experimentation time.

Protocol debugging: Lastly, debugging any wireless protocol on a testbed involves solving all the difficulties of distributed debugging. Hence, a facility for introducing controlled faults to uncover bugs in protocol implementations could provide an added advantage to wireless testbed users.

Experiment Analysis

A commonly used method of analyzing an experiment is to collect packet traces from various nodes, aggregate them, and provide the user with visualization and filtering tools to analyze the aggregated trace. The basic mechanism remains the same across wired and wireless testbed experimentations; the actual statistics that need to be collected on a per-packet basis differ.

Trace collection: The traces are collected to extract various statistics such as throughput and loss rate, as well as provide necessary information for diagnosing anomalous conditions in the protocol. The standard technique for capturing packets is to use network sniffing tools, such as *tcpdump* or *ethereal*. By default, these tools only capture the packet dynamics seen at the driver level. One can additionally switch a card to *monitor* mode, and capture all *link-level* transmissions including link-layer protocol headers and control frames. The protocol headers often provide additional useful information, such as received signal strength, noise level, and medium access control (MAC) timings.

In a distributed environment, multiple monitor nodes are needed to completely cover the entire transmission domain of the nodes. There are two ways to set up the monitoring facility of a wireless testbed. The first approach is to let the experiment nodes themselves perform the monitor functionalities and sniff the packets in their respective neighborhoods. Here, each experiment node must be equipped with *at least two* wireless cards, assuming one card is being used for the experiments. With this approach, one can reconstruct each experiment node's view of the wireless channel in terms of received and transmitted packets. The second approach is to keep the monitor nodes and experiment nodes separate. Although this increases the number of testbed nodes, the overall hardware requirement is reduced for testbeds with large number of experiment nodes. Here, the placement of monitor nodes must be carefully planned to ensure complete coverage of the transmission space. Designing the monitoring facility is an important step in architecting a usable wireless testbed.

Trace aggregation: Once the traces are collected, the next step is to aggregate them. Traces from all nodes are shipped to a central node and merged based on timestamps. This entails all monitor nodes being time synchronized at the start of the experiment. The same packet can appear in traces collected by multiple monitor nodes. During aggregation, these copies must be correlated to get a unified trace of the transmitted packets.

Trace visualization: Another component of experiment analysis is visualization of the collected trace. Visualization must show the transition of packets originating from each node with respect to time. Visualization may be done offline or in real time as the experiment progresses. For real-time monitoring, the collected traces must be transported to the controller node that must perform the parse, collate, and display operations in real time.

Data filtering: A final element of efficient experiment analysis is the use of filters to reduce the amount of trace collected on each node. With user-defined filters, it could be possible to collect only the packets pertinent to the analysis. This is analogous to collection of application, routing, or MAC layer traces in an *ns-2* simulator. In real-time visualization of traces, filtering could be very useful in reducing the amount of data that needs to be transported to and processed at the controller. A generic event search mechanism can further simplify the analysis by allowing search of specific conditions such as excessive link layer retransmissions.

Applicability

While designing a shared testbed, the goal is to incorporate the facility for conducting as diverse a set of experiments as possible. Apart from live testing of real protocol implementations, one could also enable *hybrid simulation*. In a hybrid simulation, some of the layers of the protocol stack in a simulator are replaced with real hardware. For example, the wireless MAC and physical layer in the simulator could be replaced by real hardware, while the simulator's routing, transport, and application layers are kept intact. This enables reuse of simulation modules and scripts, avoiding the need to reimplement the protocol for live testing.

In order to accommodate testing of diverse protocols, one should be able to modify individual layers of the protocol stack. Usually the MAC layer implementation is part of the wireless card firmware, and no interfaces are exposed to modify it. For testing new MAC protocols the testbed could be designed using software defined radio (SDR). In an SDR, only the wideband digital-to-analog conversion (DAC) and analog-to-digital conversion (ADC) are done in hardware; the rest of the signal generation, signal processing, and access control are performed in software on a general-purpose processor. This enables one to work with different implementations of the MAC layer.

Finally, it is also useful to support a mix of different physical layer technologies, such as General Packet Radio Service (GPRS), 3G, ultra-wideband (UWB), and 802.11, in the testbed.

Repeatability

Accurate repeatability on a wireless network testbed is hard to achieve because external factors like fading, attenuation, and presence of other interfering sources are always changing. This is unlike wired testbeds, where most nonrepeatability comes from sequencing of actual events, while the external factors stay constant. To achieve repeatability without sacrificing realism, it is possible to create controlled environments for a testbed by placing it in anechoic chambers and introducing interference in a regulated manner, or using RF cable shielding to prevent external interference. There is always a trade-off between testbed realism and repeatability. An additional advantage of a testbed supporting repeatability is that it could be used as a reference platform for various experiments.

A Survey of Wireless Testbeds

A number of researchers worldwide have built wireless testbeds for experimentation. These include some full-scale wireless LAN deployments that are additionally used for experimentation purposes. Some of the other testbeds attempt to reduce the physical space required for setting up the entire testbed without sacrificing the essential characteristics of wireless channels. Based on the available literature, we now present a critical view of these testbeds. We group them into two categories: full-scale testbeds and miniaturized testbeds. We highlight the key strengths of these testbeds, show the techniques used to address core testbed design issues, and also point out some of the limitations.

Full-Scale Testbeds

We first study some of the full-scale testbeds that are tailored to satisfy specific projects' requirements, and then look at three prominent upcoming shared testbeds that will be available to the entire research community.

CMU Testbed for Evaluating DSR (CMU-DSR) — The CMU-DSR testbed [2] is built to test the implementation of the DSR protocol for ad hoc wireless networks. The testbed is designed with five mobile nodes and two static nodes spread over an area of 700 m × 300 m. The mobile nodes are implemented with rented cars carrying laptops acting as mobile ad hoc nodes. For management of nodes spread over the wide area and real-time monitoring, the testbed implements a visualization daemon on the node at the field office. Each testbed node runs:

- A GPS daemon that collects node location information
- A position and communication tracking daemon (PCTd) that unicasts packet traces to the field office
- tcpdump for collecting packet traces

Finally, signal quality and signal strength values are collected on a per packet basis as reported by the WaveLAN-I cards. No separate control channel is used for transmitting the traces back to the field office. The CMU-DSR testbed never tried to resolve the questions related to resource sharing among multiple users or wide applicability. The purpose was to stress test a specific application and study its behavior. It took around seven months to build the complete testbed infrastructure, which points to the difficulties of setting up a similar full-scale wireless testbed.

Ad Hoc Protocol Evaluation Testbed — The APE testbed [3] is used for comparative study of different ad hoc routing protocols. Similar to any other multihop wireless testbed, APE also uses a large space in locating the nodes. Node placement is done manually, although the topology generation for experiments is aided by APE-view, a log driven animation tool showing node positions and connectivity. Mobility in APE is introduced by explicitly choreographing movement of volunteers carrying laptops. In APE, trace aggregation and visualization of the monitored data is done offline. During an experiment the packet traces are timestamped and logged into a file. Finally, the logs are collected by a collect node, and the merged trace is replayed using APE-view. The APE testbed is also not designed for shared usage, and does not focus on supporting diverse experiment scenarios.

MIT Roofnet — In order to facilitate mesh networking research, the Roofnet project at Massachusetts Institute of Technology (MIT) built a 50-node testbed [4] spread across volunteers' rooftops in Cambridge, Massachusetts. Each node is built using a desktop PC equipped with a PCI card, and external antennas are mounted on the chimneys of volunteers' houses. An attractive feature of Roofnet is its use of commodity off-the-shelf hardware, that reduces the per-node cost. The MIT Roofnet testbed is currently deployed and provides broadband Internet access to users.

To achieve repeatability without sacrificing realism, it is possible to create controlled environments for the testbed by placing it in anechoic chambers and introducing interference in a regulated manner, or using RF cable shielding to prevent external interference.

We judged some testbeds, such as Netbed, WHYNET, and ORBIT, to be high in cost, but the expenditure is justified keeping in mind the wide usage these testbeds are aiming at. Testbeds such as Roofnet keep their cost low by use of off-the-shelf hardware.

Testbed	Cost	Mgmt	Share	Expt control	Expt analysis	Repeatability	Applications
CMU-DSR [2]	X	X	X	√	√	X	X
APE [3]	X	X	X	√	√	X	X
ORBIT [9]	X	X	√	√	√	X	√
WHYNET [10]	X	X	√	√	√	X	√
Netbed [8]	√	X	√	√	√	X	X
Roofnet [4]	√	X	X	X	√	X	X
TAP [5]	X	X	X	X	√	X	X
Sarnoff-tbd [12]	√	√	X	X	—	√	X
EWANT [13]	√	√	X	X	—	X	X
CMU-emu [14]	X	√	X	X	—	√	X
MiNT [15]	√	√	√	√	√	X	√

Table 1. *Comparison of various wireless network testbeds in terms of desirable features. An X denotes that the testbed has inadequate support for the feature, and a √ denotes that it addresses the feature. In all other cases the presence/absence of the feature is not known.*

Since the Roofnet nodes are static, there is no flexibility of creating new topologies or introducing mobility. However, the topology may be altered by changing transmit power, thereby breaking/introducing links. Roofnet also features real-time visualization of various link characteristics such as SNR and error rate. The statistics are measured by transmitting broadcast packets, so there is no link-layer acknowledgment (ACK) traffic. Roofnet is mainly used for studying wireless network characteristics rather than for experimenting with diverse application scenarios.

Transit Access Points — This testbed [5] is being designed at Rice University to evaluate TAP, a high-speed wireless backbone architecture. The core building block in the testbed, a transit access point, is a node equipped with multiple radios and antennas that can be used in unison to form very-high-data-rate links with other TAPs. The lack of commercially available hardware with the specific requirements prompted custom design of TAP hardware and bumped up the testbed development cost. As the testbed is designed for specific types of experiments tuned to the project goals, it is not suitable for use across a wide variety of experiments.

Currently the TAP-based nodes have been used to set up a low-cost community network for broadband Internet access in Houston's Pecan Park neighborhood in the East End. This is a pilot network of 12 nodes that form a multihop wireless backbone.

Shared Testbeds — Keeping in tune with the growing needs of the wireless research community for testbeds, the following are some of the large-scale open platforms being designed for shared usage.

Netbed Wireless Extension — This wireless testbed is an extension to Netbed [8], which is a shared wired network emulation platform at the University of Utah. The design of the testbed proposes placing a dense mesh of wireless nodes across the department building. By selectively turning the nodes on/off, it is possible to generate many different topologies. Netbed uses ns-2-like scripts for configuration and execution of experiments.

Additionally, a separate testbed based on mobile robots has been set up for public use. It currently uses five Berkeley motes and five Stargate nodes mounted on Acroname Garcia robots. The robots are remotely controlled and are set up indoors in an office space.

ORBIT —Open Access Research Testbed for Next-Generation Wireless Networks [9] is under development at Rutgers University. Currently ORBIT comprises an indoor radio grid of 400 nodes in a 20 m × 20 m space, but is planned to be extended into an outdoor field trial network consisting of both high-speed cellular and 802.11 wireless access. A testbed node in ORBIT is custom-designed, and supports remote management capability through wired connectivity. A user can remotely login and execute experiments. However, since the radio grid is static, topology generation in ORBIT depends on selectively switching some nodes on/off, as in Netbed. However, configuring every link to a desired SNR value is a problem yet to be solved. The mobility of a node is simulated through a separate mobility server, that transfers the state of a mobile node from

one node in the grid to another. ORBIT nodes also provide privileged access to a node for performing kernel-related operations. This could taint the kernel running on a node. Hence, disk reimaging at the end of an experiment run is required. Some of the salient features of ORBIT include:
- A measurement library that minimizes coding effort of a user by providing predefined functions
- A well designed database storage mechanism of all the results from an experiment that can be accessed through standard SQL queries

Although in its beta stage, ORBIT can become a useful platform for wireless experimentation.

WHYNET — This project [10] also aims at building a scalable shared testbed for next-generation mobile wireless networking technologies. This testbed is designed to be a networked federation of geographically distributed, heterogeneous wireless physical testbeds with multiple protocol stacks (CDMA2000 cellular and IP), next-generation physical technologies including UWB, multiple-input multiple-output (MIMO), and SDR, and a parallel and distributed multitool simulation framework. The WHYNET testbed also incorporates hybrid simulation capability that allows integration of simulated, physical (real), and emulated components into a single evaluation platform for scalable and realistic wireless network experimentation. This can provide the experimenter with the best trade-off in terms of realism, scalability, and repeatability depending on the target network scenario. The scope of the WHYNET project is more than just remote testbed access, as in ORBIT. It proposes to provide the research community with validated models of emerging protocols and radio technologies, measurement data sets, software, and tools to deploy a similar setup locally.

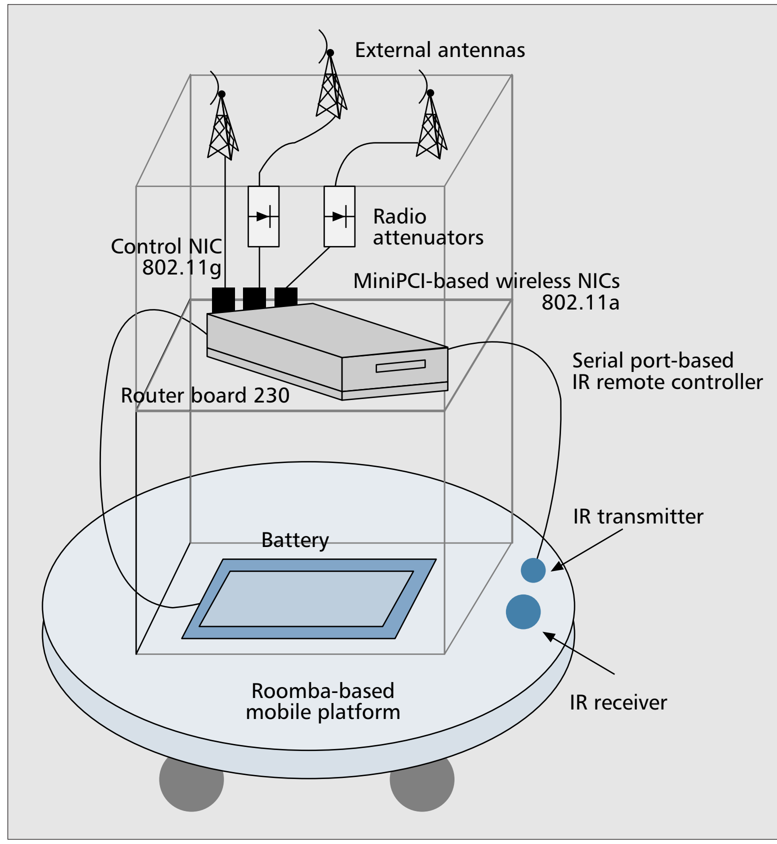

Figure 1. *Every node is equipped with at least one wireless PCI card. This PCI card does not have any internal antenna. It is connected to an external antenna through fixed radio signal attenuators. The antenna is (are) mounted on a mobile robot, which introduces node mobility.*

Miniaturized Testbeds

Although large testbeds more closely match the characteristics of full-scale deployments, such testbeds often fall short of manageability and flexibility of experimentation. In this subsection, we look at some of the testbeds that overcome the large space requirement through miniaturization.

Sarnoff Testbed (Sarnoff-tbd) — Kaba and Raichle at Sarnoff designed a *testbed on a desktop* by restricting and controlling radio ranges and propagation effects of PC cards [12]. Through this testbed the authors demonstrate the idea of using fixed radio signal attenuators to reduce the radio range of wireless cards. In order to control the external effects on signal propagation and make the experiments repeatable, the testbed uses coaxial cables that can pass the radio signals from one card to another unaffected by external interference. The PC cards are shielded with custom-made copper shields to prevent power leakage from the internal antennas.

EWANT — Sanghani *et al.* built the Emulated Wireless Ad Hoc Network Testbed (EWANT) [13] with the goal of providing a low-cost environment for wireless research. Similar to the Sarnoff testbed, they also utilize attenuators and shielding to shrink the radio ranges. The mobility of the nodes is emulated by connecting one PC card to four external antennas through a 1:4 RF demultiplexer, and switching the transmission through these antennas.

Physical Emulation Platform — In the Physical Emulation Platform (CMU-emu) for wireless experiments, Judd and Steenkiste use *digital emulation* of signal propagation using a field programmable gate array (FPGA) [14]. The aim is to make the experiments repeatable, while preserving the realism of the MAC and physical layers. To achieve this, they use coaxial cables to feed the signal from an RF device to the emulator. The emulator controls the emulation of signal propagation by taking into account the impact of external factors, like multipath interference, through use of signal propagation models. The main drawback of this approach is that external factors are still modeled and are not truly real. Additionally, use of FPGAs make the platform more expensive relative to others.

Discussion

So far we have highlighted the salient features of most of the current wireless network testbeds. In Table 1 we present a characterization of these testbeds with respect to how efficient each is in terms of addressing various design challenges discussed earlier. The characterization is based on the available literature of the testbeds. A large physical space requirement makes management of a testbed difficult. This is true for most of the full-scale testbeds, like CMU-DSR, APE, Roofnet, and TAP. We judged some testbeds, like Netbed, WHYNET, and ORBIT, to be high in cost, but the expenditure is justified keeping in mind the wide usage at which these testbeds are aimed. Testbeds such as Roofnet keep their cost low by using off-the-shelf hardware. Although technically feasible in all testbeds, only Netbed, WHYNET, and ORBIT support sharing. In evaluating a testbed for experimental control, we emphasize the flexibility of new topology generation and introduction of mobility. Testbeds like Roofnet based on fixed nodes are less flexible than others on this count. A shared testbed must be rich in its applicability to diverse scenarios, like live experimentation and hybrid simulations. WHYNET and ORBIT propose to address a more diverse set of scenarios than others.

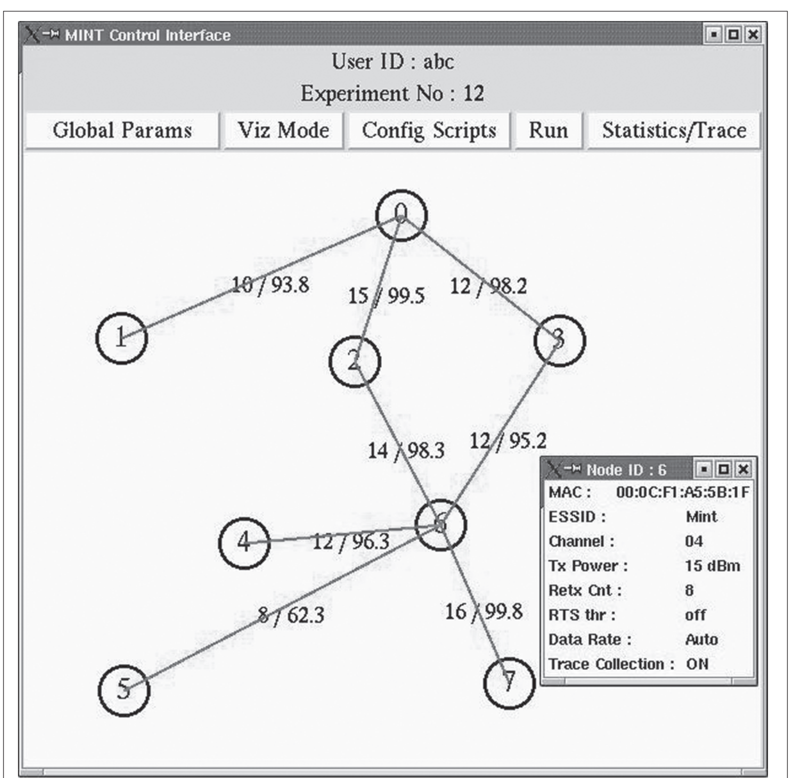

Figure 2. *GUI for controlling MiNT. The position of the core nodes in the testbed is changed by dragging their associated icons in the GUI. Signal strength and delivery rates among nodes are continuously displayed during topology setup. In this figure, signal strength and delivery rates relative to nodes 0 and 6 are displayed. Default node parameters can be set using the Global Params button, and can be overwritten on a per node basis by clicking on the associated node icon. Traffic scripts, mobility scripts, and fault injection scripts can be loaded through the Config Scripts button. Merged traces and network statistics can be viewed through the Statistics/Trace button. Finally, double-clicking on a node opens a console window to it that can be used to install protocol software/modules.*

The Proposed Miniaturized Wireless Network Testbed

In this section we present our solutions to various design challenges discussed earlier. Many of the issues related to manageability and reconfigurability of a multihop wireless testbed stem from the large area such a testbed spans. Miniaturizing, or scaling down, the testbed can therefore be an answer to many of the problems. Based on this lesson, we architected the Miniaturized Wireless Network Testbed (MiNT), which is a *reconfigurable* miniaturized mobile wireless network testbed [15]. The key to miniaturizing a multi-hop te stbed is to shrink the communication distance of individual nodes. In MiNT, we connect the network cards on each node to fixed radio signal attenuators that in turn are connected to low-gain external antennas, as shown in Fig. 1. This setup reduces each node's communication range to about 2 ft. Use of about 60 dBm of attenuation along the signal path from transmitter to receiver allows MiNT to place eight nodes in a space of 12 ft × 6 ft, and still create a truly multihop network. Miniaturization further reduces the testbed's interference with the production networks. To minimize the overall testbed cost, MiNT uses commercial off-the-shelf hardware, such as Routerboard 230 with modified Pebble Linux as the base platform, Atheros chipset-based 802.11a/b/g mini-PCI cards, with madwifi driver in ad hoc mode for communications, and iRobot's Roomba as the mobility platform. An important feature of MiNT is its ability to run 24 × 7 with minimal human intervention. The key problem of powering the wireless nodes and Roomba is solved by using batteries. For charging the batteries automatically, we use the auto-docking feature of the Roomba, which triggers a Roomba to dock when it is low on battery power.

MiNT provides a flexible experimentation environment through a comprehensive set of control mechanisms and result analysis tools [15]. Each node is placed on a mobile robot that can be remotely controlled. This enables mobility-related experiments as well as fast remote topology reconfiguration through a management GUI (Fig. 2). The displacement of each node in the GUI is translated into robot movement in the physical testbed. The interface also allows an experimenter to configure various other node parameters, such as transmit power, retry limit, that are necessary for efficient experiment control. One of the wireless cards on each node is configured in RF monitoring mode for collecting packet trace during experiments. The traces are transferred to a control node and can be visualized in real time. Standard debugging techniques are insufficient for a distributed environment as in MiNT. To alleviate this problem, MiNT features a fault injection and analysis tool. The tool is used to trigger network faults, such as packet drops, based on user-specified conditions.

In addition to supporting live multihop wireless experiments, the testbed is also designed to support hybrid ns-2 simulations where unmodified ns-2 scripts can be run with MAC and physical layers of ns-2 replaced with real hardware

[15]. Event packets are encapsulated in User Datagram Protocol (UDP) payloads, padded to the actual packet size and transmitted over a real wireless channel. Moreover, ns-2 implementation is modified to use the system clock on each node (synchronized using NTP) instead of the global virtual clock.

CONCLUSIONS

To overcome the limitations of simulators, as well as have an appropriate platform for new research areas such as cross-layer protocol design, there is a growing interest in using wireless testbeds. There have been several efforts in designing wireless testbeds, but often these efforts are geared toward specific projects. This article pinpoints the feature set necessary for a general-purpose wireless network testbed, related to controllability and analysis of wireless experiments. We looked into the challenges associated with each of these design features. We have presented the design of some of the well-known state-of-the-art testbeds. We have also presented our solutions to the design challenges leading to MiNT, which is a flexible miniaturized wireless network experimentation testbed. This article provides a useful starting point for researchers interested in designing wireless network testbeds. It is equally helpful to those looking for appropriate platforms for wireless experimentation.

REFERENCES

[1] M. Conti *et al.*, "Cross-Layering in Mobile Ad Hoc Network Design," *IEEE Comp.*, Feb 2004.

[2] D. Maltz, J. Broch, and D. Johnson, "Experiences Designing and Building a Multi-Hop Wireless Ad-Hoc Network Testbed," *CMU TR99-116*, 1999.

[3] H. Lunndgren *et al.*, "A Large-Scale Testbed for Reproducible Ad Hoc Protocol Evaluations," *Proc. WCNC*, 2002.

[4] B. A. Chambers, "The Grid Roofnet: A Rooftop Ad Hoc Wireless Network," MIT Master's thesis, June 2002.

[5] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling Large-Scale Wireless Broadband: The Case for TAPs," *Proc. Hotnets*, 2003.

[6] L. Peterson *et al.*, "A Blueprint for Introducing Disruptive Technology into the Internet," *Proc. Hotnets*, 2002.

[7] B. White *et al.*, "An Integrated Experimental Environment for Distributed Systems and Networks," *Proc. OSDI*, Dec. 2002.

[8] B. White, J. Lepreau, and S. Guruprasad, "Lowering the Barrier to Wireless and Mobile Experimentation," *Proc. Hotnets*, 2002.

[9] D. Raychaudhuri *et al.*, "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols," *Proc WCNC*, 2005.

[10] WHYNET, http://pcl.cs.ucla.edu/projects/whynet/

[11] S. Shah, E. Hernandez, and A. Helal, "CAD-HOC: A CAD Like Tool for Generating Mobility Benchmarks in Ad-Hoc Networks," *Proc. 2nd SAINT*, 2002.

[12] J. Kaba and D. Raichle, "Testbed on a Desktop: Strategies and Techniques to Support Multihop MANET Routing Protocol Development," *Proc. MobiHoc*, 2001.

[13] S. Sanghani *et al.*, "EWANT: The Emulated Wireless Ad Hoc Network Testbed," *Proc. WCNC*, 2003.

[14] G. Judd and P. Steenkiste, "Repeatable and Realistic Wireless Experimentation through Physical Emulation," *Proc. HotNets*, 2003.

[15] P. De *et al.*, "MiNT: A Miniaturized Network Testbed for Mobile Wireless Research," *Proc. Infocom*, 2005.

BIOGRAPHIES

PRADIPTA DE (prade@cs.sunysb.edu) received his B.E. in computer science from Jadavpur University in 1999. He is currently a Ph.D. candidate in computer science at Stony Brook University working in the Experimental Computer Systems Laboratory. He is also a member of the research staff at IBM India Research Laboratory. His main areas of interest are in wireless networks, specifically involved with the design of wireless network testbeds.

ASHISH RANIWALA (raniwala@cs.sunysb.edu) received a B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Delhi, and an M.S. degree in computer science from Stony Brook University, New York, in 1998 and 2000, respectively. He is currently working toward a Ph.D. degree in computer science at Stony Brook University. He conducts research on capacity, fairness, and security issues in enterprise-class wireless mesh networks, and storage systems virtualization.

SRIKANT SHARMA (srikant@cs.sunysb.edu) is a staff member with Symantec Corporation. He received his B.Tech. degree in computer science from Regional Engineering College, Calicut, India, in 1996. He is currently working toward his Ph.D. degree in computer science at Stony Brook University. He was with Wipro Corporation, Karnataka, India, as a senior research engineer. His research interests are in gigabit metro networks, wireless networks, and storage systems.

TZI-CKER CHIUEH (chiueh@cs.sunysb.edu) is a professor in the Computer Science Department of Stony Brook University. He received his B.S. in electrical engineering from National Taiwan University, his M.S. in computer science from Stanford University, and his Ph.D. in computer science from the University of California at Berkeley in 1984, 1988, and 1992, respectively. He received an NSF CAREER award in 1995, an IEEE Hot Interconnect Best Paper award in 1999, and a Long Island Software Award in 1997 and 2004.

There have been several efforts in designing wireless testbeds, but often these efforts are geared toward specific projects. This article pinpoints the feature set necessary for a general-purpose wireless network testbed, related to controllability and analysis of wireless experiments.